# RAJEEV GANDHI MEMORIAL COLLEGE OF ENGINEERING AND TECHNOLOGY, NANDYAL-518501, KURNOOL (DIST), A. P., INDIA

**AUTONOMOUS INSTITUTE**

**(Affiliated to J.N.T.U.A., Ananthapuramu)**

**ACADEMIC REGULATIONS, COURSE STRUCTURE AND DETAILED SYLLABI**

**MCA (Regular) from 2015-16**

**For pursuing Three year Post graduate Degree of study in Master of Computer Applications (MCA) offered by Rajeev Gandhi Memorial College of Engineering and Technology, Nandyal - 518501 under Autonomous status and herein referred to as RGMCET (Autonomous).**

All the rules specified herein approved by the Academic Council will be in force and applicable to students admitted from the Academic Year 2015-16 onwards. Any reference to "Institute" or "College" in these rules and regulations shall stand for Rajeev Gandhi Memorial College of Engineering and Technology (Autonomous).

All the rules and regulations, specified hereafter shall be read as a whole for the purpose of interpretation as and when a doubt arises, the interpretation of the Chairman, Academic Council is final. As per the requirements of statutory bodies, the Principal, Rajeev Gandhi Memorial College of Engineering and Technology shall be the Chairman, Academic Council.

**Academic Regulations 2015 for MCA (Regular)**

(Effective for the students admitted into first year from the Academic Year 2015-2016)
The MCA Degree of the Jawaharlal Nehru Technological University Anantapur, Ananthapuramu shall be conferred on students who are admitted to the program and fulfil all the requirements for the award of the Degree.

## 1.0 Eligibility for Admissions:

Admission to the above program shall be made subject to the eligibility, qualifications and specialization prescribed by the University from time to time.

Admissions shall be made on the basis of merit rank obtained by the qualifying candidate at ICET examination or on the basis of any other order of merit prescribed by Andhra Pradesh State Council of Higher Education (APSCHE) subject to reservations prescribed by the Govt. of A. P., from time to time.

## 2.0 Award of MCA Degree:

**2.1** The student shall be declared eligible for the award of the MCA degree, if he/she pursues a course of study and completes it successfully for not less than three academic years and not more than six academic years.

**2.2** The student, who fails to fulfil all the academic requirements for the award of the degree within six academic years from the year of his admission, shall forfeit his seat in MCA course.

**2.3** The minimum clear instruction days for semester shall be 95.

## 3.0 Attendance:

**3.1** The student shall be deemed to have eligibility to write End Semester examinations if he has secured a minimum of 75% of attendance in aggregate of all the subjects.

**3.2** Condonation of shortage of attendance up to 10%, i. e. 65% and above, and below 75% may be given by the College academic committee.

**3.3** Condonation of shortage of attendance shall be granted only on genuine and valid reasons on representation by the student with supporting evidence.

**3.4 Shortage of attendance below 65% shall in no case be condoned.**

**3.5** The student shall not be promoted to the next semester unless he fulfils the attendance requirements of the previous semester.

**Table 1: Credits**

| Subject | Semester | | MARKS | |
|---|---|---|---|---|
| | Periods / Week | Credits | Internals | Externals |
| Theory | 04 | 04 | 40 (25-internal+15-Assigment) | 60 |
| Practical | 03 | 02 | 50 | 50 |
| Mini project | | 02 | 50 | 50 |
| Comprehensive Viva-voce | | 04 | | 100 |
| Project | | 12 | | |

**Table 2: Course pattern**

| Year | Semester | No.of Subjects | Number of Labs | Total credits | |
|---|---|---|---|---|---|
| I | 1 | 05 | 03 | 5X4=20<br>3X2=06 | 26 |
| | 2 | 05 | 03 | 5X4=20<br>3X2=06 | 26 |
| II | 3 | 05 | 03 | 5X4=20 | 26 |

| | | | | | |
|---|---|---|---|---|---|
| | | | | 3X2=06 | |
| | 4 | 03<br>01MOOC/Elective<br>01 Elective | 03 | 5X4=20<br>3X2=06 | 26 |
| III | 5 | 03<br>01MOOC/Elective<br>01 Elective<br>Mini project<br>Comprehensive Viva | 02 | 5X4=20<br>2X2=04<br>1X2=02<br>1X4=04 | 30 |
| | 6 | Project work | | 12X1=12 | 12 |
| **Total credits** | | | | | **146** |

## 4.0 Evaluation:

**4.1** For theory subjects, the distribution shall be 40 marks for Internal Evaluation (25 marks for internal test and 15 marks for assignments / field work) and 60 marks for the End-Examination.

**4.2** Each Internal Test question paper shall contain 5 questions, of which the First question is compulsory and three questions are to be answered from the remaining four. Compulsory question carries 10 marks (It contains 5 short answer questions). The remaining 3 questions carry 5 marks each. Each question shall have a,b,c.... parts. The duration of internal test will be for 2 hours. First test to be conducted in 3 units in the middle of the semester and second test to be conducted in the remaining 3 units of each subject at end the semester. There shall be two assignments in each subject (problem based/ field work) for the award of 15 marks so that internal component (marks) will be 40 marks (25 marks for internal test+15 marks for assignments / field work). For awarding of 25 Internal marks the performance of the student in two internal examinations conducted will be considered by giving a weightage of 0.75 for the better score and 0.25 for the other score.

**4.3** The End Examination question paper will have 7 questions and students have to answer 5 questions. However, the first question is compulsory and it consists of 6 short answer questions, each carrying 2 marks. The next 4 questions are to be answered from the remaining 6 questions and each carries 12 marks. Each 12 marks question shall have a, b, c .. parts.

**4.4** Elective subjects will commence from 3rd semester. Out of the electives offered in 4th / 5th semesters, one elective will be MOOC / Electives offered by the department. Any student who is interested can opt for the MOOC/ Electives offered by the department and acquire the required credits. Even if the student opts MOOC, he has to write two internal tests besides the end examination conducted by the institute like other subjects. However, he has to obtain the certificate from the organization in which he has registered. Any MOOC selected by the student should be of more than 45 hours duration and also from the reputed organization. Attendance of the student who has opted for MOOC will be taken from the remaining subjects and labs only in that semester while finalizing the attendance for fulfilling the minimum requirements of attendance for promotion to next semester. Attendance will not be recorded for MOOC. Where ever MOOC is opted by the student, the evaluation procedure will be similar to any subject offered by the department.

**4.5** For practical subjects, 50 marks shall be for the End Semester Examinations and 50 marks will be for internal evaluation based on the day-to-day performance. Laboratory examination for MCA Course shall be conducted with two Examiners, one of them being Laboratory Class Teacher and second Examiner shall be outside from the institute (External examiner).

**4.6** Student has to undergo a comprehensive viva pertaining to his specialization which carries 100 marks at the end of 5th semester. He has to secure 50% marks to obtain required credits. Comprehensive viva will be conducted at the end of 5th semester by the committee consisting of HOD, senior faculty member and external Examiner from outside the institute. For this, HOD of the Department shall submit a panel of 4 Examiners, who are eminent in that field. One from the panel will be selected by the principal of the institute as external Examiner for comprehensive viva.

**4.7** The candidate shall be deemed to have secured the minimum academic requirement in a subject if he secures a minimum of 40% of marks in the End Examination and a minimum aggregate of 50% of the total marks in the End Examination and Internal evaluation taken together.

**4.8** In case the candidate does not secure the minimum academic requirement in any subject (as specified in 3.0), he has to reappear for the Semester Examination either supplementary or regular in that subject, or repeat the course when offered next or do any other specified subject as may be required.

## 5.0 Re-registration for Improvement of Internal marks:

Following are the conditions to avail the benefit of improvement of Internal marks.

**5.1** The candidate should have completed the course work and obtained examinations results for 1st & 2nd semesters.

**5.2** He should have passed all the subjects for which the internal marks secured are more than 50%.

**5.3** Out of the subjects the candidate has failed in the examination due to Internal marks secured being less than 50%, the candidate shall be given one chance for each Theory subject and for a maximum of 03 Theory subjects for Improvement of Internal marks.

**5.4** The candidate has to re-register for the chosen subjects and fulfil the academic requirements.

**5.5** For each subject, the candidate has to pay a fee equivalent to one tenth of the semester tuition fee and the amount is to be remitted in the form of D. D. in favour of the Principal, RGMCET payable at RGMCET, Nandyal branch along with the requisition through the HOD of the respective Department.

**5.6** In case of availing the Improvement of Internal marks, the internal marks as well as the End Examinations marks secured in the previous attempt (s) for the re-registered subjects stand cancelled.

## 6.0 Evaluation of Project / Dissertation work:

Every candidate shall be required to submit thesis or dissertation after taking up a topic approved by the Department.

**6.1** Registration of Project work: The candidate is permitted to register for the project work after satisfying the attendance requirement of all the courses (theory and practical courses of 1st to 5th semester)

**6.2** An Internal Department Committee (I.D.C.) consisting of HOD, Supervisor and One Internal senior expert shall monitor the progress of the project work. The project work carries no marks.

**6.3** The work on the project shall be initiated in the beginning of 6th semester and continue throughout the semester. The duration of the project is for one semester. The candidate can submit Project thesis with the approval of I.D.C. at the end of 6th semester.

**6.4** The student must submit status report at least in two different phases during the project work period. These reports must be approved by the I.D.C. before submission of the Project Report.

**6.5** The candidate shall be allowed to submit the thesis / dissertation only after passing in all the prescribed subjects (both theory and practical) and then take viva voce examination of the project. The viva voce examination may be conducted once in two months for all the candidates submitted during that period.

**6.6** Three copies of the Thesis / Dissertation certified in the prescribed form by the supervisor & HOD shall be submitted to the institute.

**6.7** The Department shall submit a panel of three experts for a maximum of 4 students at a time. However, the thesis / dissertation will be adjudicated by the board consisting of HOD, concerned supervisor and one external Examiner from other institute nominated by the principal from a panel of Examiners submitted by the Department to the Controller of Examination.

**6.8** If the report of the board is favourable viva voce examination, the board shall jointly report candidates work as:

1. Good
2. Satisfactory
3. Not satisfactory

If the report of the viva voce is not satisfactory the candidate will retake the viva voce examination after three months. If he fails to get a satisfactory report at the second viva voce examination he will not be eligible for the award of the degree unless the candidate is permitted to revise and resubmit the thesis.

## 7.0 Award of Degree and class:

After the student has satisfied the requirements prescribed for the completion of the program and is eligible for the award of MCA Degree, he shall be placed in one of the following classes:

**Table 3: Award of Class**

| **Class Awarded** | **% of marks to be secured** | Division/ Class | CGPA | From the aggregate marks secured from 146 Credits. |
|---|---|---|---|---|
| First Class with Distinction | 70% and above | First Class With Distinction | $\geq 7.5$ | |
| First Class | Below 70% but not less than 60% | First Class | 6.5 *and* $< 7.5$ | |
| Second Class | Below 60% but not less than 50% | Second Class | $\geq 5.5$ *and* $< 6.5$ | |

(The marks in Internal evaluation and End Examination shall be shown separately in the marks memorandum)

## 8.0 *Grading:*

After each subject is evaluated for 100 marks, the marks obtained in each subject will be converted to a corresponding letter grade as given below, depending on the range in which the marks obtained by the student falls.

**Table 4: Conversion into Grades and Grade points assigned**

| Range in which the % of marks in the subject fall | Grade | Grade point Assigned | Performance |
|---|---|---|---|
| 90 to 100 | O | 10 | Outstanding |
| 80 to 89.9 | $A^+$ | 09 | Excellent |
| 70 to 79.9 | A | 08 | Very good |
| 60 to 69.9 | $B^+$ | 07 | good |
| 50 to 59.9 | B | 06 | Pass |
| <50 | F | 00 | Fail |
| Ab | AB | 00 | Fail |

**8.1** Requirement for clearing any subject: The students have to obtain a minimum of 40% in End Examination and they have to score a minimum of 50% marks from Internal and external exam marks put together to clear the subject. Otherwise they will be awarded fail grade.

**8.2** 'F' is considered as a fail grade indicating that the student has to reappear for the end supplementary examination in that subject and obtain a non - fail grade for clearing that subject.

**8.3** To become eligible for the award of degree the student must obtain a minimum CGPA of 6.0.

## 9.0 Supplementary Examinations:

Apart from the regular End Examinations, the institute may also schedule and conduct supplementary examinations for all subjects for the benefit of students with backlogs. Such students writing supplementary examinations as supplementary candidates may have to write more than one examination per day. **The student is not permitted to improve his performance in any subject in which he has obtained pass grade.**

## 10.0 *Grade Point Average (GPA) and Cumulative Grade Point Average(CGPA)*

The Grade Point Average (GPA) for each semester and Cumulative Grade Point Average (CGPA) up to any semester are calculated as follows:

**i)** Semester Grade Point Average will be computed as follows:

$$GPA = \frac{\sum_1^n C_i \times GP_i}{\sum_1^n C_i}$$

Where, n is the number of subjects in that semester. $C_i$ is Credits for the subjects. $GP_i$ is the grade point obtained for the subject and the summation is over all the subjects in that semester.

**ii)** A Cumulative Grade Point Average (CGPA) will be computed for every student at the end of each semester. The CGPA would give the cumulative performance of The student from the first semester up to the end of the semester to which it refers and is calculated as follows

$$CGPA = \frac{\sum_1^m GPA_j \times TC_i}{\sum_1^m TC_j}$$

Where, m is the number of semester under consideration. $TC_j$ the total number of credits for a $j^{th}$ semester and $GPA_j$ is the Grade Point Average of the $j^{th}$ semester. Both GPA and CGPA will be rounded off to the second digit after decimal and recorded as such.

While computing the GPA / CGPA , the subjects in which the student is awarded zero grade points will also be included.

## 11.0 *Grade Sheet:*

A grade sheet (Memorandum) will be issued to each student indicating his performance in all subjects of that semester in the form of grades and also indicating the GPA and CGPA.

## 12.0 Transcripts:

After successful completion of the total course of study, a Transcript containing performance of all academic years will be issued as a final record. Duplicate transcripts will also be issued if required after the payment of requisite fee and also as per norms in vogue.

## 13.0 Minimum Instruction Days:

The minimum instruction days for each semester shall be 95 clear instruction days excluding the days allotted for tests/examinations and preparation holidays declared, if any.

## 14.0 Amendment of Regulations:

The college may, from time to time, revise, amend or change the regulations, scheme of examinations and syllabi. However the academic regulations of any student shall be same throughout the course of study in which the student has been admitted.

## 15.0 Transfers:

There shall be no branch transfers After the completion of admission process.

## 16.0 Withholding of results:

If the candidate has not paid any dues to the institute or if any case of in-discipline is pending against him, the result of the candidate will be withheld and he will not be allowed for the next semester. The issue of the degree is liable to be withheld in such cases.

## 17.0 Transitory Regulations:

Candidates who have discontinued or have been detained for want of attendance are eligible for admission to the same or equivalent subjects as and when subjects are offered, subject to 2.0 and 3.0

## 18.0 Rules of Discipline:

**18.1** Any attempt by any student to influence the teachers, Examiners, faculty and staff of Examination section for undue favours in the exams, and bribing them either for marks or attendance will be treated as malpractice cases and the student can be debarred from the college.

**18.2** When the student absents himself, he is treated as to have appeared and obtained zero marks in that subject (s) and grading is done accordingly.

**18.3** When the performance of the student in any subject (s) is cancelled as a punishment for indiscipline, he is awarded zero marks in that subject (s).

**18.4** When the student's answer book is confiscated for any kind of attempted or suspected malpractice, the decision of the Chief Superintendent is final.

## 19.0 General:

**19.1** The Academic Regulations should be read as a whole for the purpose of any interpretation.

**19.2** In the case of any doubt or ambiguity in the interpretation of the above rules, the decision of the College Academic Council is final.

**19.3** The Institute may change or amend the academic regulations or syllabi at any time and the changes or amendments made shall be applicable to all the students with effect from the dates notified by the Institute.

**19.4** Where the words “he”, “him”, “his”, occur in the regulations, they include “she”, “her”, “hers”.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

## I YEAR MCA I-SEMESTER

COURSE STRUCTURE

| Code | Subject | Theory | Practical | Credits | Scheme of Examination | | |
|---|---|---|---|---|---|---|---|
| | | | | | Internal Marks | External Marks | Total Marks |
| F0001151 | Probability and Statistics | 3+1 | - | 4 | 40 | 60 | 100 |
| F0002151 | Technical Communication | 3+1 | - | 4 | 40 | 60 | 100 |
| F0003151 | Accounting and Financial Management | 3+1 | - | 4 | 40 | 60 | 100 |
| F0004151 | Computer Organization | 3+1 | - | 4 | 40 | 60 | 100 |
| F0005151 | C-Programming | 3+1 | - | 4 | 40 | 60 | 100 |
| F0006151 | C-Programming Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0007151 | Computer Organization Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0008151 | IT Workshop | - | 3 | 2 | 50 | 50 | 100 |
| **Total** | | **20** | **9** | **26** | **350** | **450** | **800** |

## I YEAR MCA II-SEMESTER

| Code | Subject | Theory | Practical | Credits | Scheme of Examination | | |
|---|---|---|---|---|---|---|---|
| | | | | | Internal Marks | External Marks | Total Marks |
| F0009152 | Organization Structure and Personnel Management | 3+1 | - | 4 | 40 | 60 | 100 |
| F0010152 | Data Structure through C | 3+1 | - | 4 | 40 | 60 | 100 |
| F0011152 | Mathematical Foundations of Computer Science | 3+1 | - | 4 | 40 | 60 | 100 |
| F0012152 | Operating Systems | 3+1 | - | 4 | 40 | 60 | 100 |
| F0013152 | UNIX and Shell Programming | 3+1 | - | 4 | 40 | 60 | 100 |
| F0014152 | Data Structure through C Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0015152 | UNIX and Shell Programming Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0016152 | Operating Systems Lab | - | 3 | 2 | 50 | 50 | 100 |
| **Total** | | **20** | **9** | **26** | **350** | **450** | **800** |

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

## II YEAR MCA I-SEMESTER

COURSE STRUCTURE

| Code | Subject | Theory | Practical | Credits | Scheme of Examination | | |
|---|---|---|---|---|---|---|---|
| | | | | | Internal Marks | External Marks | Total Marks |
| F0017153 | Software Engineering | 3+1 | - | 4 | 40 | 60 | 100 |
| F0018153 | Database Management Systems | 3+1 | - | 4 | 40 | 60 | 100 |
| F0019153 | Computer Networks | 3+1 | - | 4 | 40 | 60 | 100 |
| F0020153 | Object Oriented Programing Using C++ | 3+1 | - | 4 | 40 | 60 | 100 |
| F0021153 | Operations Research | 3+1 | - | 4 | 40 | 60 | 100 |
| F0022153 | Object Oriented Programing Using C++ Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0023153 | Database Management Systems Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0024153 | Computer Networks Lab | - | 3 | 2 | 50 | 50 | 100 |
| | **Total** | **20** | **9** | **26** | **350** | **450** | **800** |

## II YEAR MCA II-SEMESTER

| Code | Subject | Theory | Practical | Credits | Scheme of Examination | | |
|---|---|---|---|---|---|---|---|
| | | | | | Internal Marks | External Marks | Total Marks |
| F0025154 | Object Oriented Programming Through Java | 3+1 | - | 4 | 40 | 60 | 100 |
| F0026154 | Object Oriented Analysis and Design Using UML | 3+1 | - | 4 | 40 | 60 | 100 |
| F0027154 | Software Testing Methodologies | 3+1 | - | 4 | 40 | 60 | 100 |
| | **ELECTIVE-I** | 3+1 | - | 4 | 40 | 60 | 100 |
| F0028154 | Human Computer Interaction | | | | | | |
| F0029154 | Computer Graphics | | | | | | |
| F0030154 | Information Security | | | | | | |
| | **ELECTIVE-II** | 3+1 | - | 4 | 40 | 60 | 100 |
| F0031154 | Distributed Systems | | | | | | |
| F0032154 | Artificial Intelligence | | | | | | |
| F0033154 | Advanced Computer Architecture | | | | | | |
| F0034154 | Object Oriented Programming Through Java Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0035154 | Case Tools Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0036154 | Software Testing Methodologies Lab | - | 3 | 2 | 50 | 50 | 100 |
| | **Total** | **20** | **9** | **26** | **350** | **450** | **800** |

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

**III YEAR MCA I-SEMESTER**

COURSE STRUCTURE

| Code | Subject | Theory | Practical | Credits | Scheme of Examination | | |
|---|---|---|---|---|---|---|---|
| | | | | | Internal Marks | External Marks | Total Marks |
| F0037155 | Web Technologies | 3+1 | - | 4 | 40 | 60 | 100 |
| F0038155 | Data Warehousing and Data Mining | 3+1 | - | 4 | 40 | 60 | 100 |
| F0039155 | Mobile Application Development | 3+1 | - | 4 | 40 | 60 | 100 |
| | **ELECTIVE-III** | 3+1 | - | 4 | 40 | 60 | 100 |
| F0040155 | E-Commerce | | | | | | |
| F0041155 | Distributed Databases | | | | | | |
| F0042155 | Image Processing | | | | | | |
| | **ELECTIVE-IV** | 3+1 | - | 4 | 40 | 60 | 100 |
| F0043155 | Software Project Management | | | | | | |
| F0044155 | Middleware Technologies | | | | | | |
| F0045155 | Cloud Computing | | | | | | |
| F0046155 | Web Technologies Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0047155 | Mobile Application Development Lab | - | 3 | 2 | 50 | 50 | 100 |
| F0048155 | Mini Project | - | 3 | 2 | 50 | 50 | 100 |
| F0049155 | Comprehensive Viva | - | | 4 | | 100 | 100 |
| | **Total** | **20** | **9** | **30** | **350** | **550** | **900** |

**III YEAR MCA II-SEMESTER**

| Code | Subject | Credits | Internal Marks | External Marks | Total |
|---|---|---|---|---|---|
| F0050156 | Project work | 12 | - | - | - |

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

**MASTER OF COMPUTER APPLICATIONS MCA-R15**

I Year, I-Sem (MCA) T 4 C 4

## (F0001151) PROBABILITY AND STATISTICS

**OBJECTIVES:**

- Probability & Statistics is a necessary avenue to scientific knowledge which opens new vistas of mental activity.
- There was a great need for an associate knowledge on Probability & Statistics for the engineering students.
- It is accepted that a good mathematical studying is essential for all engineers; this will definitely boost the confidence of the student in writing "Competitive examinations".

**OUTCOMES:**

By the end of module students will be expected to demonstrate

- Knowledge of Probability, Conditional Probability, Baye's theorem and its applications, Random variables, Discrete random variables, Continuous random variables, Binomial Distribution, Poisson Distribution, Normal Distribution, Population& Samples, Sampling Distribution of means,Point Estimation, interval Estimation, Bayesian Estimation
- Tests of Hypothesis, Z – Distribution, Student t – test, F – test, Chi square test Queuing theory, Basic Queuing process, Transient & Steady states, Pure birth & Death process.

**UNIT – I**

Probability: Sample Space and events – Probability – The axioms of Probability – Some Elementary theorems – Conditional Probability – Baye's theorem.

**UNIT – II**

Random Variables: Discrete and continuous - Distribution – Distribution functions – Properties – Discrete Random variables – Probability mass function – Continuous Random variables – Probability density function.

**UNIT – III**

Binomial, Poisson and Normal distributions – Related properties – Fitting distributions.

**UNIT – IV**

Sampling distribution : Population and samples – Sampling distribution of mean (known and unknown )

Estimation: Point estimation – Interval estimation – Bayesian estimation.

**UNIT – V**

Test of Hypothesis – Means – Proportions – Hypothesis concerning one and two means – One tail, two tail tests – Type I and Type II errors.

**UNIT – VI**

Tests of significance – Student's t –test, F – test, $\Psi^2$ test – Good ness of fit – Contigency test.

**TEXT BOOKS:**

1) Probability and Statistics by T.K.V. Iyengar, B. Krishna Gandhi and others, S.Chand and company.
2) A Text book of Probability and Statistics by Dr. ShahnazBathul.
3) Engineering Mathematics by B.V. Ramana, Tata McGraw Hill .

**REFERENCES:**

1) Fundamental Mathematical Statistics by S.C. Guptha and V.K. Kapoor – S. Chand Co
2) A text book of Engineering Mathematics by N.P. Bali, Iyengar – Lakshmi Publications (Pvt ltd)
3) Engineering Mathematics – III A by Dr.M.K. Venkataraman – The National Publishing co.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

**MASTER OF COMPUTER APPLICATIONS MCA-R15**

I Year, I-Sem (MCA) T 4 C 4

## (F0002151) TECHNICAL COMMUNICATION

**OBJECTIVE:**

- The objective of this Course is to understand the communication concepts and to develop the students' personality traits and professional spirit. Assuming that the students are fairly proficient in the basic communication skills of listening, speaking, reading and writing in English the course aims to train them in communicating efficiently in the workplace and professional contexts.

**OUTCOMES**

- Display competence in oral, written, and visual communication.
- Apply communication theories.
- Show an understanding of opportunities in the field of communication.
- Communicate ethically.
- Demonstrate positive group communication exchanges.
- Demonstrate critical and innovative thinking.

**UNIT-I**

Communication - Process – Barriers – Principles - Active Listening – ROAR Technique.

**UNIT-II**

Non-Verbal Communication-Kinesics – Chronemics- Oculesics- Paralinguistics.

**UNIT-III**

Corporate Communication – Types of Communication - Professional Etiquettes.

**UNIT-IV**

Effective Meetings – Video Conferencing – Negotiation Skills.

**UNIT-V**

Technical Writing - Letter Writing – Report Writing – Email.

**UNIT VI**

Reading Skills – SQR3 Technique - Fiction analysis – Robin Sharma's The Monk Who Sold His Ferari.

**REFERENCES:**


1 Essentials of Business Communication, Rajendra Pal, JS.Korlahhi, S.Chand.

2 Technical Communication (Principles and Practice) by Meenakshi Raman & Sangeeta Sharma, Oxford University 2004.

3 The Dynamics of Successful Personality and projection (Second Edition) by– J.R. Bhatti, PEARSON 2011.

4 The ACE of Soft Skills (Attitude, Communication and Etiquette for success) by- Gopalaswamy Ramesh & Mahadevan Ramesh , PEARSON 2010.

5 Technical writing 3rd edition by *Sharon J. Gerson & Steven M. Gerson* Pearson Education 2001.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

**MASTER OF COMPUTER APPLICATIONS MCA-R15**

| I Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

## (F0003151) ACCOUNTING AND FINANCIAL MANAGEMENT

**OBJECTIVES:**

- To study the objectives, functions, importance and limitations of Accounting.
- To understand the principles of preparation of Final Accounts.
- To know the preparation of financial reports.
- To know the Cost Volume Profit Analysis and understand the Break- Even- Chart.
- To know the basics of Financial Management.

**OUT COMES:**

- Find out profitability and financial position of a business organization.
- To take product production related decision in a company.
- To understand the financial function environment in a business organization.

**UNIT I:**
Introduction to Accounting: Principles, concepts and conventions, double entry system of accounting, classification of accounts and debit-credit rules.

**UNIT II:**
Financial Statements: Introduction to basic books of accounts, journal and ledger – trial balance – preparation of final accounts: trading account, profit and loss account and balance sheet.

**UNIT III:**
Financial Analysis through ratios: Ratio Analysis – classification of ratios – short term solvency and long term solvency – profitability ratios – analysis and interpretation of financial statements through ratios of liquidity, solvency and profitability.

**UNIT IV:**
Break Even Analysis: Concept of Break Even Point, Cost-Volume-Profit Analysis, Determination of Break Even Point, Margin of Safety and P/V ratio, Impact of changes in cost or selling price on BEP, Practical applications of Break Even Analysis.

**UNIT V:**
Introduction to Financial Management: Meaning and scope, role of financial manager, objectives of time value of money – goals of financial management, leverages: operation, financial leverage and combined leverage.

**UNIT VI:**
Capital Budgeting: Capital and its significance, types of capital, estimation of fixed and working capital requirements, methods and sources of raising capital. Capital budgeting: features, proposals, methods of capital budgeting, payback method, accounting rate of return (AAR), Net Present Value Method (NPV) and Internal Rate of Return (IRR) -simple problems.

**TEXT BOOKS:**

1. Financial Accounting, S.N.Maheshwari, Sultan Chand, 2009.
2. Financial Management and Policy, Van Horne, James,C., Pearson ,2009.

**REFERENCES:**

1. Financial Accounting, Tulsian, S Chand, 2009.
2. Financial Statement Analysis, Khan and Jain, PHI, 2009.
3. Financial Management, I.M.Pandey, Vikas Publications.
4. Financial Management, Bhat Sundhindra, Excel: 2009.
5. Financial Management, Prasanna Chandra, T.M.H, 2009.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

**MASTER OF COMPUTER APPLICATIONS MCA-R15**

| I Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

## (F0004151) COMPUTER ORGANIZATION

**OBJECTIVES:**

- To understand the structure, function, characteristics and performance issues of computer systems.
- To understand the design of the various functional units of digital computers.
- To understand the basic processing unit and how they are connected and how it generates control signals (using hardwired and micro programmed approaches).
- To understand the architecture of 8086.
- To understand the different types of memory and how they are related.
- To learn basics of Parallel Computing and Pipelining.

**OUTCOMES:**

- Students will learn about computer performance, computer design, and trade-offs between cost and performance as well as between hardware and software.
- Students will formulate and solve problems, understand the performance requirements of systems.
- Students will learn to communicate effectively and learn to think creatively and critically, both independently and with others.

**UNIT I**

**NUMBER SYSTEMS AND COMPUTER ARITHMETIC**- Signed and unsigned numbers, Addition and subtraction, multiplication, division, Floating point representation, logical operation, Gray code, BCD codes, Error detecting codes, Boolean algebra, Simplification of Boolean expressions, K-Maps.

**UNIT II**

**MEMORY ORGANIZATION** - Memory hierarchy, Main memory-RAM, ROM chips, Memory address map, memory contention to CPU, Associative Memory-Hardware logic, match, read and write logic, Cache Memory-Associative mapping, Direct mapping, Set-associative mapping, hit and miss ratio.

**UNIT III**

**MICRO PROGRAMMED CONTROL** - Control memory, Address sequencing, microprogram example, design of control unit, Hard wired control, Microprogrammed control.

**UNIT IV**

**BASIC CPU ORGANIZATION** - Introduction to CPU, Instruction formats-INTEL-8086 CPU architecture-Addressing modes - generation of physical address- code segment registers, Zero, one, two, and three address instructions.

**UNIT V**

**INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS** - Data transfer instructions-input-output instructions, address transfer, Flag transfer, arithmetic, logical, shift, and rotate instructions.

**UNIT VI**

**INTEL 8086 ASSEMBLY LANGUAGE INSTRUCTIONS** - Conditional and unconditional transfer, iteration control, interrupts and process control instructions, assembler directives, Programming with assembly language instructions.

**TEXT BOOKS:**

1. Computer System Architecture, M. Morris Mano , 3rd Edition, PHI/Pearson Education,2008.
2. Microprocessors and Interfacing, Douglas Hall, Tata McGraw-Hill.

**REFERENCES:**

1. Computer Organization, Carl Hamacher, ZvonkoVranesic, SafwatZaky, Vth Edition, McGraw Hill.
2. Fundamentals of Computer Organization and Design, SivaramaP.Dandamudi, Springer Int. Edition.
3. Computer Organization and Architecture, William Stallings, 7th Edition, Pearson/PHI, 2007.
4. Digital Design, M. Morris Mano, PHI/Pearson Education.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

**MASTER OF COMPUTER APPLICATIONS MCA-R15**

I Year, I-Sem (MCA) T 4 C 4

## (F0005151) C-PROGRAMING

**OBJECTIVES:**

- To make students aware about fundamentals of computer programming.
- To provide exposure on C programming language.
- To provide exposure on various C programming concepts like arrays, functions, pointers, structures, etc.
- To develop solutions for various problems by using C Programming Language by students.
- To provide exposure on various sorting and searching techniques.

**OUTCOMES:**

By the end of this course, students should be able

- To understand about the various techniques for problem solving**.**
- To understand the fundamental concepts of C language like data types, keywords, operators, Input/Output functions and control statements.
- To understand how to develop C programs to solve various kinds of problems by using different C programming concepts like arrays, functions, pointers and structures.
- To develop programs by performing I/O operations through Files.
- To implement various searching and sorting techniques.

**UNIT I**

**OVERVIEW OF COMPUTER PROGRAMMING**

**Fundamentals of computers** - Evolution of computer systems, Basic anatomy of computer system, Components of computer. Introduction to Computer Programming languages.Problem solving techniques - Algorithms and Flowcharts. How to trace an algorithm. Program control structures – sequence, selection and iteration.

**INTRODUCTION TO C LANGUAGE**

History of C language, Importance of C language, Definition of a C Program, General Form of a C Program. Various Data Types supported by the C language. C tokens – Identifiers, Key words, Variables, Constants, Operators. Operator precedence and Associativity, Expressions and their evaluation process.Type Conversions, Managing Input/Output operations, Control Statements, Loop Constructs.

**UNIT II**

**ARRAYS**

**Arrays –** Definition, Types of arrays, Declaration of One dimensional array, initialization of one dimensional array, storing and accessing the elements from a one dimensional array. Two-dimensional Arrays and their declaration, initialization, storing & accessing elements from it.Declaration of multi dimensional array, initialization of multi dimensional arrays, storing and accessing the elements from a multi dimensional array.

**Functions:** Introduction, Library Functions and User defined functions. Need for user-defined functions. General form of a function, Elements of an user defined functions-Function definition ,Function call, Function declaration, Function name, return type, parameters, return statements. Categorization of functions with respect to parameters and return values. Definition of Scope of a variable with suitable examples.

**UNIT III**

**STORAGE CLASSES AND STRINGS**

**Storage Classes** - Automatic, External, Static, and Register. Arrays and functions - Passing an entire array as an Argument to a function. Recursion – Need of recursive functions, Solving Towers of Hanoi Problem using recursive function and its trace out.

**Strings** - Definition, Declaring and initializing strings, Basic Operations on strings, String handling Functions, Table of strings.

**UNIT IV**

**POINTERS**

Introduction, Need of using pointer variables, Pointer variable declaration, initialization of pointer variables, how to access a value from a memory location through its pointer variable. Arithmetic operations on pointer variables.

**Pointers & Functions** - pointers as function arguments (i.e., call-by-reference), Pointers and Arrays, Pointers and Strings, Array of Pointers, Pointers to Pointers, Generic Pointers, Pointer to Functions. Example Programs on the topics mentioned above.

**UNIT V**

**STRUCTURE**

**Structures** – Introduction, Features of Structures. Declaration and Initialization of Structures, Accessing structure members, structure initialization, Nested Structures, Array of Structures, Arrays within structures and Pointers to Structures, Structures and Functions, Bit Fields, Unions, Union of Structures, Dynamic Memory Allocation Functions.

**UNIT VI**

**FILE INPUT/OUTPUT**

Introduction, Types of Files, File I/O Operations- High level I/O functions- Open & Close a file, Read and Write data into a file, Searching data in the file, Error handling during I/O operations on files. Command Line Arguments, Applications of Command Line Arguments.

**TEXT BOOKS:**

1. Computer programming and Data Structures, E.Balaguruswamy, Tata McGraw Hill. 2009 revised edition.
2. Programming in C and Data Structures, J.R.Hanly, Ashok N. Kamthane and A. AnandaRao, Pearson Education.

**REFERENCES:**

1. Let us C – Yeshwanthkanetkar, 5th Edition.BPB Publications.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press.
3. The C Programming Language, Brian W.Kerninghan, Dennis M.Ritchie.
4. Data Structures using C – A.M.Tanenbaum,Y.Langsam, and M.J. Augenstein, Pearson Education / PHI, Eighth Edition.
5. C Programming & Data Structures, B.A.Forouzan and R.F. Gilberg, Third Edition, Cengage Learning.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

## MASTER OF COMPUTER APPLICATIONS MCA-R15

I Year, I-Sem (MCA) P C
3 2

### (F0006151) C-PROGRAMING LAB

**Objectives**:

- To make the student learn a programming language.
- To teach the student to write programs in C to solve typical problems.
- To introduce the student to simple linear data structures such as lists, stacks, queues.

**Recommended Systems/Software Requirements:**

- Intel based desktop PC with ANSI C Compiler and Supporting Editors.

**Exercise 1:**

a) Write a C program to find the roots of a quadratic equation.

b) Write a C program to calculate the following Sum:

Sum=1-x2/2! +x4/4!-x6/6!+x8/8!-x10/10!

**Exercise 2:**

a) Write a C program, which takes two integer operands and one operator from the user, performs the specified operation and then prints the result.

(Consider the operators +,- ,*, /, % and use Switch Statement)

b) The total distance travelled by vehicle in 't' seconds is given by distance S = ut+1/2at2 where 'u' and 'a' are the initial velocity (m/sec.) and acceleration (m/sec2) respectively.

Write C program to find the distance travelled at regular intervals of time given the values of 'u' and 'a'. The program should provide the flexibility to the user to select his own time intervals and repeat the calculations for different values of 'u' and 'a'.

**Exercise 3:**

a) Write a C program to find the sum of individual digits of a positive integer.

b) Write a C program to generate the first 'n' terms of the Fibonacci sequence.

[Note: A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0 and 1. Subsequent terms are found by adding the preceding two terms in the sequence.]

c) Write a C program to generate all the prime numbers between 1 and n, where 'n' value is given by the user.

[Note: Develop each of the above programs by using different loop constructs supported by C language. (i.e., while, do while and for Loops)]

**Exercise 4:**

a) Write a C program to generate Pascal's triangle.

b) Write a C program to construct a pyramid of numbers.

**Exercise 5:**

a) Write a C program to find all the even numbers in the given one dimensional array.

b) Write a C program to print the elements of an array in reverse order.

c) Write a C program to perform the following operations:

i) Addition of Two Matrices ii) Subtraction of Two Matrices

iii)Multiplication of Two Matrices

[Note: Use functions to implement the above specified operations]

**Exercise 6:**

a) Write C programs that use both recursive and non-recursive functions

i) To find the factorial of a given integer.

ii) To find the GCD (greatest common divisor) of two given integers.

iii) To reverse a given positive integer.

**Exercise 7:**

a) Write a C Program to solve the Towers of Hanoi problem by using recursive function.

b) Write a C Program to demonstrate the various storage classes, which are supported by the C language. [i.e., automatic, external, static and register]

**Exercise 8:**

a) Write a C Program to demonstrate that, how to pass an entire array as an argument to a function with a suitable example.

b) Write a C Program to perform various operations on given two strings using string

handling functions.

**Exercise 9:**

a) Write a C Program to perform various arithmetic operations on pointer variables.

b) Write a C Program to demonstrate the following parameter passing mechanisms:

i) call-by-valueii) call-by-address

**Exercise 10:**

a) Write a C program that uses functions to perform the following operations:

i) To insert a sub-string in to a given main string from the specified position.

ii) To delete 'n' Characters from a given position in a given string.

b) Write a C program to determine if the given string is a palindrome or not.

**Exercise 11:**

a) Write a C program that displays the position or index in the string 'S' where the string 'T' begins, or – 1 if 'S' doesn't contain 'T'.

b) Write a C program to count the lines, words and characters in a given text.

**Exercise 12:**

a) 2's complement of a number is obtained by scanning it from right to left and complementing all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a C program to find the 2's complement of a binary number.

b) Write a C program to convert the given Roman numeral to its decimal equivalent value.

**Exercise 13:**

a) Write a C program that uses functions to perform the following operations:

i) Reading a complex number ii) Writing a complex number

iii) Addition of two complex numbers iv) Multiplication of two complex numbers

(Note: Represent the complex number using a structure.)

**Exercise 14:**

a) Write a C program which copies contents of one file to another file.

b) Write a C program to reverse the first 'n' characters in a file.

(Note: The file name and n are specified on the command line.)

**Exercise 15:**

a) Write a C program to display the contents of a file using command line arguments.

b) Write a C program to merge two files into a third file (i.e., the contents of the first file followed by the contents of the second file are put in the third file)

(Note: The file name and n are specified on the command line.)

**REFERENCE BOOKS**

1. The Spirit of C, an introduction to modern programming, M.Cooper, Jaico Publishing House.
2. Mastering C, K.R. Venugopal and S.R. Prasad, TMH Publications.
3. Computer Basics and C Programming, V. Raja Raman, PHI Publications.

I Year, I-Sem (MCA) P C
3 2

**(F0007151) COMPUTER ORGANIZATION LAB**

**OBJECTIVES:**

- To understand the design of the various functional units of digital computers.
- To understand the basic processing unit and how they are connected and how it generates control signals.
- To understand the architecture of 8086.
- To understand the different types of memory and how they are related.

**OUTCOMES:**

- Students will learn about computer performance, computer design, and trade-offs between cost and performance as well as between hardware and software.
- Students will formulate and solve problems, understand the performance requirements of systems.
- Students will learn to communicate effectively and learn to think creatively and critically, both independently and with others.

**List of Sample Problems/Experiments:**

Write assembly language programs for the following using TASM.

1. Write an 8086 assembly language program to add two 8-bit numbers?
2. Write an 8086 assembly language program to add two 16- bit numbers?
3. Write an 8086 assembly language program to illustrate AAA instruction?
4. Write an 8086 assembly language program to illustrate AAS instruction?
5. Write an 8086 assembly language program to subtract two 8-bit numbers?
6. Write an 8086 assembly language program two subtract two 16-bit numbers?
7. Write an 8086 assembly language program to find one 's complement?
8. Write an 8086 assembly language program to two's complement?
9. Write an 8086 assembly language program to find sum of 1 to 100 numbers?
10. Write an 8086 assembly language program to find two's complement using NEG instruction?
11. Write an 8086 assembly language program to illustrates shift instructions?
12. Write an 8086 assembly language program to illustrates rotate instructions?
13. Write an 8086 assembly language program to find the biggest of two numbers?
14. Write an 8086 assembly language program to find the smallest of two numbers?
15. Write an 8086 assembly language program to find average of two numbers?
16. Write an 8086 assembly language program to calculate the following expression?
    A=B+C-D*E
17. Write an 8086 assembly language program to calculate the following expression?
    Z=X*Y+W-V+U/K
18. Write an 8086 assembly language program to illustrates AND,OR,XOR instructions?
19. Write an 8086 assembly language program to illustrates test instructions?
20. Write an 8086 assembly language program to illustrates MUL instruction?

21. Write an 8086 assembly language program to illustrate AAM instruction?
22. Write an 8086 assembly language program to divide two numbers?
23. Write an 8086 assembly language program to sort numbers using bubble sort?
24. Write an 8086 assembly language program to sort numbers in descending order?
25. Write an 8086 assembly language program to find factorial using procedures?
26. Write an 8086 assembly language program to add two 8-bit numbers using procedures?
27. Write an 8086 assembly language program to add two 16-bit numbers using procedures?
28. Write an 8086 assembly language program to multiply two 16-bit numbers using procedures?
29. Write an 8086 assembly language program to find sum of array elements?
30. Write an 8086 assembly language program to find average of array elements?
31. Write an 8086 assembly language program to find largest number in an array?
32. Write an 8086 assembly language program to find smallest number in an array?
33. Write an 8086 assembly language program to convert decimal number to hexadecimal number?
34. Write an 8086 assembly language program to convert ASCII number to packed BCD?
35. Write an 8086 assembly language program to accept a character from key board and print character?

**REFERENCES:**

1) IBM PC Assembly Language and Programming, P. Abel, 5th Edition, PHI/Pearson Education.
2) Introduction To Assembly Language Programming, Sivarama P.Dandamudi, Springer Int. Edition,2003.
3) The 8088 and 8086 Microprocessors: Programming , Interfacing, Software, Hardware and Application,4th edition, W.A.Triebel, A.Singh, N.K.Srinath, Pearson Education.

I Year, I-Sem (MCA) P 3 C 2

**(F0008151) IT WORKSHOP**

**Objectives:**
The IT Workshop for engineers is a training lab course spread over 40 hours. The modules include training on PC Hardware, Internet & World Wide Web and Productivity tools including Word, Excel, Power Point.

**PC Hardware** introduces the students to a personal computer and its basic peripherals, the process of assembling a personal computer, installation of system software like MS Windows, Linux and the required device drivers. In addition hardware and software level troubleshooting process, tips and tricks would be covered. **The students should work on working PC to disassemble and assemble to working condition and install Windows and Linux on the same PC. Students are suggested to work similar tasks in the Laptop scenario wherever possible.**
**Internet & World Wide Web** module introduces the different ways of hooking the PC on to the internet from home and workplace and effectively usage of the internet. Usage of web browsers, email.

**Productivity tools** module would enable the students in crafting professional word documents, excel spread sheets and power point presentations.

**PC Hardware**
**PC Hardware**
**Exercise 1 - Task 1:** Identify the peripherals of a computer, components in a CPU and its functions. Draw the block diagram of the CPU along with the configuration of each peripheral and submit to your instructor.
**Exercise 2 - Task 2:** Every student should disassemble and assemble the PC back to working condition. Lab instructors should verify the work and follow it up with a Viva. Also students need to go through the video which shows the process of assembling a PC. A video shall be given as part of the course content.
**Exercise 3 - Task 3:** Every student should individually install MS windows on the personal computer. Lab instructor should verify the installation and follow it up with a Viva.
**Exercise 4 - Task 4:** Every student should install Linux on the computer. This computer should have windows installed. The system should be configured as dual boot with both windows and Linux. Lab instructors should verify the installation and follow it up with a Viva
**Exercise 5 - Task 5:** Hardware Troubleshooting: Students have to be given a PC which does not boot due to improper assembly or defective peripherals. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva
**Exercise 6 - Task 6:** Software Troubleshooting: Students have to be given a malfunctioning CPU due to system software problems. They should identify the problem and fix it to get the computer back to working condition. The work done should be verified by the instructor and followed up with a Viva.

**OFFICE TOOLS**

**WORD**
**Exercise 7** - **Word Orientation:** The mentor needs to give an overview of LaTeX and Microsoft (MS) office 2007,2010/ equivalent tool word: Importance of LaTeX and MS office 2007,2010/ equivalent tool Word as word Processors, Details of the four tasks and features that would be covered in each. Accessing, overview of toolbars, saving files, Using help and resources, rulers, format painter in word.

**Task 1-Task IV:** Using Word to create project certificate. Features to be covered:- Formatting Fonts in word, Drop Cap in word, Applying Text effects, Using Character Spacing, Borders and Colors, Inserting Header and Footer, Using Date and Time option in Word.

**INTRODUCTION TO LATEX**

**Excel**

**Exercise 8 - Excel Orientation:** The mentor needs to tell the importance of MS office 2007,2010/ equivalent tool Excel as a Spreadsheet tool, give the details of the four tasks and features that would be covered in each. Using Excel – Accessing, overview of toolbars, saving excel files, Using help and resources.

**Task 1-Task IV:** Features to be covered:- Gridlines, Format Cells, Summation, auto fill, Formatting Text, Formulas, Functions

**Power Point**

**Exercise 9 - Task1**: Students will be working on basic power point utilities and tools which help them create basic power point presentation. Topic covered during this Exercise includes :- PPT Orientation, Slide Layouts, Inserting Text, Word Art, Formatting Text, Bullets and Numbering, Auto Shapes, Lines and Arrows in Powerpoint. Students will be given model power point presentation which needs to be replicated (exactly how it's asked).

**Exercise 10 - Task 2 :** Second Exercise helps students in making their presentations interactive. Topic covered during this Exercise includes: Hyperlinks, Inserting –Images, Clip Art, Audio, Video, Objects, Tables and Charts

**Internet & World Wide Web 2 Exercises**

**Exercise 11 - Task 1:** Orientation & Connectivity Boot Camp: Students should get connected to their Local Area Network and access the Internet. In the process they configure the TCP/IP setting. Finally students should demonstrate, to the instructor, how to access the websites and email. If there is no internet connectivity preparations need to be made by the instructors to simulate the WWW on the LAN.

**Web Browsers, Surfing the Web:** Students customize their web browsers with the LAN proxy settings, bookmarks, search toolbars and pop up blockers.

**Exercise 12 - Task 2:** Search Engines & Netiquette: Students should know what search engines are and how to use the search engines. A few topics would be given to the students for which they need to search on Google. This should be demonstrated by the student to the satisfaction of instructors.

**Cyber Hygiene:** Students would be exposed to the various threats on the internet and would be asked to configure their computer to be safe on the internet. They need to first install an anti virus software, configure their personal firewall and windows update on their computer.

**REFERENCES:**

1. Comdex Information Technology course tool kit Vikas Gupta, WILEY Dream tech
2. The Complete Computer upgrade and repair book,3rd edition Cheryl A Schmidt, WILEY Dreamtech
3. Introduction to Information Technology, ITL Education Solutions limited, Pearson Education.
4. PC Hardware and A+Handbook – Kate J. Chase PHI (Microsoft)
5. LaTeX Companion – Leslie Lamport, PHI/Pearson.
6. IT Essentials PC Hardware and Software Companion Guide Third Edition by David Anfinson and Ken Quamme. – CISCO Press, Pearson Education.
7. IT Essentials PC Hardware and Software Labs and Study Guide Third Edition by Patrick Regan – CISCO Press, Pearson Education.
8. Troubleshooting,Maintaining and Repairing PCs,S.J.Bigelow,5th edition,TMH.

I Year, II-Sem (MCA) T 4 C 4

## (F0009152) ORGANISATION STRUCTURE AND PERSONNEL MANAGEMENT

**OBJECTIVES:**

- To know the concept of management and administration.
- To elaborate the organizational structures according to the strategy.
- To understand the personnel management and its functions.
- To identify behavioral models of an individual.

**UNIT I-** Introduction to Management**:** Concepts of Management and organization – nature, importance and functions and theories of management, systems approach to management, leadership styles, social responsibilities of management.

**UNIT II-** Introduction to Organization**:** Designing Organizational structures: Basic concepts related to Organization – Departmentation and Decentralization, types and evaluation of mechanistic and structures of organization and suitability **-** Decision Process Approach **–** Types of decision making - development of corporate strategy, dynamics of decision.

**UNIT III-** Personnel Management**:** Evolution, objectives, personnel policies. Personnel management vs HRM, position of the personnel department in the organization, Role of personnel manager as line manager and staff manager.

**UNIT IV-** Man Power Planning: Need-strategies and limitations, manpower inventory, manpower forecasting, job description, recruitment, job specification and selection, interviewing techniques, transfers and promotion policies.

**UNIT V-** Training and Development: Objectives and policies planning, organizing the training department, training manager and his job, on and off the job training techniques, career planning, objectives of performance appraisal.

**UNIT VI-** Understanding Human Behavior**:** Personality – Johari Window – Transactional Analysis. Perception: Perceptual process, Development of Attitudes and Values, Understanding Group Dynamics, Team Effectiveness, Strategies to deal with conflicts and stress.

**Text Books**:

1. Organisational Behaviour, Robbins:Pearson, 2008.
2. Management and Organizational Behavior, P.Subbarao HPH, 2009.

**References**:

1. Industrial Business Management, Martand T Telsang, S.Chand.
2. Human resources Management, Dr L.M.Prasad, S.Chand.
3. Dynamic personnel Administration, Rudrabasavaraj MN, Himalaya.
4. Personnel Management, Mamoria & Gankar, HPH, 2009.
5. Essentials of Management, Koontz & Weihrich, TMH, 2009.
6. Understanding Organisational Behaviour, Udai Pareek, P.H.I, 2009.

| I Year, II-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0010152) DATA STRUCTURES THROUGH C**

**OBJECTIVES:**

- This course explores fundamental data structures, algorithms for manipulating them, and the practical problems of implementing those structures in real programming languages and environments.

**OUTCOMES:**

After the successful completion of this course, the students should.

Understand and implement the linear data structures and implement the Non-linear data structures.

**UNIT – I**

**Introduction**: Definition, Classification of Data Structures.

Linked list: Definition, Single linked lists, Doubly linked lists, Circular linked lists, Circular Double linked lists, Applications of Linked list: Sparse Matrix Manipulation, Polynomial Representation.

**UNIT – II**

**Stacks:** Introduction, Definition, Representation of Stacks- Arrays and Linked lists, Operations on stacks, Applications of stacks-Evaluation of Arithmetic Expression, Recursion.

**UNIT - III**

**Queues**: Introduction, Definition, Representation of Queues- Arrays and Linked lists, Various Queue structures, Operations on Queues, Applications, Priority queues.

**UNIT - IV**

**Sortin**g: Bubble Sort, Selection Sort, Insertion Sort, Merge Sort, and Quick Sort

**Searching:** Linear Search, Binary Search, and Fibonacci Search.

**UNIT - V**

**Tress:** Binary Tree, Binary Search Tree (BST), Representation and operations on BST, Tree Traversal Techniques: In order, post order, pre order. Applications of BST, AVL trees, Heap, Heap sort.

**UNIT – VI**

**Graphs** –Introduction, Graph Terminologies, Representation of Graphs, Graph traversal techniques-BFS, DFS.

**TEXT BOOKS:**

1. Samanta, Classic Data Structures, 1/e, 2001, PHI.
2. Programming in C and Data Structures, J.R. Hanly, Ashok N. Kamthane, A. Ananda Rao, Pearson Education.
3. An Introduction to Data Structures with Applications, Trembley, Sorenson, 2/e, TMH.
4. C Programming & Data Structures, B.A.Forouzan and R.F. Gilberg, Third Edition, Cengage Learning.
5. Computer Programming and Data Structures, E. Balagurusamy Education / PHI, Eighth Edition.

**REFERENCES:**

1. Programming in C – Stephen G. Kochan, III Edition, Pearson Eductaion.
2. C Programming with problem solving, J.A. Jones & K. Harrow, Dreamtech Press.
3. Data Structures using C – A.M.Tanenbaum, Y.Langsam, and M.J. Augenstein, Pearson.

I Year, II-Sem (MCA) T 4 C 4

**(F0011152) MATHEMATICAL FOUNDATIONS OF COMPUTER SCIENCE**

**OBJECTIVES:**

- To teach students notations used in the discrete mathematics associated with computer science and engineering.
- To teach the rudiments of elementary mathematical reasoning (elementary proofs; proofs by induction).
- To prepare students for the theoretical parts of all further courses in CSE.
- To study logic and Boolean algebra from a mathematical perspective, but relating it to computer engineering applications.
- To introduce basic set-theoretical notions: relations, functions, graphs, equivalence relations and orderings.
- To relate these notions to applications in CSE.

**OUTCOMES:**

- Understand truth tables, the concept of logical equivalence and its relationship to equivalent logic circuits and Boolean functions. Know some Boolean laws of equivalence.
  Extend this to predicate calculus and in predicate calculus using quantifiers.
- Be able to express English assertions in propositional calculus and in predicate calculus using quantifiers.
- Understand and use the basics of set theory notation, Boolean operations on sets Be able to work with functions.
- Be able to carry out simple direct and indirect proofs about domains like the integers and the real numbers, using quantified statements about these domains. Be able to do simple proofs by mathematical induction.
- Be able to understand and write recursive definitions, in mathematical form.
- Understand binary and n-ary relations and their applications. Know the major types of binary relations on a set. Be able to use graphs as representing relations, algorithms for relations based on graphs or matrices (e.g. transitive closure).
- Know the properties of equivalence relations and partial orderings.
- Understand lattices and Boolean algebras as universal algebras.
- Know the fundamentals of counting.

**UNIT-I**

**Mathematical Logic**: Statements and notations, Connectives, Well formed formulas, Truth Tables, tautology, equivalence implication, Normal forms.

**UNIT-II**

**Predicates**: Predicative logic, Free & Bound variables, Rules of inference, Consistency, proof of contradiction, Quantifiers.

**UNIT-III**

**Relations**: Properties of Binary Relations, equivalence, transitive closure, compatibility and partial ordering relations, Hasse diagram, Lattice and its Properties,
**Functions**: Inverse Function, Composition of functions, Pigeonhole principles and its application.

**UNIT-IV**

**Algebraic structures**: Algebraic systems examples and general properties, Semi groups and monads, groups, sub groups, homomorphism, Isomorphism.

**UNIT-V**

Elementary Combinatorics: Basis of counting, Enumerating Combinations & Permutations, with repetitions, Constrained repetitions, Binomial Coefficients, Binomial & Multinomial theorems, the principles of Inclusion – Exclusion.

**UNIT-VI**

**Graph Theory**: Representation of Graph, DFS, BFS, Planar Graphs.
**Graph Theory Applications**: Multi graphs and Euler circuits, Hamiltonian graphs, Chromatic Numbers

**TEXT BOOKS:**

1. Discrete Mathematical Structures with Application to Computer Science, Tremblay, Manohar McGraw Hill Publication (for unit-1 to unit-4).
2. Discrete Mathematics for Computer Scientists & Mathematicians, 2/e, J.L.Mott, A. Kandel, T.P. Baker, PHI (for unit-5 to unit-8).

**REFERENCE:**

1. Discrete and Combinatorial Mathematics- An Applied Introduction, Ralph. P.Grimaldi,5/e, Pearson Education.
2. Mathematical foundation of computer science by Dr.D.S.Chandra sekharaiah, prism publication.

I Year, II-Sem (MCA) T 4 C 4

**(F0012152) OPERATING SYSTEMS**

**OBJECTIVES:**

- This course deals with functions, structures and history of operating systems.
- To understand the design issues associated with operating systems.
- To understand various process management concepts including scheduling, synchronization, deadlocks.
- To be familiar with multithreading and the concepts of memory management including virtual memory.
- To understand the issues related to file system interface and implementation, disk management with protection and security mechanisms.
- Some example operating systems (UNIX, Windows, Solaris etc.)

**OUTCOMES:**

- At the end of the course the students knows the need and requirement of an interface between Man and Machine.
- To enable them to identify the difference between the system software and the application software and their design requirements.
- Students will be able to relate the features of operating systems and the fundamental theory associated with process, memory and file management's components of different operating systems.
- Students will learn about and understand theoretical concepts and programming constructs used for the operation of modern operating systems.
- Students will gain practical experience with software tools available in modern operating systems such as semaphores, system calls, sockets and threads.

**UNIT I**

**Operating System Introduction**, Structures - Simple Batch, Multi programmed, timeshared, Personal Computer, Parallel, Distributed Systems, Real-Time Systems, System components, Operating-System services, System Calls.

**UNIT II**

**Process and CPU Scheduling** - Process concepts and scheduling, Operation on processes, Cooperating Processes, Threads, and Inter process Communication Scheduling Criteria, Scheduling Algorithm, Multiple -Processor Scheduling, Real-Time Scheduling.

**UNIT III**

**Memory Management and Virtual Memory** - Logical versus Physical Address Space, Swapping, Contiguous Allocation, Paging, Segmentation, Performance of Demanding Paging, Page Replacement ,Page Replacement Algorithm, Allocation of Frames, Thrashing.

**UNIT IV**

**Deadlocks** - System Model, Dead locks Characterization, Methods for Handling Dead locks Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, and Recovery from Deadlock.

**UNIT V**

**Process Management and Synchronization**- The Critical Section Problem, Synchronization Hardware, Semaphores, and Classical Problems of Synchronization, Critical Regions, Monitors.

**UNIT VI**

**File System Interface and Implementation** -Access methods, Directory Structure, Protection, File System Structure, Allocation methods, Free-space Management, Directory Management, Directory Implementation, Efficiency and Performance.

**TEXT BOOKS:**

1. Operating System Principles- Abraham Silberchatz, Peter B. Galvin, Greg Gagne 7th Edition, John Wiley.
2. Andrew S.Tanenbaum: Distributed operating system, Prentice Hall International Inc.1995.

**REFERENCE BOOKS:**

1. Operating System A Design Approach-Crowley,TMH.
2. Modern Operating Systems, Andrew S Tanenbaum 2nd edition Pearson/PHI.
3. Operating Systems, Dhamdhere, TMH.

| I Year, II-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0013152) UNIX AND SHELL PROGRAMMING**

**OBJECTIVES:**

Upon successful completion of this course, you should be able to:

- Define and redirect standard files.
- Use the pipe operator to connect two commands.
- Use wildcard met characters and the history command to recall commands.
- Use shell variables.
- Define and use foreground and background processes.
- Define and use processes and co-processes.
- Use directory information in scripts.
- Set and use positional parameters and escape sequences.
- Pass arguments to shell scripts and test arguments.
- Use conditions, control statements and the if command in a decision.
- Use while and for loops.
- Debug scripts using execute trace and verbose trace.
- Use command substitution and Group shell commands.
- Create and use aliases and functions.
- Identify and use signals and handle signals in a shell script.
- Use integer and floating-point arithmetic.
- Create and use variable arrays.
- Perform command evaluation using eval.
- Use here documents, file I/O operators and command options processing.
- Use a lock file to synchronize access.

**OUTCOMES:**

- Be familiar with Unix and Linux operating Systems.
- Master the techniques to use a Linux system.
- Be familiar with the Unix file system and its basic operations.
- Be familiar with the Unix command interpreters.
- Master the techniques of shell programming.

**UNIT I**

**Introduction to Unix:-** Architecture of Unix, Features of Unix , Unix Commands – PATH, man, echo, printf, script, passwd, uname, who, date, stty, pwd, cd, mkdir, rmdir, ls, cp, mv, rm, cat, more, lp, od, tar, gzip.

**UNIT II**

**Unix Utilities:-**Introduction to unix file system, vi editor, file handling utilities, security by file permissions, process utilities: ps, disk utilities:du, df, networking commands:arp, ftp, telnet, rlogin, Unix commands - unlink, find, umask, ulimit, w, finger.

**Unit III:**

**Introduction to Shells :** Unix Session, Standard Streams, Redirection, Pipes, Tee Command, Command Execution, Command-Line Editing, Quotes, Command Substitution, Job Control, Aliases, Variables, Predefined Variables, Options, Shell/Environment Customization.

**UNIT III**

**Filters:** Filters and Pipes, Concatenating files, Display Beginning and End of files, Cut and Paste, Sorting, Translating Characters, Files with Duplicate Lines, Count Characters, Words or Lines, Comparing Files.

**Grep:** Operation, grep Family, Searching for File Content.

**UNIT IV**

**awk:** Execution, Fields and Records, Scripts, Operations, Patterns, Actions, Associative Arrays, String Functions, String Functions, Mathematical Functions, User – Defined Functions, Using System commands in awk, Applications, awk and grep, sed and awk.

**UNIT VI**

**File Management :**File Structures, System Calls for File Management – create, open, close, read, write, lseek, link, symlink, unlink, stat, fstat, lstat, chmod, chown, Directory API – opendir, readdir, closedir, mkdir, rmdir, umask.

**TEXT BOOKS:**

1. Unix and shell Programming Behrouz A. Forouzan, Richard F. Gilberg.Thomson.
2. Your Unix the ultimate guide, Sumitabha Das, TMH. 2nd Edition.

**REFERENCES:**

1. Unix for programmers and users, 3rd edition, Graham Glass, King Ables, Pearson Education.
2. Unix programming environment, Kernighan and Pike, PHI. / Pearson Education.
3. The Complete Reference Unix, Rosen, Host, Klee, Farber, Rosinski, Second Edition, TMH.

I Year, II-Sem (MCA) P 3 C 2

**(F0014152) DATA STRUCTURES THROUGH C LAB**

**Objectives:**

- To make the student learn a programming language.
- To teach the student to write programs in C to solve typical problems.
- To introduce the student to simple linear data structures such as lists, stacks, queues.

**Recommended Systems/Software Requirements:**

- Intel based desktop PC with ANSI C Compiler and Supporting Editors.

**Exercise 1**

Write a C program that uses functions to perform the following operations:
i) Creation ii) Insertion iii) Deletion iv) Traversal on Singly linked list

**Exercise 2**

Write a C program that uses functions to perform the following operations:
i) Creation ii) Insertion iii) Deletion iv) Traversal on Doubly linked list

**Exercise 3**

Write a C program that uses functions to perform the following operations:
i) Creation ii) Insertion iii) Deletion iv) Traversal on Circular linked list

**Exercise 4**

**a)** Write C programs that implement stack (its operations) using
i) Arrays ii) Pointers
**b)** Write C programs that implement Queue (its operations) using
i) Arrays ii) Pointers

**Exercise 5**

Write a C program that uses Stack operations to perform the following:
i) Converting infix expression into postfix expression
ii) Evaluating the postfix expression

**Exercise 6**

Write a C program that implements the following sorting methods to sort a given list of integers in ascending order
i) Bubble sort
ii) Selection sort
iii) Insertion sort

**Exercise 7**

Write C program that implements the following sorting method to sort a given list of integers in ascending order:
i)Quick sort
ii)Merge sort

**Exercise 8**

Write C programs that use both recursive and non recursive functions to perform the following searching operations for a Key value in a given list of integers :
i) Linear search ii) Binary search

**Exercise 9**

Write C programs to create BST and perform operations on it.

Write C programs to implement recursive and non recursive Tree traversal techniques.

**Exercise 10**

Write C programs that implement

i) BSF

ii) DSF

**REFERENCES:**

1. The Spirit of C, an introduction to modern programming, M.Cooper, Jaico Publishing House.
2. Mastering C, K.R. Venugopal and S.R. Prasad, TMH Publications.
3. Computer Basics and C Programming, V. Rajaraman, PHI Publications

I Year, II-Sem (MCA) P 3 C 2

**(F0015152) UNIX AND SHELL PROGRAMMING LAB**

**Objectives:**

- To teach students various unix utilities and shell scripting.

**Recommended Systems/Software Requirements:**

- Intel based desktop PC with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space LAN Connected.
- Any flavour of Unix / Linux.

**Week1**

**Session-1**

a) Log into the system
b) Use vi editor to create a file called myfile.txt which contains some text.
c) correct typing errors during creation.
d) Save the file
e) logout of the system

**Session-2**

a) Log into the system
b) Open the file created in session 1
c) Add some text
d) Change some text
e) Delete some text
f) Save the Changes
g) Logout of the system

**Week2**

a) Log into the system.
b) Use the cat command to create a file containing the following data. Call it mytable use tabs to separate the fields.

| | | |
|---|---|---|
| 1425 | Ravi | 15.65 |
| 4320 | Ramu | 26.27 |
| 6830 | Sita | 36.15 |
| 1450 | Raju | 21.86 |

c) Use the cat command to display the file, mytable.
d) Use the vi command to correct any errors in the file, mytable.
e) Use the sort command to sort the file mytable according to the first field. Call the sorted file my table (same name)
f) Print the file mytable
g) Use the cut and paste commands to swap fields 2 and 3 of mytable. Call it my table (same name)
h) Print the new file, mytable
i) Logout of the system.

**Week3**

1) a) Login to the system.
   b) Use the appropriate command to determine your login shell.
   c) Use the /etc/passwd file to verify the result of step b.
   d) Use the who command and redirect the result to a file called myfile1. Use the more command to see the contents of myfile1.
   e) Use the date and who commands in sequence (in one line) such that the output of date will display on the screen and the output of who will be redirected to a file called myfile2. Use the more command to check the contents of myfile2.

2. a) Write a sed command that deletes the first character in each line in a file.
   b) Write a sed command that deletes the character before the last character in each line in a file.
   c) Write a sed command that swaps the first and second words in each line in a file.

**Week4**

a) Pipe your /etc/passwd file to awk, and print out the home directory of each user.
b) Develop an interactive grep script that asks for a word and a file name and then tells how many lines contain that word.
c) Repeat
d) Part using awk

**Week5**

a) Write a shell script that takes a command –line argument and reports on whether it is directory, a file, or something else.
b) Write a shell script that accepts one or more file name as arguments and converts all of them to uppercase, provided they exist in the current directory.
c) Write a shell script that determines the period for which a specified user is working on the system.

**Week6**

a) Write a shell script that accepts a file name starting and ending line numbers as arguments and displays all the lines between the given line numbers.
b) Write a shell script that deletes all lines containing a specified word in one or more files supplied as arguments to it.

**Week7**

a) Write a shell script that computes the gross salary of a employee according to the following rules:
   i)If basic salary is < 1500 then HRA =10% of the basic and DA =90% of the basic.
   ii)If basic salary is >=1500 then HRA =Rs500 and DA=98% of the basic
   The basic salary is entered interactively through the key board.
b) Write a shell script that accepts two integers as its arguments and computers the value of first number raised to the power of the second number.

**Week8**

a) Write an interactive file-handling shell program. Let it offer the user the choice of copying, removing, renaming, or linking files. Once the user has made a choice, have the program ask the user for the necessary information, such as the file name, new name and so on.
b) Write shell script that takes a login name as command – line argument and reports when that person logs in

c) Write a shell script which receives two file names as arguments. It should check whether the two file contents are same or not. If they are same then second file should be deleted.

**Week9**

a) Write a shell script that displays a list of all the files in the current directory to which the user has read, write and execute permissions.
b) Develop an interactive script that ask for a word and a file name and then tells how many times that word occurred in the file.
c) Write a shell script to perform the following string operations:
i)To extract a sub-string from a given string.
ii)To find the length of a given string.

**Week10**

Write a C program that takes one or more file or directory names as command line input and reports the following information on the file:

i)File type
ii)Number of links
iii)Read, write and execute permissions
iv)Time of last access

(Note: Use stat/fstat system calls)

**Week11**

Write C programs that simulate the following unix commands:

a) mv
b) cp

(Use system calls)

**Week12**

Write a C program that simulates ls Command
(Use system calls / directory API)

**TEXT BOOKS**

1. Introduction to UNIX & SHELL programming, M.G. Venkatesh Murthy, Pearson Education.
2. Unix concepts and applications, Fourth Edition, Sumitabha Das, TMH.
3. Unix for programmers and users, 3rd edition, Gaham Glass & K. Ables, pearson education.
4. Unix and shell Programming –A text book, B.A. Forouzan & R.F. Giberg, Thomson.
5. Beginning shell scripting, E. Foster – Johnson & other, Wile Y- India.

I Year, II-Sem (MCA) P C
3 2

**(F0016152) OPERATING SYSTEMS LAB**

**OBJECTIVES:**

A student will be able to:

- Prepare students for easy transfer from academia into practical life.
- Get an Insight into the Computer Technologies.
- Obtain Basic Knowledge of Operating Systems.

**OUTCOMES:**

The main learning outcomes are:

- Master functions, structures and history of operating systems.
- Master understanding of design issues associated with operating systems.
- Master various process management concepts including scheduling, synchronization, deadlocks.
- Be familiar with multithreading.
- Master concepts of memory management including virtual memory.
- Master system resources sharing among the users.
- Master issues related to file system interface and implementation, disk management.
- Be familiar with protection and security mechanisms.
- Be familiar with various types of operating systems including Unix.

**List of Sample Problems/Experiments:**

1. Simulate the following CPU scheduling algorithms
   a) Round Robin b) SJF c) FCFS d) Priority

2. Simulate all file allocation strategies
   a) Sequential b) Indexed c) Linked

3. Simulate MVT and MFT

4. Simulate all File Organization Techniques
   a) Single level directory b) Two level
5. Simulate Bankers Algorithm for Dead Lock Avoidance

6. Write a C program to create a child process and allow the parent to display "Hello" and the child to display "Welcome" on the screen.

7. Simulate all page replacement algorithms such as
   a) FIFO b) LRU c) Optimal

8. Simulate Paging Technique of memory management.

9. Write C programs that make a copy of a file using i) standard I/O and ii) system calls.

10. Write C programs that count the number of blanks in a text file using i) Standard I/O and ii) system calls.

**REFERENCES:**

1. Operating Systems, P.P. Choudhury, PHI Learning Private Ltd.
2. Operating Systems, R.Chopra, S.Chand and Company Ltd.

II Year, I-Sem (MCA) T C
4 4

**(F0017153) SOFTWARE ENGINEERING**

**OBJECTIVES:**

- To understand Software development as a process.
- Various software process models and system models.
- Various software designs: Architectural, object oriented, user interface etc.
- Software testing methodologies overview: various testing techniques including white box testing black box testing regression testing etc.
- Software quality: metrics, risk management quality assurance etc.

**OUTCOMES:**

- Students will learn to work as a team and to focus on getting working project done on time with each student being held accountable for their part of the project.
- Student will learn about risk management and quick prototyping de-risk project management.
- Students will learn about and go through the software life cycle with emphasis on different process requirements design and implementation phases.
- Students will learn about software process models and hoe to choose an appropriate model for their project will learn about risk management and quick prototyping to de-risk projects.
- Students will gain confidence and conceptualized.

**UNIT I**

**Introduction to Software Engineering:** The evolving role of software, Changing Nature of Software, Software myths.

**A Generic view of process:** Software engineering- A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process patterns.

**UNIT II**

**Process models:** The waterfall model, Incremental process models, Evolutionary process models, The Unified process.

**System models:** Context Models, Behavioral models, Object models.

**UNIT III**

**Requirements engineering process:** Feasibility studies, Requirements elicitation and analysis, Requirements validation, Requirements management.

**Software Requirements:** Functional and non-functional requirements, User requirements, System requirements, the software requirements document.

**UNIT IV**

**Design Engineering:** Design process and Design quality, Design concepts.

**Creating an architectural design:** Software architecture, Data design, Architectural styles and patterns, Architectural Design.

**UNIT V**

**Performing User interface design**: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

**Testing Strategies:** A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, System testing.

**UNIT VI**

**Risk management:** Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

**Quality Management :** Quality concepts, Software quality assurance, Software Reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

**TEXT BOOKS**

1. Software Engineering, A practitioner's Approach- Roger S. Pressman, 6th edition.McGrawHill International Edition.
2. Software Engineering- Sommerville, 7th edition, Pearson education.

**REFERENCES**

1. Software Engineering- K.K. Agarwal & Yogesh Singh, New Age International Publishers
2. Software Engineering, an Engineering approach- James F. Peters, Witold Pedrycz, John Wiely.
3. Systems Analysis and Design- Shely Cashman Rosenblatt,Thomson Publications.
4. Software Engineering principles and practice- Waman S Jawadekar, The McGraw-Hill Companies.

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0018153) DATABASE MANAGEMENT SYSTEMS**

**OBJECTIVES:**

- Advantages applications of DBMS and Database system structure.
- Schema design: ER model and conceptual design.
- Relational model and SQL basics.
- Relational algebra and Query optimization.
- Storage and efficient retrieval of data: various indexing techniques.
- Schema refinement: normalization and redundancy removal and functional dependent.
- Transaction management: locking protocols, serializability concepts etc.
- Concurrency control and crash recovery: various mechanisms, ARIES algorithm and deadlock concepts.

**OUTCOMES:**

- Students will learn about the need for DBMS, the largeness of the data and why it gives rise to steam oriented processing and strategies and are at higher level than general purpose programming language such as JAVA.
- Students will learn about storage and efficient retrieval of large Information via algebraic query optimization and the use of indexing.
- Students will also learn basics of SQL and about primary key concepts and foreign key concepts. They will also learn about data manipulation (insertions deletions & updation) and triggers.
- Students will learn about functional dependency and the need for schema refinement (normalization) to remove redundancy of data.
- Students will also learn about transaction management concurrency Control and crash recovery.

**UNIT I**

Data base System Applications, data base System VS file System – View of Data – Data Abstraction –Instances and Schemas – data Models – the ER Model – Relational Model – Other Models – Database Languages – DDL – DML – database Access for applications Programs – data base Users and Administrator – Transaction Management – data base System Structure – Storage Manager – the Query Processor.

**UNIT II**

History of Data base Systems. Data base design and ER diagrams – Beyond ER Design Entities, Attributes and Entity sets – Relationships and Relationship sets – Additional features of ER Model – Concept Design with the ER Model – Conceptual Design for Large enterprises.

**UNIT III**

Introduction to the Relational Model – Integrity Constraint Over relations – Enforcing Integrity constraints – Querying relational data – Logical data base Design – Introduction to Views – Destroying /altering Tables and Views.
Relational Algebra – Selection and projection set operations – renaming – Joins – Division – Examples of Algebra overviews – Relational calculus – Tuple relational Calculus – Domain relational calculus – Expressive Power of Algebra and calculus.

**UNIT IV**

Form of Basic SQL Query – Examples of Basic SQL Queries – Introduction to Nested Queries – Correlated Nested Queries Set – Comparison Operators – Aggregative Operators – NULL values – Comparison using Null values – Logical connectivity‘s – AND, OR and NOT –

Impact on SQL Constructs – Outer Joins – Disallowing NULL values – Complex Integrity Constraints in SQL Triggers and Active Data bases.

**UNIT V**

Schema refinement – Problems Caused by redundancy – Decompositions – Problem related to decomposition – reasoning about FDS – FIRST, SECOND, THIRD Normal forms – BCNF – Lossless join Decomposition – Dependency preserving Decomposition – Schema refinement in Data base Design – Multi valued Dependencies – FORTH Normal Form.

**UNIT VI**

**Overview of Transaction Management:** ACID Properties – Transactions and Schedules– Concurrent Execution of transaction – Lock Based Concurrency Control – Performance Locking – Transaction Support in SQL – Introduction to Crash recovery.

**Concurrency Control:** Serializability, and recoverability – Introduction to Lock Management – Lock Conversions – Dealing with DeadLocks – Specialized Locking Techniques – Concurrency without Locking.

**Crash recovery:** Introduction to ARIES – the Log – Other Recovery related Structures – the Write-Ahead Log Protocol – Check pointing – recovering from a System Crash –Media recovery – Other approaches and Interaction with Concurrency control.

**TEXT BOOKS:**

1. Data base Management Systems, Raghurama Krishnan, Johannes Gehrke, TATAMcGrawHill 3rd Edition
2. Data base System Concepts, Silberschatz, Korth, McGraw hill, V edition.

**REFERENCES:**

1. Data base Systems design, Implementation, and Management, Peter Rob & Carlos Coronel 7th Edition.
2. Fundamentals of Database Systems, ElmasriNavrate Pearson Education
3. Introduction to Database Systems, C.J.Date Pearson Education

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0019153) COMPUTER NETWORKS**

**Objectives:**

- An understanding of the overriding principles of computer networking, including protocol
  design, protocol layering, algorithm design, and performance evaluation.
- An understanding of computer networking theory, including principles embodied in the
  protocols designed for the application layer, transport layer, network layer, and link layer of a networking stack.
- An understanding of specific implemented protocols covering the application layer, transport layer, network layer, and link layer of the Internet (TCP/IP) stack.
- An understanding of security issues.

**Outcomes:**

- Students will learn to list and classify network services, protocols and architectures, explain why they are layered.
- Student will learn to explain key Internet applications and their protocols.
- Students will learn to explain security issues in computer networks.
- To master the terminology and concepts of the OSI reference model and the TCP-IP reference model.
- To master the concepts of protocols, network interfaces, and Design/performance issues in local area networks and wide area networks.
- To be familiar with wireless networking concepts.
- To be familiar with contemporary issues in networking technologies.
- To be familiar with network tools and network programming.

**UNIT – I**
**Introduction:** Uses of Computer Networks, Networks Hardware and Software, OSI, TCP/IP, Examples of Networks: Novell Networks, Arpanet, Internet, Network Topologies WAN, LAN, MAN.

**UNIT - II**
**Physical Layer:** Transmission media copper, twisted pair wireless, Co axial cable, Narrow band, broad band ISDN and ATM.

**UNIT - III**
**Data link layer:** Design issues, framing, error detection and correction, CRC, Elementary Protocol-stop and wait, Sliding Window, Data link layer in HDLC, Internet.

**UNIT - IV**
**Network Layer:** Virtual circuit and Datagram subnets-Routing algorithm shortest path routing, Flooding, Hierarchical routing, Broad cast, Multi cast, distance vector routing.

**UNIT – V**
**Dynamic Routing** – Congestion, Control Algorithms – General Principles – of Congestion prevention policies. Internetworking: The Network layer in the internet.
**Transport Layer:** Transport Services, Connection management, TCP and UDP protocolsl.

**UNIT – VI**
**Application Layer** – DNS, WWW, E-mail, Multimedia.
**Network Security-** Cryptography, Public key Algorithm.

**TEXT BOOKS:**

1. Computer Networks — Andrew S Tanenbaum, 4th Edition. Pearson Education/PHI
2. Data Communications and Networking – Behrouz A. Forouzan.Third Edition TMH.

**REFERENCES:**

1. An Engineering Approach to Computer Networks-S.Keshav, 2nd Edition, Pearson Education
2. Understanding communications and Networks, 3rd Edition, W.A. Shay, Thomson

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0020153) OBJECT ORIENTED PROGRAMING USING C++**

**Objectives:**

- To get a clear understanding of object-oriented concepts.
- To understand object oriented programming through C++.

**Course Outcome:**

- Gain the basic knowledge on Object Oriented concepts.
- Ability to develop applications using Object Oriented Programming Concepts.
- Ability to implement features of object oriented programming to solve real world problems.

**UNIT I**

**Introduction:** Different paradigms for problem solving need for OOP paradigm, fundamental characteristics of OOP (Alan key), differences between OOP and Procedure Oriented Programming.

**C++ Basics:** Structure of a C++ program, Data types, Declaration of variables, Expressions, Operators, Operator Precedence, Evaluation of expressions, Type conversions, Pointers, Arrays, Pointers and Arrays, Strings, Structures, References. Flow control statements- if, switch, while, for, do, break, continue, goto statements.

**UNIT II**

**C++ Functions:** Scope of variables, Parameter passing methods, Default arguments, inline functions, Recursive functions, Pointers to functions.

**C++ Classes And Data Abstraction:** Class definition, Class objects, Class scope, this pointer, Friends to a class, Static class members, Constant member functions, Constructors and Destructors, Data abstraction, ADT and information hiding.

**UNIT III**

**Dynamic memory allocation and de-allocation operators:** new and delete, Dynamic creation and destruction of objects, Preprocessor directives.

**Polymorphism:** Function overloading, Operator overloading.

**Generic programming:** necessity of templates, Function templates and class templates.

**UNIT IV**

**Inheritance:** Defining a class hierarchy, Different forms of inheritance, Defining the Base and Derived classes, Access to the base class members, Base and Derived class construction, Destructors, Virtual base class.

**UNIT V**

**Virtual Functions and Run Time Polymorphism:** Overriding, Static and Dynamic bindings, Base and Derived class virtual functions, Dynamic binding through virtual functions, Virtual function call mechanism, Pure virtual functions, Abstract classes.

**UNIT VI**

**C++ I/O:** I/O using C functions, C++ Stream classes hierarchy, Stream I/O, File streams and String streams, File Operations, Overloading << and >> operators, Error handling during file operations, Formatted I/O.

**Exception Handling:** Benefits of exception handling, Throwing an exception, The try block, Catching an exception, Exception objects, Exception specifications, Re-throwing an exception, Catching all exceptions.

**TEXT BOOKS:**

1. Object Oriented Programming in C++, 4th Edition, R.Lafore, SAMS, Pearson Education
2. Object Oriented Programming with c++,Balaguruswamy.

**REFERENCES:**

1. An Introduction to OOP, 3rd Edition, T. Budd, Pearson Education, 2008.

2. Programming Principles and Practice Using C++, B.Stroutstrup, Addison- Wesley, Pearson Education.
3. Problem solving with C++, 6th Edition, Walter Savitch, Pearson Education, 2007.
4. OOP in C++, 3rd Edition, T.Gaddis, J.Walters and G.Muganda, Wiley DreamTech Press.
5. C++, The Complete Reference, 4th Edition, Herbert Schildt, TMH.

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0021153) OPERATIONS RESEARCH**

**Objectives:**

To help the students in getting thorough understanding of the fundamentals of operations research and usage of linear programming problems, transportation problems, assignment models, game theory & queuing theory.

**Outcomes:**

By the end of module students will be expected to demonstrate proper understanding of the concepts of Operation Research and use these to solve the problems in Industry.

**UNIT I**

Introduction to Operations Research: Basics definition, Linear programming problem – Formulation of LPP, Graphical solution of Lpp, Simplex method, Artificial variables, big – M method, two – phase method.

**UNIT II**

Transportation problem – Formulation, Finding basic feasible solutions – Northwest corner rule, least cost method and vogel's approximation method, optimality test – stepping stone method. Assignment model – Formulation, Hungarian method for optimal solution, solving unbalanced, maximization, travelling salesman problem.

**UNIT III**

Sequencing models, solution of sequencing problem – processing n jobs through 2 machines – processing n jobs through 3 machines – processing n jobs through m machines- processing 2 jobs through m machines.

**UNIT IV**

Games Theory, competitive games, rectangular game, saddle poiont, minimax (maximin) Method of optimal strategies, value of the game, solution of games with saddle points, dominance principle, rectangular games without saddle point – mixed strategy for 2X2 games.

**Unit –V:**

Statistical quality control: Concept of quality of a manufactured product –Defects and Defectives – causes of variations – Random and assignable – The principle of Shewhart control chart-Charts for attribute and variable quality characteristics-Constructions and operation of -Chart, R-Chart, P-chart and C-chart.

**Unit – VI**

Queuing theory: Pure birth and Death process, M/M/1 and M/M/S and their related simple Problems.

**TEXT BOOKS:**

1. Operations Research, P.K. Gupta and D.S. Hira, S. Chand & co., 2007.
2. Operation Reasearch by S.D.Sarma, Kedharnath Publications.

**REFERENCES:**

1. Operations Research, A.M. Natarajan, P.Balasubramani, A. Tamilarasi, Pearson Education, 2005.
2. Operations Research, R. Paneerselvam, 2/e, PHI 2008.
3. Operations research, P SankaraIyer, Tata McGraw Hill, 2008.

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 3 | 2 |

**(F0022153) OBJECT ORIENTED PROGRAMING USING C++ LAB**

**Objectives:**

• To get a clear understanding of object-oriented concepts.
• To understand object oriented programming through C++.

**Course Outcome:**

• Gain the basic knowledge on Object Oriented concepts.
• Ability to develop applications using Object Oriented Programming Concepts.
• Ability to implement features of object oriented programming to solve real world problems.

**List of Sample Problems/Experiments:**

1. Write a C++ program to find the sum of individual digits of a positive integer.

2. A Fibonacci sequence is defined as follows: the first and second terms in the sequence are 0

and 1. Subsequent terms are found by adding the preceding two terms in the sequence. Write

a C++ program to generate the first n terms of the sequence.

3. Write a C++ program to generate all the prime numbers between 1 and n ,

where n is a value supplied by the user.

4. Write C++ programs that use both recursive and non-recursive functions

a. To find the factorial of a given integer.

b. To find the GCD of two given integers.

c. To find the nth Fibonacci number.

5. Write a C++ program that uses a recursive function for solving Towers of Hanoi problem.

6. Write a C++ program that uses functions

a. To swap two integers.

b. To swap two characters.

c. To swap two reals. Note: Use overloaded functions.

7. Write a C++ program to find both the largest and smallest number in a list of integers.

8. Write a C++ program to sort a list of numbers in ascending order.

9. Write a C++ program that uses function templates to solve problems-7&8.

10. Write a C++ program to sort a list of names in ascending order.

11. Write a C++ program to implement the matrix ADT using a class. The operations supported

by this ADT are:

a) Reading a matrix. b) Printing a matrix. c) Addition of matrices.

d) Subtraction of matrices. e)Multiplication of matrices.

12. Implement the matrix ADT presented in the problem-11 using overloaded operators

(<<, >>, +, -, *) and templates.

13. Implement the complex number ADT in C++ using a class. The complex ADT is used to

represent complex numbers of the form c=a+ib, where a and b are real numbers. The operations supported by this ADT are:

a) Reading a complex number.
b) Writing a complex number.
c) Addition of Complex numbers.
d) Subtraction of complex numbers.
e) Multiplication of complex numbers.
f) Division of complex numbers.

14. Write a C++ program that overloads the + operator and relational operators (suitable) to perform

the following operations:

a) Concatenation of two strings.

b) Comparison of two strings.

15. Implement the complex number ADT in C++ using a class. The complex ADT is used to represent

complex numbers of the form c=a+ib, where a and b are real numbers. The operations supported by this ADT are:

a) Reading a complex number.
b) Writing a complex number.
c) Addition of Complex numbers.
d) Subtraction of complex numbers.
e) Multiplication of complex numbers.
f) Division of complex numbers.

Note: 1. overload << and >> operators in part a and part b.

2. overload +, - , * , / operators in parts c, d, e and f.

16. Write a template based C++ program that determines if a particular value occurs in an array

of values.

17. Write a C++ program that uses functions to perform the following operations:

a. Insert a sub-string into the given main string from a given position.

b. Delete n characters from a given position in a given string.

18. Write a C++ program that uses a function to reverse the given character string in place, without

any duplication of characters.

19. Write a C++ program to make the frequency count of letters in a given text.

20. Write a C++ program to count the lines, words and characters in a given text.

21. Write a C++ program to determine if the given string is a palindrome or not.

22. Write a C++ program to make frequency count of words in a given text.

23. Write a C++ program that displays the position or index in the string S where the string t

begins, or –1 if S doesn't contain t.

24. 2's complement of a number is obtained by scanning it from right to left and complementing

all the bits after the first appearance of a 1. Thus 2's complement of 11100 is 00100. Write a

C++ program to find the 2's complement of a binary number.

25. Write a C++ program that counts the number of 1 bit in a given integer.

26. Write a C++ program to generate Pascal's triangle.

27. Write a C++ program to construct of pyramid of numbers.

28. Write a C++ program to compute the Sine series.

29. Write a C++ program to display the contents of a text file.

30. Write a C++ program which copies one file to another.

31. Write a C++ program that counts the characters, lines and words in the text file.

32. Write a C++ program to change a specific character in a file.

Note: Filename, number of the byte in the file to be changed and the new character are specified on the command line.

33. Write a C++ program to reverse the first n characters in a file.

34. Write a C++ program that uses a function to delete all duplicate characters in the given string.

35. Write a C++ program that uses a function to convert a number to a character string.

36. Write a C++ program that uses a recursive function to find the binary equivalent of a given non-negative integer n.

37. Write a C++ program to generate prime numbers up to n using Sieve of Eratosthenes method.

38. Write a C++ program

a) To write an object to a file.

b) To read an object from the file

39. Write C++ programs that illustrate how the following forms of inheritance are supported:

a) Single inheritance b) Multiple inheritance

c) Multi level inheritance d) Hierarchical inheritance

40. Write a C++ program that illustrates the order of execution of constructors and destructors when new class is derived from more than one base class.

41. Write a C++ program that illustrates how run time polymorphism is achieved using virtual functions.

42. Write a C++ program that illustrates the role of virtual base class in building class hierarchy.

43. Write a C++ program that illustrates the role of abstract class in building class hierarchy.

REFERENCES:

1. Mastering C++, K.R.VenuGopal, Raj Kumar and T.Ravi Shankar, TMH.
2. C++ Programming, D.S.Malik, Cengage Learning.
3. Practical C++ Programming, S.Qualline,SPD.
4. Object Oriented Programming with C++, E.Balaguruswamy, 4th Edition, TMH, 2008.
5. OOP with C++, S.Sahay,Oxford Higher Education.
6. C++ and OOP Paradigm, D.Jana,2nd Edition, PHI
7. Fundamentals of C++ Programming,S.Subramanian,Jaico Publishing House.
8. C++ Programming, Al Stevens, 7thedition, Wiley India.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| II Year, I-Sem (MCA) | P | C |
|---|---|---|
| | 3 | 2 |

**(F0023153) DATA BASE MANAGEMENT SYSTEMS LAB**

**OBJECTIVES**

- Student will be able to:
- Create and delete database schemas and execute SQL queries
- Inserting data, Altering and dropping the tables.
- Various types of data conversions using the functions.
- Make Use of PL/SQL Language Components.
- Make Use of PL/SQL Variables.
- Handle PL/SQL Reserved Words.
- Make Use of Identifiers in PL/SQL
- Make Use of Anchored Data type

**OUTCOMES**

Upon completion of the lab, the student should be able to:

- Map the model into a relational database system.
- Implement the given schema on a relational DBMS.
- Design, develop, and maintain Oracle Database Objects.
- Use a database language for manipulating and querying data.
- Develop advanced packages, stored procedures, and triggers and functions using PL/SQL

**Recommended Systems/Software Requirements:**

- Intel based desktop PC
- Mysql /Oracle latest version Recommended.

1) Creation, altering and droping of tables and inserting rows into a table (use constraints while creating tables) examples using SELECT command.
2) Queries (along with sub Queries) using ANY, ALL, IN, EXISTS, NOTEXISTS, UNION, INTERSET, Constraints.

   Example:- Select the roll number and name of the student who secured fourth rank in the class.
3) Queries using Aggregate functions (COUNT, SUM, AVG, MAX and MIN), GROUP BY, HAVING and Creation and dropping of Views.
4) Queries using Conversion functions (to_char, to_number and to_date), string functions (Concatenation, lpad, rpad, ltrim, rtrim, lower, upper, initcap, length, substr and instr), date functions (Sysdate, next_day, add_months, last_day, months_between, least, greatest, trunc, round, to_char, to_date)
5) i)Creation of simple PL/SQL program which includes declaration section, executable section and exception –Handling section (Ex. Student marks can be selected from the table and printed for those who secured first class and an exception can be raised if no records were found)

   ii)Insert data into student table and use COMMIT, ROLLBACK and SAVEPOINT in PL/SQL block.
6) Develop a program that includes the features NESTED IF, CASE and CASE expression. The program can be extended using the NULLIF and COALESCE functions.
7) Program development using WHILE LOOPS, numeric FOR LOOPS, nested loops using ERROR Handling, BUILT –IN Exceptions, USE defined Exceptions, RAISE-APPLICATION ERROR.
8) Programs development using creation of procedures, passing parameters IN and OUT of PROCEDURES.

9) Program development using creation of stored functions, invoke functions in SQL Statements and write complex functions.
10) Program development using creation of package specification, package bodies, private objects, package variables and cursors and calling stored packages.
11) Develop programs using features parameters in a CURSOR, FOR UPDATE CURSOR, WHERE CURRENT of clause and CURSOR variables.
12) Develop Programs using BEFORE and AFTER Triggers, Row and Statement Triggers and INSTEAD OF Triggers

**TEXT BOOKS:**

1. ORACLE PL/SQL by example. Benjamin Rosenzweig, Elena Silvestrova, Pearson Education 3rd Edition
2. ORACLE DATA BASE LOG PL/SQL Programming SCOTT URMAN, Tata Mc- Graw Hill.
3. SQL & PL/SQL for Oracle 10g, Black Book, Dr.P.S. Deshpande.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| II Year, I-Sem (MCA) | T | C |
|---|---|---|
| | 3 | 2 |

**(F0024153) COMPUTER NETWORKS LAB**

**Objective:**

- To Understand the functionalities of various layers of OSI model
- To understand the operating System functionalities

**System/ Software Requirement**

- Intel based desktop PCs LAN CONNECTED with minimum of 166 MHZ or faster processor with atleast 64 MB RAM and 100 MB free disk space

1. Implement the data link layer framing methods such as character, character stuffing and bit stuffing.
2. Implement on a data set of characters the three CRC polynomials – CRC 12, CRC 16 and CRC CCIP.
3. Implement Dijkstra‘s algorithm to compute the Shortest path thru a graph.
4. Take an example subnet graph with weights indicating delay between nodes. Now obtain Routing table art each node using distance vector routing algorithm.
5. Take a 64 bit playing text and encrypt the same using DES algorithm.
6. Write a program to break the above DES coding.
7. Using RSA algorithm Encrypt a text data and Decrypt the same

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| II Year, II-Sem (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0025154) Object Oriented Programming Through Java**

**OBJECTIVES:**

After taking this course, the student should be able to:

- Describe the Windows event-driven programming model
- Build simple JAVA applications according to the model
- Write fluent JAVA code for creating classes
- Use JAVA variables, data, expressions and arrays
- Design and create forms, menus and controls
- Write clear, elementary Java programs (applets and applications)
- Use a Java-enabled browser and/or the applet viewer to execute Java applets
- Use the Java interpreter to run Java applications
- Design and construct effective graphic user interfaces for application software.
- Use Java Beans, RMI to build complex business applications

**OUTCOMES:**

- Understand the syntax and concepts of JAVA
- Write JAVA programs for processing data
- Write JAVA programs to interface with windows.
- Write JAVA programs that use data from flat files and databases.
- Develop programs with GUI features such as dialog boxes, menus etc.
- Write JAVA programs that form the GUI front-end for database applications.
- Write applications using distributed objects.
- A passing student shall demonstrate knowledge of GUI-based event-driven programming in a working.
- Program assignment utilizing Java GUI components, event listeners and event-handlers.

**UNIT-I**

**Introduction**: Creation of Java, importance of Java to internet, byte code, Java buzzwords, OOP Principles, Encapsulation, Inheritance and Polymorphism, data types, variables, declaring variables, dynamic initialization, scope and life time of variables, arrays, operators, control statements, type conversion and casting, compiling and running of simple Java program. **Strings:** Strings, string functions.

**UNIT-II**

**Classes and Objects**: Introducing class, methods, constructors, usage of static with data and methods, usage of final with data, access control, this key word, garbage collection, overloading methods and constructors, parameter passing – call by value, recursion

Inheritance: Basic concepts, member access rules, usage of super key word, forms of inheritance, method overriding, abstract classes, dynamic method dispatch, using final with inheritance, the Object class.

**UNIT-III**

**Packages and Interfaces**: Defining, Creating and Accessing a Package, Understanding CLASSPATH, importing packages, differences between classes and interfaces, defining an interface, implementing interface, applying interfaces, variables in interface and extending interfaces.

**UNIT-IV**

**Exception Handling and Multithreading**: Concepts of Exception handling, types of exceptions, usage of try, catch, throw, throws and finally keywords, Built-in exceptions, creating own exception sub classes, Concepts of Multithreading, differences between process and thread, thread life cycle, creating multiple threads using Thread class, Runnable interface, Synchronization, thread priorities, inter thread communication, daemon threads, deadlocks, thread groups.

**UNIT-V**

**Applets** – Concepts of Applets, differences between applets and applications, life cycle of an applet, types of applets, creating applets, passing parameters to applets.

**Event Handling**: Events, Event sources, Event classes, Event Listeners, Delegation event model, handling mouse and keyboard events, Adapter classes.

**UNIT-VI**

**AWT:** Concepts of components, container, panel, window, frame, canvas, Font class, Color class and Graphics, Controls: Buttons, Labels, Text fields, Text area, Check boxes, Checkbox groups, Lists, Choice, Scrollbars, Menus, Layout Managers – Flow, Border, Grid and Card.

**TEXT BOOKS:**

1. The Complete Reference Java J2SE 5th Edition, Herbert Schildt, TMH Publishing Company Ltd, NewDelhi.
2. Programming with java primer 3rd Edition, E.Balaguruswamy, McGraHill.

**REFERENCES:**

1. Java How to Program, Sixth Edition, H.M.Dietel and P.J.Dietel, Pearson Education/PHI.
2. Core Java 2, Vol 1, Fundamentals, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.
3. Core Java 2, Vol 2, Advanced Features, Cay.S.Horstmann and Gary Cornell, Seventh Edition, Pearson Education.
4. Beginning in Java 2, Iver Horton, Wrox Publications.

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0026154) OBJECT ORIENTED ANALYSIS AND DESIGN USING UML**

**OBJECTIVES:**

- Building block of UML: things, relationships and diagrams.
- Contents and common modeling techniques to model static aspects of any system using.
- Class diagrams, object diagrams, deployment diagram, use case diagram.
- Contents and common modeling techniques to model dynamic aspects of any system using.
- Sequence diagram, collaboration diagram, activity diagram, state chart diagram.
- Contents and common modeling techniques to model any system using component diagram to implement physically.

**OUTCOMES:**

- Analyze system requirements and model problem domains.
- Evaluate quality of analysis.
- Design and build object oriented systems.
- Explain and justify designs based on design principles patterns and heuristics.
- Write object-oriented code to correctly implement a design.
- Be able to read and write analysis and design documentation the Unified Modeling language.

Be able to read and write object-oriented code.

**UNIT – I:**

**Introduction to UML:** Importance of modeling, principles of modeling, object oriented modeling, conceptual model of the UML, Architecture, Software Development Life Cycle.

**UNIT – II:**

**Basic Structural Modeling:** Classes, Relationships, common Mechanisms, and diagrams.
**Advanced Structural Modeling:** Advanced classes, advanced relationships, Packages.

**UNIT – III:**

**Class & Object Diagrams:** Terms, concepts, modeling techniques for Class & Object Diagrams.

**UNIT- IV:**

**Basic Behavioral Modeling-I:** Interactions, Interaction diagrams.
**Basic Behavioral Modeling-II:** Use cases, Use case Diagrams, Activity Diagrams.

**UNIT – V:**

**Advanced Behavioral Modeling:** Events and signals, state machines.
**Architectural Modeling:** Component, Deployment, Component diagrams and Deployment diagrams.

**UNIT – VI:**

**Case Study:** The Unified Library application, ATM Application.

**TEXT BOOKS:**

1. Grady Booch, James Rumbaugh, Ivar Jacobson: The Unified Modeling Language User Guide, Pearson Education.
2. Hans-Erik Eriksson, Magnus Penker, Brian Lyons, David Fado: UML 2 Toolkit, WILEY-Dreamtech India Pvt. Ltd.

**REFERENCE BOOKS:**

1. Meilir Page-Jones: Fundamentals of Object Oriented Design in UML, Pearson Education.
2. Pascal Roques: Modeling Software Systems Using UML2, WILEY-Dreamtech India Pvt. Ltd.
3. Atul Kahate: Object Oriented Analysis & Design, The McGraw-Hill Companies.
4. Mark Priestley: Practical Object-Oriented Design with UML,TATA McGrawHill
5. Appling UML and Patterns: An introduction to Object – Oriented Analysis and Design and Unified Process, Craig Larman, Pearson Education.

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0027154) SOFTWARE TESTING METHODOLOGIES**

**OBJECTIVES:**

Upon successful completion of this course students will be able to:

1. Understand the basic concepts of software testing.
2. Understand the various techniques and strategies of software testing and inspection and pointing out the importance of testing in achieving high-quality software.
3. Perform effective and efficient structural testing of software.
4. Integrate and test the various units and components of a software system.
5. Perform effective and efficient functional testing of software.
6. Select the appropriate tests to regression test your software after changes have been made.
7. Plan, track and control the software testing effort.
8. Understand the need of automated testing tools and various kinds of automated testing tools.

**OUTCOMES:**

1. Have an ability to apply software testing knowledge and engineering methods.
2. Have an ability to design and conduct a software test process for a software testing project.
3. Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
4. Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.

**UNIT I:**

**Introduction**: Purpose of testing, Dichotomies, model for testing, consequences of bugs, taxonomy of bugs

**UNIT II:**

**Flow graphs and Path testing:** Basics concepts of path testing, predicates, path predicates and achievable paths, path sensitizing, path instrumentation, application of path testing.

**UNIT III:**

**Transaction Flow Testing**: Transaction flows, transaction flow testing techniques. Dataflow testing:-Basics of dataflow testing, strategies in dataflow testing, application of dataflow testing.

**UNIT IV:**

**Paths, Path products and Regular expressions:** Path products & path expression, reduction procedure, applications, regular expressions & flow anomaly detection.

**UNIT V:**

**Logic Based Testing:** path expressions, kv charts, specifications.

**Graph Matrices and Application**: Motivational overview, matrix of graph, relations, power of a matrix, node reduction algorithm, building tools.

**UNIT VI:**

**State, State Graphs and Transition testing**: State graphs, good & bad state graphs, state testing, Testability tips.

**TEXT BOOKS:**

1. Software Testing techniques - Baris Beizer, Dreamtech, second edition.
2. Software Testing Tools – Dr.K.V.K.K.Prasad, Dreamtech.

**REFERENCES:**

1. The craft of software testing - Brian Marick, Pearson Education.
2. Software Testing Techniques – SPD(Oreille)
3. Software Testing in the Real World – Edward Kit, Pearson.
4. Effective methods of Software Testing, Perry, John Wiley.
5. Art of Software Testing – Meyers, John Wiley.

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0028154) HUMAN COMPUTER INTERACTION**
(ELECTIVE-I)

**OBJECTIVES:**

1. To expose students to the central concepts of Human-Computer Interaction.
2. Establish target users, functional requirements, and interface requirements for a given computer application
3. Describe and explain user interface design principles, and apply them to designing an interface.
4. Develop quick-and-dirty interface designs using rapid prototyping methods.
5. Evaluate user interface designs through usability inspection and user models
6. Develop user studies and analyze study data to gain information about users, tasks, and interface designs.
7. Choose appropriate HCI methods for specified goals and justify the choice

**OUTCOMES:**

The Student will be able to:

1. Apply HCI principles and a user-centered approach to interaction design.
2. Analyze user needs and requirements.
3. Design and develop prototypes based on user assessments (needs and requirements), while applying HCI principles and models.
4. Apply evaluation and usability testing methods to interactive products to validate design decisions.
5. Develop pre-design and post-design usability testing techniques on the developed website.
6. Assess user needs and requirements.
7. Categorize, design and develop information in proper architectural structures.
8. Create interface design prototypes based on a range of design principles and user data, and user assessments.
9. Apply prototype principles and a user-centered approach to interaction design.

**UNIT I**

**Introduction:** Importance of user Interface – definition, importance of good design. Benefits of good design. A brief history of Screen design.

**UNIT II**

**The graphical user interface** – popularity of graphics, the concept of direct manipulation, graphical system, Characteristics, Web user – Interface popularity, characteristics- Principles of user interface.

**UNIT III**

**Design process –** Human interaction with computers, importance of human characteristics human consideration, Human interaction speeds, understanding business junctions.
Screen Designing: - Design goals – Screen planning and purpose, organizing screen elements, ordering of screen data and content.

**UNIT IV**

**Windows –** New and Navigation schemes selection of window, selection of devices based and screen based controls.

**UNIT V**

**Components –** text and messages, Icons and increases – Multimedia, colors, uses problems, choosing colors.

**UNIT VI**

**Software tools –** Specification methods, interface – Building Tools.
Interaction Devices – Keyboard and function keys – pointing devices – speech recognition digitization and generation – image and video displays – drivers.

**TEXT BOOKS:**

1. The essential guide to user interface design, Wilbert O Galitz, Wiley DreamTech.
2. Designing the user interface. 3rd Edition Ben Shneidermann , Pearson Education Asia.

**REFERENCES:**

1. Human – Computer Interaction. Alan Dix, Janet Fincay, Gre Goryd, Abowd, Russell Bealg, Pearson Education
2. Interaction Design Prece, Rogers, Sharps. Wiley Dreamtech.
3. User Interface Design, Soren Lauesen , Pearson Education.
4 .Human –Computer Interaction,D.R.Olsen,Cengage Learning.
5. Human –Computer Interaction,Smith - Atakan,Cengage Learning.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

## (F0029154) COMPUTER GRAPHICS

(Elective-I)

**OBJECTIVES:**

This course is designed to provide a comprehensive introduction to computer graphics leading to the ability to understand contemporary terminology, progress, issues, and trends.

- The interdisciplinary nature of computer graphics is emphasized in the wide variety of examples and applications.
- Course material is structured to meet the needs of both designers and users of interactive computer graphics systems
- Provide an understanding of how a computer draws the fundamental graphics primitives - lines and filled polygons in both 2-D and 3-D.
- Understand the fundamental mathematics involved in generating a 3-D scene. Includes coordinate systems, transformations, and vector operations.
- Understand the 3-D graphics pipeline, i.,e. the steps taken to transform and draw an object. This includes 3-D transformations, lighting and shading, rasterization and texture mapping. Programmable shaders will be introduced.
- A thorough introduction to computer graphics techniques, focusing on 3D modelling, image synthesis, and rendering. Topics cover: geometric transformations, geometric algorithms, software systems (OpenGL), 3D object models (surface, volume and implicit), visible surface algorithms, image synthesis, shading and mapping, ray tracing, radiosity, global illumination, photon mapping, and anti-aliasing.
- Apply knowledge gained in a series of exercises using OpenGL that demonstrate the fundamental principles of computer graphics. A functional scene graph will be developed to render 3-D scenes with lighting, texture mapping, and a variety of 3-D objects

**OUTCOMES**:

- Be familiar with drawing primitive objects (lines, circles, polygons) on a display.
- Be exposed to graphical input and output devices
- Master two dimensional modelling and 2-D transformations.
- Be familiar with master-instance structure.
- Master three dimensional modelling and 3-D transformations.
- Be familiar with projection of 3-D objects on a 2-D plane.
- Master clipping, fill, and rendering techniques.
- Be exposed to color and shading models.
- Be familiar with the visible surface detection methods.
- Animation techniques and languages

**UNIT I**

**A Survey of Computer Graphics:** Overview of graphics systems: Video-Display Devices, Raster-scan Systems, Random-scan Systems, Graphics Monitors and Work stations, Input devices, Hardcopy Devices and Graphics Software.

**UNIT II**

**Output Primitives:** Points and lines, Line drawing algorithms- DDA, Bresenhams line algorithm, Circle generation algorithm and Ellipse Generating algorithms.

**UNIT III**

**2-D Geometrical transforms:** Basic Transformations, Matrix representations and Homogeneous coordinates, Composite transforms, Other Transformations, Transformations between coordinate systems.

**UNIT IV**

**2-D Viewing:** Definition of view port, clipping and window. The Viewing Pipeline, Viewing coordinate reference frame, Window to View-port coordinate transformation, 2D Viewing functions, Clipping Operation, Point Clipping, Line Clipping: Cohen-Sutherland and Liang - Barsky line clipping algorithms, Polygon Clipping: Sutherland – Hodgeman polygon clipping algorithm.

**UNIT V**

**3-D Geometric and Modeling Transformations:** Translation, Rotation, Scaling, other Transformations, Composite Transformations.

**UNIT VI**

**3-D viewing:** Viewing Pipeline, Viewing coordinates, Projections.
Computer Animation: Design of Animation Sequence, General Computer Animation functions, Raster Animation, Computer Animation Languages, Key-Frame systems, Motion Specifications.

**TEXT BOOKS:**

1. "Computer Graphics C version", Donald Hearn and M.Pauline Baker, Pearson Education.

**REFERENCES:**

1. "Computer Graphics Principles & practice", second edition in C, Foley, VanDam, Feiner and Hughes, Pearson Education.
2. "Computer Graphics", second Edition, Donald Hearn and M.Pauline Baker, PHI/Pearson Education.
3. "Computer Graphics Second edition", Zhigand xiang, Roy Plastock, Schaum's outlines, Tata Mc- Graw hill edition.
4. Procedural elements for Computer Graphics, David F Rogers, Tata Mc Graw hill, 2nd edition.
5. "Principles of Interactive Computer Graphics", Neuman and Sproul, TMH.
6. Principles of Computer Graphics, Shalini Govil, Pai, 2005, Springer.
7. Computer Graphics, Steven Harrington, TMH

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0030154) INFORMATION SECURITY**

(Elective-I)

**Objectives:**

- This course covers the major aspects of computer and network security. It starts with a general introduction to information security, then proceeds to cover types of threats and attacks, hacking techniques, network vulnerabilities, security policies and standards, firewalls, cryptography, Authentication & digital signatures, the SSL protocol, Wireless security, intrusion detection and prevention

**Outcomes:**

The student will be able to:

- Define information security and outline its major components.
- Identify the major types of threats to information security and the associated attacks.
- Develop strategies to protect organization information assets from common attacks.
- Understand how security policies, standards and practices are developed.
- Understand the role of management in enforcing security policies, standards and practices.
- Identify the major techniques, approaches and tools used to discover network and system vulnerabilities.
- Understand firewalls and packet filtering.

**UNIT - I**

Security Attacks (Interruption, Interception, Modification and Fabrication), Security Services (Confidentiality, Authentication, Integrity, Non-repudiation, access Control and Availability) and Mechanisms, A model for Internetwork security, Internet Standards and RFCs.

**UNIT - II**

Conventional Encryption Principles, Conventional encryption algorithms, cipher block modes of operation, location of encryption devices, key distribution Approaches of Message Authentication, Secure Hash Functions and HMAC.

**UNIT - III**

Public key cryptography principles, public key cryptography algorithms, digital signatures, digital Certificates, Certificate Authority and key management Kerberos, X.509 Directory Authentication Service.

**UNIT - IV**

Email privacy: Pretty Good Privacy (PGP) and S/MIME.

**UNIT - V**

IP Security Overview, IP Security Architecture, Authentication Header, Encapsulating Security Payload, Combining Security Associations and Key Management.

**UNIT - VI**

Basic concepts of SNMP, SNMPv1 Community facility and SNMPv3.
Intruders, Viruses and related threats.

**TEXT BOOKS:**

1. Network Security Essentials (Applications and Standards) by William Stallings Pearson Education.
2. Hack Proofing your network by Ryan Russell, Dan Kaminsky, Rain Forest Puppy, Joe Grand, David Ahmad, Hal Flynn IdoDubrawsky, Steve W.Manzuik and Ryan Permeh, wileyDreamtech.

**REFERENCES:**

1. Fundamentals of Network Security by Eric Maiwald (Dreamtech press)
2. Network Security - Private Communication in a Public World by Charlie Kaufman, Radia Perlman and Mike Speciner, Pearson/PHI.
3. Cryptography and network Security, Third edition, Stallings, PHI/Pearson
4. Principles of Information Security, Whitman, Thomson.
5. Network Security: The complete reference, Robert Bragg, Mark Rhodes, TMH
6. Introduction to Cryptography, Buchmann, Springer.

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0031154) DISTRIBUTED SYSTEMS**
(ELECTIVE II)

**OBJECTIVES:**

- Present the principles underlying the functioning of distributed systems;.
- Create an awareness of the major technical challenges in distributed systems design and implementation.
- Expose students to modern and classic technology used in distributed systems and their software.
- Provide experience in the implementation of typical algorithms used in distributed systems.

**OUTCOMES:**

After completing this course the student will be able to.

- Explain what a distributed system is, why you would design a system as a distributed system, and what the desired properties of such systems are.
- List the principles underlying the functioning of distributed systems, describe the problems and challenges associated with these principles, and evaluate the effectiveness and shortcomings of their solutions.
- Recognise how the principles are applied in contemporary distributed systems, explain how they affect the software design, and be able to identify features and design decisions that may cause problems.
- Design a distributed system that fulfils requirements with regards to key distributed systems properties (such as scalability, transparency, etc.), be able to recognise when this is not possible, and explain why.

**UNIT-I**

Introduction of Distributed System: Goals, Hardware Concepts, Software Concepts, Client-Server Model.

**UNIT-II**

Communication: Remote Procedure Call, Remote Object Invocation, Message Oriented Communication, Stream-Oriented Communication.

**UNIT-III**

Processes: Threads, Clients, Servers, Code Migration, Software Agents.

**UNIT-IV**

Synchronization: Clock Synchronization, Logical Clocks, Global State, Election Algorithms, Mutual Exclusion, Distributed Transactions.

**UNIT-V**

Consistency and Replication: Introduction, Data-Centric Consistency Models, Client Centric Consistency Models.

**UNIT-VI**

Fault Tolerance: Introduction to Fault Tolerance, Process Resilience, Reliable Client-Server Communication, Reliable Group Communication, Distributed Commit, Recovery.

**Text Books:**

1. Andrew S. Tanenbaum, Maarten Van Steen. Distributed Systems – Principles and Paradigms 2/e, PHI, 2004.

**Reference Books:**

1. Pradeep K. Sinha, “Distributed Operating Systems Concepts and Design”, PHI 2002.

2. Randy Chow Theodore Johnson, "Distributed Operating Systems and Algorithm Analysis", PEA, 2009.
3. GeorgeCouloris, Jean Dollimore, Tim Kind berg, "Distributed Systems Concepts and Design", 3/e, PEA, 2002.

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0032154) ARTIFICIAL INTELLIGENCE**
(ELECTIVE II)

**OBJECTIVES:**

- To introduce the fundamental concepts of artificial intelligence.
- To equip students with the knowledge and skills in logic programming using Prolog.
- To explore the different paradigms in knowledge representation and reasoning.
- To understand the contemporary techniques in machine learning.
- To evaluate the effectiveness of hybridization of different artificial intelligence techniques.

**OUTCOMES:**

- Understand the history, development and various applications of artificial intelligence.
- Familiarize with propositional and predicate logic and their roles in logic programming.
- Understand the programming language Prolog and write programs in declarative programming style.
- Learn the knowledge representation and reasoning techniques in rule-based systems, case-based systems, and model-based system.

**UNIT - I**

**Introduction:** AI problems, foundation of AI and history of AI.
Intelligent agents: Agents and Environments, The concept of rationality, the nature of environments, structure of agents.
**Solving Problems by Searching**: Problem solving agents, Example problems.
**Searching for solutions, Uniformed search strategies**:– Breadth first search, depth first search, Depth limited search, Iterative deepening depth first search, Bi-direction search – comparing uniform search strategy.

**UNIT - II**

**Informed Search and Exploration:** Informed (Heuristic search) Greedy best first search, A* search, Memory bounded heuristic search, Heuristic functions: The effect of heuristic accuracy on performance, Inventing admissible heuristic functions, Learning heuristic from experience.

**Unit-III**

**Constrain satisfaction problems:** Constrained satisfaction problems, Backtracking search for CSPs.
**Knowledge and Reasoning**: Knowledge – Based Agents, the Wumpus world logic.
propositional logic: A very simple logic, syntax, semantics, a simple Knowledge base, inference, equivalence, validity, and sarisfiability, Resolution patterns in propositional logic, Resolution, Forward & Backward, Chaining.

**UNIT - IV**

**First order logic:** Representation revisited, Syntax and semantics of first-order logic: models for first-order logic, symbols and interpretation, terms, atomic sentences, complex sentences, quantifiers, equality.

**Using first order logic:** assertions and queries in first-order logic, the kinship domain, numbers, sets, and lists, the wumpus world.

**UNIT - V**

**Uncertainty–** Acting under uncertainty: handling uncertain knowledge, uncertainty and rational decisions, Design for a decision-theoretic agent, Basic probability notation: propositions, atomic events, prior probability, conditional probability, The Axioms of probability: using the axioms of probability, why the axioms of probability are reasonable,

Interface using full joint distributions, independence, Bayes' rule and its use: Applying Bayes' rule: the simple case, using Bayes' rule: combining evidence.

**UNIT - VI**

**Learning –** Forms of learning, Induction learning.

**Learning Decision Tree:** Decision tree as performance elements, expressiveness of decision tree, Including decision trees from examples, Choosing attribute test, Assessing the performance of the learning algorithm, noise and overfitting, broadening the applicability of decision trees, Ensemble learning.

**TEXT BOOKS:**

1. Artificial Intelligence – A Modern Approach. Second Edition, Stuart Russel, Peter Norvig, PHI/Pearson Education.
2. Artificial Intelligence, 3rd Edition, Patrick Henry Winston., Pearson Edition.

**REFERENCES:**

1. Artificial Intelligence, 2nd Edition, E.Rich and K.Knight (TMH).
2. Artificial Intelligence and Expert Systems – Patterson PHI.
3. Expert Systems: Principles and Programming- Fourth Edn, Giarrantana/ Riley, Thomson.
4. PROLOG Programming for Artificial Intelligence. Ivan Bratka- Third Edition – Pearson Education

| II YEAR, II-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0033154) ADVANCED COMPUTER ARCHITECTURE**

(ELECTIVE - II)

**OBJECTIVES:**

- A broad understanding of computer architecture.
- To the extent possible, an understanding of the current state-of-the-art in uni-processor computer architecture.
- Study how to use technology to build the best computer/processor.
- To know different levels of parallelism.
- Issues in interconnection of networks and cluster design.

**OUTCOMES:**

Upon completion of this course, a student should have developed:

- Broad understanding of the design of computer systems, including modern architectures and alternatives.
- Understanding of the interaction amongst architecture, applications and technology.
- Understanding of a framework for evaluating design decisions in terms of application requirements and performance measurements.
- A historical perspective on computer system design.

**UNIT -I**

**Fundamentals of computer design-**Technology Trends, Cost, Measuring and reporting performance, Quantitative principles of computer design.

**UNIT -II**

**Instruction set principle and examples-**Classifying instruction set, Memory addressing, Addressing modes for signal processing, Operations in the instruction set, Instructions for control flow, encoding an instruction set, the role of the compiler.

**UNIT –III**

**Instruction level parallelism-**Over coming data hazards, Reducing branch costs, High performance instruction delivery, Hardware based speculation, Compiler techniques, Static branch prediction, H.W. vs S.W solutions, Limitations of ILP

**UNIT –IV**

**Memory hierarchy design-**Cache performance, Reducing cache miss penalty and miss rate, Virtual memory, Protection and examples of virtual memory.

**UNIT –V**

**Multi processors and thread-level parallelism-**symmetric shared memory architectures , Distributed -shared memory architectures, Synchronization, Multi-threading .

**UNIT –VI**

**Storage systems** Types of Storage Devices, RAID, Errors and failures in real time systems,

**Interconnection Networks-:** Interconnection network media, Practical issues for commercial interconnection of networks, Clusters, Designing a cluster.

**TEXT BOOK:**

1. Computer Architecture A quantitative approach 3rd edition John L. Hennessy & David A. Patterson Morgan Kufmann (An Imprint of Elsevier)

**REFERENCES:**

2. "Computer Architecture and parallel Processing" Kai Hwang and A.Briggs International Edition McGraw-Hill.
3. Advanced Computer Architectures, Dezso Sima, Terence Fountain, Peter Kacsuk, Pearson.
4. Parallel Computer Architecture, A Hardware / Software Approach, David E. Culler, Jaswinder Pal singh with Anoop Gupta, Elsevier

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

II Year, II-Sem (MCA) P C
3 2

**(F0034154) OBJECT ORIENTED PROGRAMMING THROUGH JAVA LAB**

**OBJECTIVES:**

- To make the student operating systems.
- Learn a object oriented way of solving problems.
- To teach the student to write programs in Java to solve the problems

**OUTCOMES:**

**After Completion of the Lab Course student should be able:**

- To make the student learn a object oriented way of solving problems.
- To teach the student to write programs in Java to solve the problems

**Recommended Systems/Software Requirements:**

- Intel based desktop PC with minimum of 166 MHZ or faster processor with at least 64 MB RAM and 100 MB free disk space.
- JDK Kit. Recommended.

1. Write a Java program that prints all real solutions to the quadratic equation ax2 + bx + c = 0. Read in a, b, c and use the quadratic formula. If the discriminant b2 -4ac is negative, display a message stating that there are no real solutions.
2. The Fibonacci sequence is defined by the following rule. The fist two values in the sequence are 1 and 1. Every subsequent value is the run of the two values preceding it. Write a Java program that uses both recursive and non recursive functions to print the nth value in the Fibonacci sequence.
3. Write a Java program that prompts the user for an integer and then prints out all prime numbers up to that. Integer.
4. a) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome.

   b) Write a Java program that checks whether a given string is a palindrome or not. Ex: MADAM is a palindrome (using StringBuffer class).
5. Write a Java program for sorting a given list of names in ascending order.
6. Write a Java program to multiply two given matrices.
7. Write a Java Program that reads a line of integers, and then displays each integers, and the sum of all the integers (use StringTokenizer class)
8. Write a Java program that illustrates how run time polymorphism is achieved.
9. Write a Java program that illustrates
   a) Constructor Overloading using this keyword
   b) Method Overloading.
10. Write a java program implement
    a) Single Inheritance b) Multilevel
11. Write a java program implement Multiple Inheritance using Interface.
12. Write a java program to create a Package and perform the Arithmetic operations and user can select the operation.
13. Write a java program convert the currency into different currency (Rupees, Pounds and Dollars) using Interface.
14. Write a Java program to make frequency of letters/words in a given test.
15. Write a Java program that displays the number of characters, lines and words in a given text.

16. Write a Java program that:
    a) Implements stack ADT.
17. Write a Java program for creating multiple threads.
18. Write a Java program that correctly implements producer consumer problem using the concept of inter thread communication.
19. Write an applet that displays a simple message.
20. Write a Java program that works as a simple calculator. Use a grid layout to arrange buttons for the digits and for the + - X % operations. Add a text field to display the result.
21. Write a Java program for handling mouse events.
22. Write a Java program for handling keyboard events.
23. Write a Java program to create a button when the user clicks the buttons it display the color (AWT).
24. Write a Java program that allows the user to draw lines, rectangles and Ovals.

**TEXT BOOKS:**

1. The Complete Reference Java J2SE 5th Edition, Herbert Schildt, TMH Publishing Company Ltd, New Delhi.
2. Programming with java primer 3rd Edition, E.Balaguruswamy, McGraHill.

| II YEAR, II-SEM (MCA) | P | C |
|---|---|---|
| | 3 | 2 |

**(F0035154) CASE TOOLS LAB**

1) The student should take up the case study of Unified Library application which is mentioned in the theory, and Model it in different views i.e Use case view, logical view, component view, Deployment view, Database design, forward and Reverse Engineering, and Generation of documentation of the project.

2) Student has to take up another case study of his/her own interest and do the same what ever mentioned in first problem. Some of the ideas regarding case studies are given in reference books which were mentioned in theory syllabus can be referred for some idea.

**Note :** The analysis, design, coding, documentation, database design of mini project which will be carried out in 4th year should be done in object-oriented approach using UML and by using appropriate software which supports UML, otherwise the mini project will not be evaluated.

| II YEAR, II-SEM (MCA) | P | C |
|---|---|---|
| | 3 | 2 |

**(F0036154) SOFTWARE TESTING METHODOLOGIES LAB**

**OBJECTIVES:**

Upon successful completion of this course students will be able to:

- Understand the basic concepts of software testing.
- Understand the various techniques and strategies of software testing and inspection and pointing out the importance of testing in achieving high-quality software.
- Perform effective and efficient structural testing of software.
- Integrate and test the various units and components of a software system.
- Perform effective and efficient functional testing of software.
- Select the appropriate tests to regression test your software after changes have been made.
- Plan, track and control the software testing effort.
- Understand the need of automated testing tools and various kinds of automated testing tools.

**OUTCOMES:**

- Have an ability to apply software testing knowledge and engineering methods.
- Have an ability to design and conduct a software test process for a software testing project.
- Have an ability to identify the needs of software test automation, and define and develop a test tool to support test automation.
- Have an ability understand and identify various software testing problems, and solve these problems by designing and selecting software test models, criteria, strategies, and methods.

1. Write programs in 'C' Language to demonstrate the working of the following constructs: i) do…while ii) while….do iii) if…else iv) switch v) for
2. "A program written in 'C' language for Matrix Multiplication fails" Introspect the causes for its failure and write down the possible reasons for its failure.
3. Take any system (e.g. ATM system) and study its system specifications and report the various bugs.
4. Write the test cases for any known application (e.g. Banking application)
5. Create a test plan document for any application (e.g. Library Management System)
6. Study of any two of the following testing tools:
   a) Study of any testing tool (e.g. Win runner)
   b) Study of any web testing tool (e.g. Selenium)
   c) Study of any bug tracking tool (e.g. Bugzilla, bugbit)
   d) Study of any test management tool (e.g. Test Director)
   e) Study of any open source-testing tool (e.g. Test Link)
7. Take a mini project (e.g. University admission, Placement Portal) and execute it. During the Life cycle of the mini project create the various testing documents* and final test report document.

*Note: To create the various testing related documents refer to the text "Effective Software Testing
Methodologies by William E. Perry"

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

III YEAR, I-SEM (MCA) T 4 C 4

## (F0037155) WEB TECHNOLOGIES

**Objectives:**

This course demonstrates an in-depth understanding of the tools and Web technologies necessary for business application design and development. The course covers client side scripting like HTML, JavaScript and server side scripting like servlets, JSPs. And also XML and web servers and database interfacing.

**Outcomes:**

- The main learning outcomes are:
- Development of a business application.
- Implementation of given client side and server side technologies.
- Design and develop static and dynamic web pages.
- Validate web page data with database data.

**UNIT I**

**HTML Common tags**- List, Tables, images, forms, Frames; Cascading Style sheets.
Introduction to Java Scripts, Objects in Java Script, Dynamic HTML with Java Script.

**UNIT I**

**XML**: Document type definition, XML Schemas, Document Object model, Presenting XML, Using XML Processors: DOM and SAX

**UNIT III**

**Java Beans**: Introduction to Java Beans, Advantages of Java Beans, BDK Introspection, Using Bound properties, Bean Info Interface, Constrained properties Persistence, Customizes, JavaBeans API.

**UNIT IV**

**Web Servers and Servlets:** Tomcat web server, Introduction to Servelets: Lifecycle of a Serverlet, JSDK, The Servelet API, The javax.servelet Package, Reading Servelet parameters, Reading Initialization parameters. The javax.servelet HTTP package, Handling Http Request & Responses, Using Cookies-Session Tracking, Security Issues.

**UNIT V**

**Introduction to JSP**: The Problem with Servelet. The Anatomy of a JSP Page, JSP Processing. JSP Application Design with MVC.

**Database Access:** Database Programming using JDBC, Studying Javax.sql.* package,Accessing a Database from a JSP Page

**UNIT VI**

**JSP Application Development:** Generating Dynamic Content, Using Scripting Elements Implicit JSP Objects, Conditional Processing – Displaying Values Using an Expression to Set an Attribute, Declaring Variables and Methods Error Handling and Debugging Sharing Data Between JSP pages, Requests, and Users Passing Control and Date between Pages – Sharing Session and Application Data – Memory Usage Considerations.

**TEXT BOOKS:**

1. Web Programming, building internet applications, Chris Bates 2nd edition,WILEY Dreamtech (UNIT s 1,2 ,3)
2. The complete Reference Java 2 Fifth Edition by Patrick Naughton and Herbert Schildt. TMH (Chapters: 25) (UNIT 4)
3. Java Server Pages –Hans Bergsten, SPD O'Reilly (UNITs 5,6,7,8)

**REFERENCE BOOKS:**

1. Programming world wide web-Sebesta,Pearson

2. Core SERVLETS ANDJAVASERVER PAGES VOLUME 1: CORE TECHNOLOGIES By Marty Hall and Larry Brown Pearson
3. Internet and World Wide Web – How to program by Dietel and Nieto PHI/Pearson Education Asia.
4. An Introduction to web Design and Programming –Wang-Thomson
5. Web Applications Technologies Concepts-Knuckles,John Wiley
6. Programming world wide web-Sebesta,Pearson

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| III YEAR, I-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0038155) DATA WAREHOUSING AND DATA MINING**

**OBJECTIVES:**

The main objective of this course is to provide students

- With the basic data warehousing and data mining concepts
- To learn mining rules in large databases
- To get idea on clustering analysis
- Applications that can enable them to set up and manage an industrial data warehousing and data mining system.
- To learn about OLTP and OLAP systems

**OUTCOMES:**

Upon completion of the course students should:

- Ability to do Conceptual, Logical, and Physical design of Data Warehouses
- Familiarity with Requirements Engineering for Data Warehouses
- OLAP applications and OLAP deployment
- Have a good knowledge of the fundamental concepts that provide the foundation of data mining.
- Learn broad classes of data mining technologies
- Understand how theses concepts are engineered to use some of the basic data mining tools.

**UNIT – I**

**Introduction:** Fundamentals of data mining, Data Mining Functionalities, Classification of Data Mining systems, Data Mining task primitives, Integration of data mining system with Data base or Data Warehouse system, Major issues in Data Mining.

**UNIT – II**

**Data Warehouse and OLAP Technology:** Overview of Data Warehouse, Multidimensional Data Model, Data Warehouse Architecture, Data Warehouse Implementation, From Data Warehousing to Data Mining.

**UNIT – III**

**Data Preprocessing:** Need of preprocessing the data, Descriptive Data Summarization, Data Cleaning, Data Integration and Transformation, Data Reduction, Data Discretization and Concept Hierarchy Generation.

**UNIT – IV**

**Mining Frequent patterns, Associations, and Correlations:** Basic Concepts, Efficient and Scalable Frequent Itemset Mining methods, Mining Various Kinds of Association Rules, From Association Mining to Correlation Analysis, Constraint-Based Association Mining.

**UNIT – V**

**Classification :** Overview of Classification and Prediction, Issues Regarding Classification and Prediction, Bayesian Classification, Classification by Decision Tree Induction, Rule-Based Classification, Classification by Back propagation,.

**UNIT – VI**

**Cluster Analysis:** Overview of Cluster Analysis, Types of data in Cluster Analysis, A Categorization of Major Clustering Methods, Partitioning Methods, Hierarchical Methods, Density-Based Methods. Grid-Based Methods, Model-Based Clustering Methods.

**TEXT BOOKS:**

1. Data Mining – Concepts and Techniques - JIAWEI HAN & MICHELINE KAMBER Harcourt India, second Edition.

**REFERENCES:**

1. Data Mining Introductory and advanced topics –MARGARET H DUNHAM, PEARSON EDUCATION
2. Data Mining Techniques – ARUN K PUJARI, University Press.
3. Data Warehousing in the Real World – SAM ANAHORY & DENNIS MURRAY. Pearson Edn Asia.
4 Data Warehousing Fundamentals – PAULRAJ PONNAIAH WILEY STUDENT EDITION
5. The Data Warehouse Life cycle Tool kit – RALPH KIMBALL WILEY STUDENT EDITION.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| III YEAR, I-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0039155) Mobile Application Development**

**OBJECTIVES:**

- Describe those aspects of mobile programming that make it unique from programming for other platforms,
- Critique mobile applications on their design pros and cons,
- Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,
- Program mobile applications for the Android operating system that use basic and advanced phone features, and
- Deploy applications to the Android marketplace for distribution.

**OUTCOMES:**

- Be exposed to technology and business trends impacting mobile applications
- Be competent with the characterization and architecture of mobile applications.
- Be competent with understanding enterprise scale requirements of mobile applications
- Be competent with designing and developing mobile applications using one application development framework

**UNIT I**

**J2ME Overview:** Java 2 Micro Edition and the World of Java, Inside J2ME, J2ME and Wireless Devices.

**Small Computing Technology:** Wireless Technology, Radio Data Networks, Microwave Technology, Mobile Radio Networks, Messaging, Personal Digital Assistants, Mobile Power, Set-Top Boxes, Smart Cards.

**UNIT II**

**J2ME Architecture and Development Environment:** J2ME Architecture, Small Computing Device Requirements, Run-Time Environment, MIDlet Programming, Java Language for J2ME, J2ME Software Development Kits, Hello World J2ME Style, Multiple MIDlets in a MIDlet Suite, J2ME Wireless Toolkit.

**UNIT III**

**J2ME Best Practices and Patterns***:* The Reality of Working in a J2ME World, Best Practices**.**

**Commands, Items and Event Processing:** J2ME User Interfaces, Display Class, The Palm OS Emulator, Command Class, Item Class, Exception Handling.

**UNIT IV**

**High-Level Display Screens:** Screen Class, Alert Class, Form Class, Item Class, List Class, Text Box Class, Ticker Class.

**UNIT V**

**Low-Level Display Canvas:** The Canvas, User Interactions, Graphics, Clipping Regions, Animation.

**UNIT VI**

**Record Management System-** Record Storage, Writing and Reading Records, Record Enumeration, Sorting Records, Searching Records, Record Listener.

**JDBC Objects:** The Concept of JDBC, JDBC Driver Types, JDBC Packages, Overview of the JDBC Process, Database Connection, statement Objects, Result set, Transaction Processing, Metadata, Data Types, Exceptions.

**TEXT BOOKS:**

1. J2ME: The Complete Reference, James Keogh, Tata McGrawHill.

**REFERENCES:**

1. Enterprise J2ME: Developing Mobile Java Applications – Michael Juntao Yuan, Pearson Education, 2004
2. Beginning Java ME Platform, Ray Rischpater, Apress, 2009
3. Beginning J2ME: From Novice to Professional, Third Edition, Sing Li, Jonathan B. Knudsen, Apress, 2005
4. Kicking Butt with MIDP and MSA:Creating Great Mobile Applications,1st edition,J.Knudsen,Pearson.

| III YEAR, I-SEM (MCA) | T | C |
|---|---|---|
| | 4 | 4 |

**(F0040155) E – COMMERCE**
**(Elective-III)**

**OBJECTIVES:**

- Have a general understanding of the Internet and related technologies
- Understanding of policy issues related to privacy, content selection, intellectual property rights, and establishing identity that are germane to electronic commerce.
- Capability to analyze the impact that electronic commerce is having and will likely have on key sectors of the economy and assess the strategic implications this analysis holds for an organization.
- Be able to analyze the organizational fit between strategy and technology
- Recognize and understand ways of using electronic commerce technologies to improve intra and inter-organizational processes.
- Be able to specify the development of electronic commerce capabilities in a company

**OUTCOMES**

- Identify the organizational requirements of eCommerce systems on data protection;
- Demonstrate knowledge of the factors which have impacts upon the security of eCommerce systems;
- Make realistic assessment on the security of eCommerce systems;
- Design and analyze security measures to protect organizational data against various attacks;
- Describe relevant regulations governing electronic transactions, data privacy protection, and web access.

**UNIT - I**

**Electronic Commerce:-**Frame work, Anatomy of E-Commerce applications, E-Commerce Consumer Applications, E-Commerce Organization Applications, Consumer Oriented Electronic Commerce: - Consumer Oriented Applications, Mercantile Process models.

**UNIT - II**

**Electronic Payment Systems:-** Types of Electronic Payment System, Digital Token-Based, Smart Cards, Credit Cards, Risks and Electronic Payment Systems.

**UNIT-III**

**Inter Organizational Commerce and EDI:**- Electronic Data Interchange, EDI in Application Business, EDI: Legal, Security and Privacy Issues, EDI and Electronic Commerce.

**EDI Implementation, MIME and Value Added Networks:-** Standardization and EDI, EDI Software Implementation, EDI Envelope for Message Transport, Value added networks, Internet Based EDI.

**UNIT - IV**

**Corporate Digital Library –** Dimensions of Internal EC Systems, Making a Business Case for a Document Library, Types of Digital Document, Issues behind Document Infrastructure, Corporate Data Warehouses. **Advertising and Marketing -** Information Based Marketing, Advertising on Internet, On-line Marketing Process, Market Research.

**UNIT - V**

**Consumer Search and Resource Discovery -** Information search and Retrieval, Commerce Catalogues, Information Filtering.

**UNIT - VI**

**Multimedia -** Key Multimedia Concepts, Digital Video and Electronic Commerce, Desktop Video Processing's, Desktop Video Conferencing.

**TEXT BOOK:**

1. Frontiers of electronic commerce – Kalakata, Whinston, Pearson.

**REFERENCES:**

1. E-Commerce fundamentals and applications Hendry Chan, Raymond Lee, Tharam Dillon, Ellizabeth Chang, John Wiley.
2. E-Commerce, S.Jaiswal – Galgotia.
3. E-Commerce, Efrain Turbon, Jae Lee, David King, H.Michael Chang.
4. Electronic Commerce – Gary P.Schneider – Thomson.
5. E-Commerce – Business, Technology, Society, Kenneth C.Taudon, Carol Guyerico Traver.

III YEAR, I-SEM (MCA) T 4 C 4

**(F0041155) DISTRIBUTED DATABASES**
**(Elective-III)**

**OBJECTIVES:**
The first is study of the classical distributed database management
- Issues in Distribution design, distributed query processing, and distributed transaction management.
- To learn about Optimization in DDB
- Architectural Issues in DDB

**OUTCOMES:**
Upon completion of the course students are able to
- Understand distributed database management;
- Good knowledge of complex topics like Distribution design, distributed query processing
- Solution to the some of the Architectural issues.
- Examples of DDB's.

**UNIT I**
**Introduction to Distributed Databases:** Features of Distributed versus Centralized Databases, why distributed databases?, Distributed Database Management Systems, Review of databases, Review of computer networks.

**UNIT II**
**Principles Of Distributed Databases:** Levels Of Distribution Transparency, Reference Architecture for Distributed Databases, Types of Data Fragmentation, Distribution Transparency for Read only Applications, Distribution Transparency for Update Applications, Distributed Database Access Primitives, Integrity Constraints in Distributed Databases.
**Distributed Database Design:** A Framework for Distributed database Design, The Design of database Fragmentation.

**UNIT III**
**Translation of Global Queries to Fragment Queries:** Equivalence Transformations for Queries, Transforming Global Queries into Fragment Queries, Distributed Grouping and Aggregate Function Evaluation, Parametric Queries.

**UNIT IV**
**The Management of Distributed Transactions:** A Framework for Transaction Management, Supporting Atomicity of Distributed Transactions, Concurrency Control for Distributed Transactions, Architectural Aspects of Distributed Transactions.

**UNIT V**
**Concurrency Control:** Foundation of Distributed Concurrency Control, Distributed Deadlocks, Concurrency Control based on Timestamps, Optimistic Methods for Distributed Concurrency Control.

**UNIT VI**
**Reliability:** Basic Concepts, Nonblocking Commitment Protocols, Reliability and concurrency Control, Determining a Consistent View of the Network, Detection and Resolution of Inconsistency, Checkpoints and Cold Restart.
**Distributed Database Administration:** Catalog Management in Distributed Databases, Authorization and Protection.

**TEXT BOOKS:**
1. Distributed Database Principles & Systems, Stefano Ceri, Giuseppe Pelagatti McGraw-Hill.

**REFERENCES:**
1. Principles of Distributed Database Systems, M.TamerOzsu, Patrick Valduriez – Pearson Education.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

III YEAR, I-SEM (MCA) T C
4 4

## (F0042155) IMAGE PROCESSING

(ELECTIVE III)

**OBJECTIVES:**

- To know the fundamentals of Image Processing.
- To know sampling and reconstruction procedures.
- To know various transforms used in image Processing.
- To know about various techniques of image enhancement, reconstruction and image compression.
- To design image processing systems.

**OUTCOMES:**

- Develops ability to identify, formulate &solve problems involving images.
- Develops ability to design &conduct experiments, analyze &interpret image data.
- To design a software, Component or process as per needs &specifications.
- It will demonstrate the skills to use modern engineering tools, software's &equipment to analyze problems.
- Develop confidence for self-education &ability for life-long learning.
- It will show the ability to participate &try to succeed in competitive Exams.

**UNIT - I**

**Introduction:** Examples of fields that use digital image processing, fundamental steps in digital image processing, components of image processing system.. Digital Image Fundamentals: A simple image formation model, image sampling and quantization, basic relationships between pixels.

**UNIT - II**

**Image enhancement in the spatial domain:** Basic gray-level transformation, histogram processing, enhancement using arithmetic and logic operators, basic spatial filtering, smoothing and sharpening spatial filters, combining the spatial enhancement methods.

**UNIT - III**

**Image restoration:** A model of the image degradation/restoration process, noise models, restoration in the presence of noise–only spatial filtering, Weiner filtering, constrained least squares filtering, geometric transforms, estimating the degradation function.

**UNIT - IV**

**Color Image Processing:** Color fundamentals, color models, pseudo color image processing, basics of full–color image processing, color transforms, smoothing and sharpening, color segmentation.

**UNIT - V**

**Image Compression:** Fundamentals, image compression models, error-free compression, lossy predictive coding, image compression standards.

**UNIT - VI**

**Morphological Image Processing:** Preliminaries, dilation, erosion, open and closing, hit or miss transformation, basic morphologic algorithms

**Image Segmentation:** Detection of discontinuous, edge linking and boundary detection, thresholding, region–based segmentation.

**TEXT BOOK:**

1. Digital Image Processing, RafealC.Gonzalez, Richard E.Woods, Second Edition, Pearson Education/PHI.

**REFERENCES:**

1. Image Processing, Analysis, and Machine Vision, Milan Sonka, Vaclav Hlavac and Roger Boyle, Second Edition, Thomson Learning.
2. Introduction to Digital Image Processing with Matlab, Alasdair McAndrew, Thomson Course Technology
3. Computer Vision and Image Processing, Adrian Low, Second Edition, B.S.Publications
4. Digital Image Processing using Matlab, RafealC.Gonzalez, Richard E.Woods, Steven L. Eddins, Pearson Education.
5. Digital Image Processing, William K. Prat, Wily Third Edition
6. Digital Image Processing and Analysis, B. Chanda, D. DattaMajumder, Prentice Hall of India, 2003.

III YEAR, I-SEM (MCA) T 4 C 4

**(F0043155) SOFTWARE PROJECT MANAGEMENT**

(Elective-IV)

**Objective:**

The objective of the course is to familiarise students in practice with the initiation, management and supervision of a software project. During the course, actual software projects are defined and their implementation is managed and supervised. To provide basic project management skills with a strong emphasis on issues and problems associated with delivering successful IT projects. The module is designed to provide an understanding of the particular issues encountered in handling IT projects and to offer students methods, techniques and 'hands-on' experience in dealing with them.

**OUTCOMES:**

The student will be able to:

1. Understand and practice the process of project management and its application in delivering successful IT projects;
2. evaluate a project to develop the scope of work, provide accurate cost estimates and to plan
3. the various activities;
4. understand and use risk management analysis techniques that identify the factors that put a project at risk and to quantify the likely effect of risk on project timescales;
5. identify the resources required for a project and to produce a work plan and resource schedule;
6. monitor the progress of a project and to assess the risk of slippage, revising targets or counteract drift;
7. Distinguish between the different types of project and follow the stages needed to negotiate an appropriate contract.

**UNIT - I**

**Conventional Software Management:** The waterfall model, conventional software Management performance.

**Evolution of Software Economics:** Software Economics, pragmatic software cost estimation.

**UNIT - II**

**Improving Software Economics:** Reducing Software product size, improving software processes, improving team effectiveness, improving automation, Achieving required quality, peer inspections.

**The old way and the new:** The principles of conventional software Engineering, principles of modern software management, transitioning to an iterative process.

**UNIT - III**

**Life cycle phases:** Engineering and production stages, inception, Elaboration, construction, transition phases.

**Artifacts of the process:** The artifact sets, Management artifacts, Engineering artifacts, programmatic artifacts.

**UNIT - IV**

**Model based software architectures:** A Management perspective and technical perspective.

**Work Flows of the process:** Software process workflows, Iteration workflows.

**UNIT - V**

**Checkpoints of the process:** Major mile stones, Minor Milestones, Periodic status assessments.

**Iterative Process Planning:** Work breakdown structures, planning guidelines, cost and schedule estimating, Iteration planning process, Pragmatic planning.

**UNIT - VI**

**Project Organizations and Responsibilities:** Line-of-Business Organizations, Project Organizations, evolution of Organizations.

**Process Automation:** Automation Building blocks, The Project Environment.

**TEXT BOOK:**

1. Software Project Management, Walker Royce: Pearson Education, 2005.

**REFERENCES:**

1. Software Project Management, Bob Hughes and Mike Cotterell: Tata McGraw-Hill Edition.
2. Software Project Management, Joel Henry, Pearson Education.
3. Software Project Management in practice, PankajJalote, Pearson Education.2005

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

III YEAR, I-SEM (MCA) T 4 C 4

## (F0044155) MIDDLEWARE TECHNOLOGIES

(Elective-IV)

**OBJECTIVES:**

- Course will primarily deal with the idea of service-oriented architecture (SOA) and apply it to the domain of educational resources. It has the overall aim of demonstrating how a set of application interfaces can be used to deliver services to learners and learning facilitators.
- Illustrating the concept of SOA, the Open ICOPER Content Space (OICS) together offer an umbrella for combining a portfolio of interoperable repositories, content and tools, where the services, quality attributes and design principles of a SOA are well represented.

**OUTCOMES:**

The student will be able to:

- Describe Service Oriented Architecture (SOA) principles (the basic methodology employed for the OICS creation).
- Understand the usages and services provided by the Open ICOPER Content Space (OICS) as an example of SOA learning content.
- Identify the ICOPER Reference Model (IRM) that regroups the shareable online content.
- Give examples relating to how the Middle Layer API facilitates the implementation of OICS client interfaces.

**UNIT-I:**

**Introduction to client server computing:** Evolution of corporate computing models from centralized to distributed computing, client server models. Benefits of client server computing, pitfalls of client server programming.

**UNIT-II:**

**CORBA with Java:** Review of Java concept like RMI, RMI API, JDBC. Client/Server CORBA-style, The object web: CORBA with Java.

**UNIT III:**

**Introducing C# and the .NET Platform;** Understanding .NET Assemblies; Object – Oriented Programming with C#; Callback Interfaces, Delegates, and Events.

**UNIT IV:**

**Building c# applications:** Type Reflection, Late Binding, and Attribute-Based Programming; Object Serialization and the .NET Remoting Layer; Data Access with ADO.NET; XML Web Services.

**UNIT-V:**

**Core CORBA / Java:** Two types of Client/ Server invocations-static, dynamic. The static CORBA, first CORBA program, ORBlets with Applets, Dynamic CORBA-The portable count, the dynamic count multi count.

**Existential CORBA:** CORBA initialization protocol, CORBa activation services, CORBAIDL mapping CORBA java- to- IDL mapping, The introspective CORBA/Java object.

**UNIT-VI:**

**Java Bean Component Model:** Events, properties, persistency, Intrespection of beans, CORBA Beans

**EJBs and CORBA:** Object transaction monitors CORBA OTM's, EJB and CORBA OTM's, EJB container frame work, Session and Entity Beans, The EJB client/server development Process The EJB container protocol, support for transaction EJB packaging EJB design Guidelines.

**Text Books:**

1. Client/Server programming with Java and CORBA Robert Orfali and Dan Harkey, John Wiley & Sons, SPD 2nd Edition

2. Java programming with CORBA 3rd Edition, G.Brose, A Vogel and K.Duddy, Wiley-dreamtech, India John wiley and sons
3. C# and the .NET Platform Andrew Troelsen, Apress Wiley-dreamtech, India Pvt Ltd

**Reference: Books:**

1. Distributed Computing, Principles and applications, M.L.Liu, Pearson Education
2. Client/Server Survival Guide 3rd edition Robert Orfali Dan Harkey and Jeri Edwards, John Wiley & Sons
3. Client/Server Computing D T Dewire, TMH.
4. IBM Webspere Starter Kit Ron Ben Natan Ori Sasson, TMh, New Delhi
5. Programming C#, Jesse Liberty, SPD-O'Reilly.
6. C# Preciesely Peter Sestoft and Henrik I. Hansen, Prentice Hall of India
7. Intoduction to C# Using .NET Pearson Education

III YEAR, I-SEM (MCA) T C
4 4

**(F0045155) CLOUD COMPUTING**
(Elective-IV)

**OBJECTIVES**:

- The objective focuses on technologies specific to the networked, distributed dimension of software and access to services and data. It will support long-term research on new principles, methods, tools and techniques enabling software developers in the EU to easily create interoperable services based on open standards, with sufficient flexibility and at a reasonable cost.

**OUTCOMES**:

- Intelligent and autonomic management of cloud resources, ensuring agile elastic scalability. Scalable data management strategies, addressing the issues of heterogeneity, consistency, availability, privacy and supporting security.
- Technologies for infrastructure virtualization, cross platforms execution as needed for service composition across multiple, heterogeneous environments, autonomous
- Management of hardware and software resources.
- Interoperability amongst different clouds, portability, protection of data in cloud environments, control of data distribution and latency.
- Seamless support of mobile, context-aware applications.

**UNIT-I**

**Introduction:** What Is the Cloud? The Emergence of Cloud Computing, The Global Nature of the Cloud, Cloud-Based Service Offerings, Grid Computing or Cloud Computing?,Is the Cloud Model Reliable?, Benefits of Using a Cloud Model, What About Legal Issues When Using Cloud Models?, What Are the Key Characteristics of Cloud Computing?, Challenges for the Cloud.

**The Evolution of Cloud Computing:** Hardware Evolution, Internet Software Evolution, Server Virtualization.

**UNIT-II**

**Web Services Delivered from the Cloud:** Communication-as-a-Service (CaaS), Infrastructure-as-a-Service (IaaS), Monitoring-as-a-Service (MaaS), Platform-as-a-Service (PaaS), Software-as-a-Service (SaaS), **Building Cloud Networks:** The Evolution from the MSP Model to Cloud.

Computing and Software-as-a-Service, the Cloud Data Center, Collaboration, Service-Oriented Architectures as a Step Toward Cloud Computing, Basic Approach to a Data Center-Based SOA.

**UNIT III**

**Virtualization Practicum:** Chapter Overview, Downloading Sun xVM VirtualBox, Installing Sun xVM VirtualBox, Adding a Guest Operating System to VirtualBox, Downloading FreeDOS as a Guest OS, Downloading the 7-Zip Archive Tool, Adding a Guest OS to Sun xVM VirtualBox.

**UNIT IV**

**Federation, Presence, Identity, and Privacy in the Cloud:** Federation in the cloud, Presence in the Cloud, Privacy and Its Relation to Cloud-Based Information Systems,

**UNIT V**

**Security in the Cloud**: Cloud security challenges, Software- as-a-service security, Is Security-as-a-Service the New MSSP?

**UNIT VI**

**Common Standards in Cloud Computing:** The open cloud consortium- The distributed management task force - standards for application developers - standards for messaging - standards for security

**TEXTBOOKS:**

1. Cloud Computing implementation, management and security by Joh n W.Ruttinghouse, James F. Ransome. CRC Press, Taylor & Franc is group, 2010.
2. Cloud computing a practical approach by Anthony T.velte, Toby J.velte Robert Elsenpeter. Tata Me Graw Hill edition , 2010

**REFERENCES: 1.** Cloud Application Architectures by George Reese. Oreilly publishers.

2. Cloud computing and SOA convergence in your enterprise, by David S. Linthicum, Addison- Wesley.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

III YEAR, I-SEM (MCA) P 3 C 2

## (F0046155) WEB TECHNOLOGIES LAB

**Objective :**

To create a fully functional website with mvc architecture. To Develop an online Book store using we can sell books (Ex amazon .com).

**Hardware and Software required:**

1. A working computer system with either Windows or Linux
2. A web browser either IE or firefox
3. Tomcat web server and Apache web server
4. XML editor like Altova Xml-spy [www.Altova.com/**XML**Spy – free ] , Stylusstudio , etc.,
5. A database either Mysql or Oracle
6. JVM(Java virtual machine) must be installed on your system
7. BDK(Bean development kit) must be also be installed

**Week-1**:

Design the following static web pages required for an online book store web site.

1) **HOME PAGE:**

The static home page must contain three **frames**.

Top frame : Logo and the college name and links to Home page, Login page, Registration page, Catalogue page and Cart page (the description of these pages will be given below).

Left frame : At least four links for navigation, which will display the catalogue of respective links.

For e.g.: When you click the link **"CSE"** the catalogue for **CSE** Books should be displayed in the Right frame.

Right frame: The *pages to the links in the left frame must be loaded here*. Initially this page contains description of the web site.

<table>
<tr><td>Logo</td><td colspan="4">Web Site Name</td></tr>
<tr><td>Home</td><td>Login</td><td>Registration</td><td>Catalogue</td><td>Cart</td></tr>
<tr><td>CSE<br>ECE<br>EEE<br>CIVIL</td><td colspan="4">Description of the Web Site</td></tr>
</table>

Fig 1.1

**2) LOGIN PAGE:**

This page looks like below:

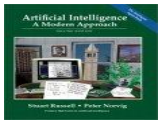
| Logo | Web Site Name | | | |
|---|---|---|---|---|
| Home | Login | Registration | Catalogue | Cart |
| CSE<br>ECE<br>EEE<br>CIVIL | Login :<br>Password: | | Submi | Reset |

**3) CATOLOGUE PAGE:**

The catalogue page should contain the details of all the books available in the web site in a table.

The details should contain the following:

1. Snap shot of Cover Page.
2. Author Name.
3. Publisher.
4. Price.
5. Add to cart button.

| Logo | Web Site Name | | | |
|---|---|---|---|---|
| Home | Login | Registration | Catalogue | Cart |
| CSE<br>ECE<br>EEE<br>CIVIL |  | Book : XML Bible<br>Author : Winston<br>Publication : Wiely | $ 40.5 | Add to cart |
| |  | Book : AI<br>Author : S.Russel<br>Publication :<br>Princeton hall | $ 63 | Add to cart |
| |  | Book : Java 2<br>Author : Watson<br>Publication : BPB<br>publications | $ 35.5 | Add to cart |
| |  | Book : HTML in<br>24 hours<br>Author : Sam<br>Peter<br>Publication : Sam<br>publication | $ 50 | Add to cart |

Note: Week 2 contains the remaining pages and their description.

**Week-2**:

**4) CART PAGE:**

The cart page contains the details about the books which are added to the cart.
The cart page should look like this:

| Logo | Web Site Name | | | |
|---|---|---|---|---|
| Home | Login | Registration | Catalogue | Cart |
| CSE<br>ECE<br>EEE<br>CIVIL | **Book name**<br>Java 2<br>XML bible<br><br>**Amount**<br>$130.5 | **Price**<br>$35.5<br>$40.5<br><br>**Total amount** | **Quantity**<br>2<br>1 | $70<br>$40.5 |

**5) REGISTRATION PAGE:**

Create a "*registration form* "with the following fields

1) Name (Text field)
2) Password (password field)
3) E-mail id (text field)
4) Phone number (text field)
5) Sex (radio button)
6) Date of birth (3 select boxes)
7) Languages known (check boxes – English, Telugu, Hindi, Tamil)
8) Address (text area)

**WEEK 3:**

**VALIDATION:**

Write *JavaScript* to validate the following fields of the above registration page.

1. Name (Name should contains alphabets and the length should not be less than 6 characters).
2. Password (Password should not be less than 6 characters length).
3. E-mail id (should not contain any invalid and must follow the standard pattern name@domain.com)
4. Phone number (Phone number should contain 10 digits only).

**Note** : You can also validate the login page with these parameters.

**Week-4**:

Design a web page using **CSS (C**ascading **S**tyle **S**heets**)** which includes the following:

1) Use different font, styles:

In the style definition you define how each selector should work (font, color etc.).
Then, in the body of your pages, you refer to these selectors to activate the styles.

For example:

```
<HTML>
<HEAD>
<style type="text/css">
B.headline {color:red; font-size:22px; font-family:arial; text-
decoration:underline}
</style>
</HEAD>
<BODY>
<b>This is normal bold</b><br>
Selector {cursor:value}
For example:
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
```

```
</style>
</head>
<body>
<b>
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
<b class="headline">This is headline style bold</b>
</BODY>
</HTML>
```

2) Set a background image for both the page and single elements on the page.
You can define the background image for the page like this:

```
BODY {background-image:url(myimage.gif);}
```

3) Control the repetition of the image with the background-repeat property.
As background-repeat: repeat Tiles the image until the entire page is filled, just like an ordinary
background image in plain HTML.

4) Define styles for links as
A:link
A:visited
A:active
A:hover
Example:

```
<style type="text/css">
A:link {text-decoration: none}
A:visited {text-decoration: none}
A:active {text-decoration: none}
A:hover {text-decoration: underline; color: red;}
</style>
```

5) Work with layers:
For example:

```
LAYER 1 ON TOP:
<div style="position:relative; font-size:50px; z-index:2;">LAYER 1</div>
<div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-
                        index:1">LAYER 2</div>

LAYER 2 ON TOP:
<div style="position:relative; font-size:50px; z-index:3;">LAYER 1</div>
<div style="position:relative; top:-50; left:5; color:red; font-size:80px; z-
                        index:4">LAYER 2</div>
```

6) Add a customized cursor:
Selector {cursor:value}
For example:

```
<html>
<head>
<style type="text/css">
.xlink {cursor:crosshair}
.hlink{cursor:help}
</style>
</head>
<body>
<b>
```

```
<a href="mypage.htm" class="xlink">CROSS LINK</a>
<br>
<a href="mypage.htm" class="hlink">HELP LINK</a>
</b>
</body>
</html>
```

**Week-5**:
Write an XML file which will display the Book information which includes the following:

1) Title of the book
2) Author Name
3) ISBN number
4) Publisher name
5) Edition
6) Price

Write a Document Type Definition (DTD) to validate the above XML file.
Display the XML file as follows.
The contents should be displayed in a table. The header of the table should be in color GREY. And the Author names column should be displayed in one color and should be capitalized and in bold. Use your own colors for remaining columns.
Use XML schemas XSL and CSS for the above purpose.
Note: Give at least for 4 books. It should be valid syntactically.
Hint: You can use some xml editors like XML-spy

**Week-6**:
**VISUAL BEANS:**
Create a simple visual bean with a area filled with a color.
The shape of the area depends on the property shape. If it is set to true then the shape of the area is Square and it is Circle, if it is false.
The color of the area should be changed dynamically for every mouse click. The color should also be changed if we change the color in the "property window ".

**Week-7**:

1) Install TOMCAT web server and APACHE.
   While installation assign port number 4040 to TOMCAT and 8080 to APACHE. Make sure that these ports are available i.e., no other process is using this port.
2) Access the above developed static web pages for books web site, using these servers by putting the web pages developed in week-1 and week-2 in the document root. Access the pages by using the urls : http://localhost:4040/rama/books.html (for tomcat)
   http://localhost:8080/books.html (for Apache)

**Week-8:**
**User Authentication :**

Assume four users user1,user2,user3 and user4 having the passwords pwd1,pwd2,pwd3 and pwd4 respectively. Write a servelet for doing the following.
1. Create a Cookie and add these four user id's and passwords to this Cookie.
2. Read the user id and passwords entered in the Login form (week1) and authenticate with the values (user id and passwords ) available in the cookies.
If he is a valid user(i.e., user-name and password match) you should welcome him by name(user-name) else you should display " You are not an authenticated user ".
Use init-parameters to do this. Store the user-names and passwords in the webinf.xml and access them in the servlet by using the getInitParameters() method.

**Week-9:**

Install a database(Mysql or Oracle).
Create a table which should contain at least the following fields: name, password, email-id, phone number(these should hold the data from the registration form).

Practice 'JDBC' connectivity.

Write a java program/servlet/JSP to connect to that database and extract data from the tables and display them. Experiment with various SQL queries.

Insert the details of the users who register with the web site, whenever a new user clicks the submit button in the registration page (week2).

**Week-10:**

Write a JSP which does the following job:

Insert the details of the 3 or 4 users who register with the web site (week9) by using registration form. Authenticate the user when he submits the login form using the user name and password from the database ( similar to week8 instead of cookies).

**Week-11:**

Create tables in the database which contain the details of items (books in our case like Book name , Price, Quantity, Amount )) of each category. Modify your catalogue page (week 2)in such a way that you should connect to the database and extract data from the tables and display them in the catalogue page using JDBC.

**Week-12**:

**HTTP** is a stateless protocol. Session is required to maintain the state.

The user may add some items to cart from the catalog page. He can check the cart page for the selected items. He may visit the catalogue again and select some more items. Here our interest is the selected items should be added to the old cart rather than a new cart. Multiple users can do the same thing at a time(i.e., from different systems in the LAN using the ip-address instead of localhost). This can be achieved through the use of sessions. Every user will have his own session which will be created after his successful login to the website. When the user logs out his session should get invalidated (by using the method session.invalidate() ).

Modify your catalogue and cart JSP pages to achieve the above mentioned functionality using sessions.

RAJEEV GANDHI MEMORIAL COLLEGE OF ENGG AND TECHNOLOGY
AUTONOMOUS

# MASTER OF COMPUTER APPLICATIONS MCA-R15

| III Year, I-Sem (MCA) | P | C |
|---|---|---|
| | 3 | 2 |

**(F0047155) Mobile Application Development Lab**

**OBJECTIVES:**

- Critique mobile applications on their design pros and cons,
- Utilize rapid prototyping techniques to design and develop sophisticated mobile interfaces,
- Program mobile applications for the Android operating system that use basic and advanced phone features, and
- Deploy applications to the Android marketplace for distribution.

**OUTCOMES:**

- Be competent with the characterization and architecture of mobile applications.
- Be competent with understanding enterprise scale requirements of mobile applications.

## J2ME LAB PROGRAMS

1. Write a program for hello world.
2. Write a program for user interface.
3. Write a program for Text field class.
4. Write a program for Command class.
5. Write a program for login example to validate user.
6. Write a program for Check box.
7. Write a program for Gauge class.
8. Write a program for Ticker class.
9. Write a program for validate user using **JDBC type4 driver**.
10. Write a program for Registration using **Prepared Statement**.