



SACRED HEART COLLEGE (AUTONOMOUS)

Tirupattur – 635 601, Tamil Nadu, S.India

Resi : (04179) 220103

College : (04179) 220553

Fax : (04179) 226423

Ready for
Every Good Work

A Don Bosco Institution of Higher Education, Founded in 1951 * Affiliated to Thiruvalluvar University, Vellore * Autonomous since 1987

Accredited by NAAC (4th Cycle – under RAF) with CGPA of 3.31 / 4 at ‘A+’ Grade

B.Sc., Computer Science

Programme Structure

(from 2021 - 2022 onwards)

Sem	Part	Subcode	Subtitle	Hours	Credits
1	I	LT114	Tamil – I	5	3
	II	LE115AT	English –I	5	2
	III	AM114C	Allied Mathematics -I	6	5
	III	CS120	Problem Solving Techniques	3	3
	III	CS121	Web Development Using HTML	4	4
	IV	VE105A/B	Christian Religion –I / Value Education –I	2	1
	IV	SK104	Communication Skills	2	1
	IV	CE103	Communicative English –I	-	1
	II	LE115AP	English Lab –I	-	1
	III	PCS108	Practical -I: Web Development Using HTML	3	2
2	I	LT214	Tamil –II	5	3
	II	LE215AT	English –II	5	2
	III	AM214C	Allied Mathematics –II	6	5
	III	CS221	Digital Computer Fundamentals	3	3
	III	CS222	Programming Using C	4	4
	IV	VE205A/B	Christian Religion –II / Value Education –II	2	1
	IV	SK204	Leadership Skills	2	1
	IV	CE203	Communicative English –II	-	1
	II	LE215AP	English Lab –II	-	1
	III	PCS212	Practical -II: Programming Using C	3	2
3	I	LT312	Tamil –III	5	3
	II	LE309T	English –III	5	2
	III	AP309B	Allied Physics for Computer Science I	4	3
	III	CS322	Computer Organization And Architecture	3	3
	III	CS323	Data Structures and Algorithms Using C	4	4
	IV	VE306	Human Rights	2	1
	IV	SK304	Technical Skills	2	1
	III	PCS309	Practical -III: Data Structures And Algorithms Using C	3	2
	II	LE309P	English Lab –III	-	1

Sem	Part	Subcode	Subtitle	Hours	Credits
4	I	LT411P / SS	Tamil -IV :Poem / Short Story	5	3
	II	LE409T	English –IV	5	2
	III	AP409B	Allied Physics For Computer Science II	4	3
	III	CS422	Software Engineering	3	3
	III	CS423	Relational Database Management Systems	4	4
	IV	VE406	Environmental Science	2	1
	IV	SK404	Employability Skills	2	1
	III	PAP409B	Allied Physics Practical's for Computer Science	2	1
	III	PCS412	Practical -IV: Relational Database Management Systems	3	2
	II	LE409P	English Lab –IV	-	1
	V	CO-SHE	Co-Curricular – Groups and Movements	-	2
	V	CO-DED	Co-Curricular – Outreach	-	2
5	III	CS540	Programming Using Java	4	4
	III	CS541	Web Development Using XML	4	4
	III	CS542	Programming Using PHP	3	3
	III	CS4543	Operating Systems	4	4
	III	CS544 A / B / C /D	Elective I : Computer Graphics / Data Mining And Warehousing / Decision Support System / Software Testing And Quality Assurance	4	4
	III	PCS515	Practical -V :Programming Using Java	3	2
	III	PCS516	Practical -VI :Web Development Using XML	3	2
	III	PCS517	Practical -VII :Programming Using PHP	3	2
	III		Non Major Elective -I	2	1
6	III	CS633	Mobile Applications Development	4	4
	III	CS634	Linux and Shell Programming	4	4
	III	CS635	Programming Using Python	3	4
	III	CS636	Microprocessor Using 8086/88	4	4
	III	CS637 A / B / C /D	Elective II :Computer Networks / Software Project Management / Security Systems / Cognitive Computing	4	4
	III	PCS627	Practical - VIII :Mobile Applications Development	3	2
	III	PCS628	Practical -IX :Programming Using Python	3	2
	III	PCS629	Practical -X :Linux and Shell Programming/Microprocessor Using 8086/88	3	2
	III	PCS630J	Project Work	-	4
	III		Non Major Elective II	2	1

I. DISTRIBUTION OF CREDITS FOR B.SC. (CS) PROGRAMME

Part	Subject	Credits Distribution (Semester Wise)						
		1	2	3	4	5	6	Total
I	Language I: Tamil	3	3	3	3	-	-	12
II	Language II: English Theory	2	2	2	2	-	-	8
	Language II: English Practical	1	1	1	1	-	-	4
III	Allied I: Mathematics	5	5	-	-	-	-	10
	Allied II: Physics Theory	-	-	3	3	-	-	6
	Allied II: Physics Practical	-	-	-	1	-	-	1
	Main : Theory	7	7	7	7	15	16	59
	Main : Theory – Elective	-	-	-	-	4	4	8
	Main : Practical	2	2	2	2	6	6	20
	Project Work	-	-	-	-	-	4	4
	Non Major Elective	-	-	-	-	1	1	2
IV	Skill Elective & Value Education	03	03	02	02	-	-	10
V	Co – Curricular	-	-	-	04	-	-	4
VI	Additional Credits	-	-	-	-	-	-	-
Total Credits								148

II. LEGEND OF THE CIPHER USED FOR THE COURSES

The cipher attributed for each course/subject in [L:P:C – CA:SE] shall be interpreted as [Lecture hours : Practical hours : Credits – Continuous Assessment Maximum mark : Semester Examination Maximum mark].

III. REGULATIONS FOR THEORY COURSES

1. Each theory course will have a maximum of 100 marks.
2. For a theory course, Continuous Assessment (CA) is 50 marks and Semester Examination is 50 marks.
3. There is no passing minimum for CA
4. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
1.	First CA	15
2.	Second CA	15
3.	Attendance	5
4.	MCQ Test Using MOODLE LMS	5
5.	Assignment/Open Book Test /Problem Solving	5
6.	Other Components (Seminars, Library reference works, Group Discussions, Field Visits and Quiz)	5
TOTAL		50

5. Question Paper Pattern for Continuous Assessment Tests

Time: 2 Hrs.

The Question Paper shall consist of three sections

S.No	CA Question Paper Pattern	Mark
1	Part- A (6 x 2 =12) Answer all Questions.	12
2	Part – B (3 x 6 =18) 3 Questions with internal choice (either or type)	18
3	Part – C (2 x 10 =20) Answer any two questions out of 3 questions.	20
Total		50

6. Question Paper Pattern for Semester

Time: 3 Hrs.

The Question Paper shall consist of three sections

S.No	Semester Question Paper Pattern	Marks
1	Part - A (10 x 2 = 20) Answer all Questions. Two questions from each unit.	20
2	Part - B (5 x 7 = 35) 5 Questions with internal choice (either or type).One question from each	35
3	Part - C (3 x15 = 45) Answer any three questions out of 5 questions. One question from eachunit.	45
Total		100

7. Blue Print of Theory Courses for Semester Examinations

Section	Description Type and Choice	Marks	Unit wise distribution					Total Questions in each Section
			Unit I	Unit II	UnitIII	UnitIV	Unit V	
A	Short Answer Questions NO CHOICE	Each Question Carries 2 Marks	2	2	2	2	2	10
B	Medium Answer Questions EITHER OR TYPE	Each Question Carries 7 Marks	1 Pair	1 Pair	1 Pair	1 Pair	1 Pair	5 Pairs
C	Long Answer Questions ANY THREE	Each Question Carries 15 Marks	1	1	1	1	1	5
Total Number of Questions			3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	15 + 5 Pair
TotalMarks			26	26	26	26	26	130

IV. REGULATIONS FOR PRACTICAL COURSES

- Each practical course will have a maximum of 100 marks.
- For a practical course, CA is 50 marks and Semester Examination is 50 marks.
- There is no passing minimum forCA
- The duration of semester practical examination is three hours. The student should submit a bonafide record of the experiments done at the time of the semester examination. The student shall not be allowed to appear for the semester examination without the bonafide record. The bonafide record should contain a certificate, program

list and source code listing of all the programs with outputs

5. The features of every programming language are listed in the syllabus; however, the students are expected to carry out several exercises in each feature of the programming language.

6. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
1.	CA For each practical paper, only one CA test will be conducted.	20
2.	Performance in the practical session Every practical session will carry a maximum of 10 marks and it is divided as follows: Initial Preparation & Observation : 5 marks. Debugging & Execution of Program : 5 marks. The students must prepare for the practical exercises by writing programs in the observation notebook. The observation notebook should be submitted for evaluation. Marks will be deducted for late as well as incomplete or incorrect submission. Ten marks will be awarded for each exercise subject to the successful completion of the entire exercise as directed by the staff concerned	20
3.	Attendance	5
4.	Module Development / Viva Voce / MCQ Test Using MOODLE LMS	5
TOTAL		50

7. Question pattern for practical examination

Time:3Hrs

Max. Marks: 40+10(forRecord) = 50

The question paper pattern for continuous assessment test is same as that the semester practical examination.

Each student will get a single question to be answered. The question may have subdivisions.

No more than three candidates should get the same question in a batch of 30 students.

Hence a question paper in practical should have 10 questions.

8. Blue Print of Practical Courses for the continuous Assessment tests and Semester Practical examination

Section	Description Type and Choice	Marks	Total Questions in each Section
A	Programming Questions	Each Question Carries 20 Marks	1 Pair
B	Programming Questions	Each Question Carries 20 Marks	1 Pair
Total Number of Questions			2 Pair
Total Marks			40

In each section, one question will be asked from the list of exercise completed in the practical. Another question will be a general question covering the features of the programming language.

V. REGULATIONS FOR PROJECT WORK COURSE

Learning Objective

The students are every given adequate exposure and opportunity to develop a full-pledged software according to his taste and ability.

Guidelines

- 1) A group of students can choose a problem related to application area or system software and solve it by presenting appropriate computer programs using any programming language.
- 2) During the course of V Semester, the student groups will be guided to select a topic for the project work that is scheduled for the VI semester.
- 3) A faculty member as a guide will be available to 2 groups of students. Maximum of 5 students can be placed in a group.
- 4) Students will be allotted a lot by the guide. Among the allotted students the group can be formed by the faculty.
- 5) Guidance will be given to plan the entire project and do system study. Actual coding and debugging using the machine must be done by the student's groups.
- 6) Evaluation of the project and allotting of marks and will be done under the components CA and semester.
- 7) Passing minimum is 40%. A failed candidate has to improve his project work and submit it.
- 8) Methodology of project work is to be taught in the V and the VI Semester.
- 9) Each student can submit a separate report for the evaluation purpose.
- 10) Two copies of the project report are to be prepared.
 - a. One will be kept by the candidate
 - b. The other copy is meant for the Library

11) Evaluation

CA Components			50 Marks
1.	First Review	10 Marks	
2.	Second Review	10 Marks	
3.	Implementation and Testing	30 Marks	
Total		50 Marks	
Semester Examinations			50 Marks
1.	Evaluation of Project Work Document	40 Marks	
2.	Viva – Voce	10 Marks	
Total		50 Marks	
Total			100 Marks

12) Project Report Contents

Requirement Analysis	10 Marks
Design	10 Marks
Implementation	10 Marks

Internal examiner and the external examiner will evaluate the project report separately and average will be calculated as the final semester mark of the student.

13) Project Plan

Project area	Work product
Project Management	Project Proposal
	Project Plan
	Project Review Record -1
Requirement analysis	System Study (SSD)
	Vision Document (VSD)
	Use Case Specification
	Project Review Record – 2
Design	Design Document
	Architectural Design
	Database Design
	Interface Design
	Procedural Design
	Test Case Design
First Review	Project Review Record -3
	Draft Report (Combination of all the above work products)
	PPT for project presentation
Implementation	Project Presentation
	Overview of the Project
	Pseudo Code (Algorithms)
Test	Project Review Record – 4
	Test Case Document
	Unit Testing
	Integration Testing
Second Review	System Testing
	Project Review Record – 5
	Draft Copy of the Project Report
	PPT for Project Presentation
	Project Presentation
	Application Demo

VI. DETAILED SYLLABI

Semester - I

[3:0:3-50:50]

PROBLEM SOLVING TECHNIQUES

1. Learning Objectives

- To develop problem solving skills with top down design principles.
- To become competent in algorithm design and program implementation.
- To develop skills to apply appropriate standard methods in problem solving

2. Blue Print of the Question Paper

Section	Unit-I	Unit-II	Unit-III	Unit-IV	Unit-V
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11. a) Theory (or)	12. a) Theory (or)	13. a) Theory (or)	14. a) Theory (or)	15.a) Theory (or)
	b) Theory	b) Theory	b) Algorithm	b) Algorithm	b) Algorithm
Section-C	16. Theory	17. Theory	18. Theory	19. Theory (or) Program	20. Theory (or) Program

3. Course Outline

UNIT – I: INTRODUCTION TO COMPUTER PROBLEM SOLVING

Introduction – Problem Solving Aspect – Implementation of Algorithms – Program verification – Efficiency of Algorithms – Analysis of Algorithms.

UNIT – II: FUNDAMENTAL ALGORITHMS

Exchanging the Values of Two Variables – Counting – Summation of a Set of Numbers – Factorial Computation – Generation of The Fibonacci Sequence – Base Conversion

UNIT – III: FACTORING METHODS

Finding the Square Root of a Number – Smallest Divisor of an Integer – GCD of Two Integer – Generating Prime Numbers – Generation of Pseudo-Random Numbers

UNIT – IV: ARRAY TECHNIQUES

Array Order Reversal – Finding Maximum Number in a Set – Removal of Duplicates from an Ordered Array

UNIT – V: MERGING, SORTING AND SEARCHING

Two-way Merge, Sorting by Exchange, Binary Search, Hash Searching.

4. Teaching Resources

i. Text

1. Dromey R G, “How to Solve it by Computer”, Dorling Kindersley India Pvt.Ltd, Pearson Education, 2007.

Unit - I : Ch. 1.1, 1.2, 1.4, 1.5, 1.6, 1.7

Unit - II : Ch. 2.1, 2.2, 2.3, 2.4, 2.6, 2.8.

Unit - III : Ch. 3.1, 3.2, 3.3, 3.4, 3.6.

Unit - IV : Ch. 4.1, 4.3, 4.4.

Unit - V : Ch. 5.1, 5.3, 5.7, 5.8

ii. References

1. Michael Schneider, Steven W. Weingart, David M. Perlman, “An Introduction to Programming and Problem Solving with Pascal”, Wiley Eastern Limited, New Delhi, 1982.
2. Harold Abelson and Gerald Sussman with Julie Sussman, “Structure and Interpretation of Computer Programs”, MIT Press, 1985.
3. Ronald A. Pasko, “Problem Solving Basics and Computer Programming”, Jones And Bartlett Publishers, 2nd Edition, 2001.

iii. Web references

Online Tutorial

1. <http://nptel.ac.in/courses/106104074/>
2. <http://javahungry.blogspot.com/2014/06/algorithm-problem-solving-techniques-or-approaches-for-software-programmer.html>

Online Quiz

1. https://www.tutorialspoint.com/cplusplus/cpp_online_quiz.htm
2. <http://www.withoutbook.com/OnlineTestStart.php?quizId=11>

Online Compiler

1. https://www.tutorialspoint.com/compile_cpp11_online.php
2. <https://www.codechef.com/ide>

5. Learning Outcomes

Upon Completing the Course, Students will be able to:

- Develop programming techniques required to solve a given problem.
- Develop problem solving skill using top – down design principles.
- Design an algorithm for a problem.
- Develop techniques to handle array structure
- Develop techniques such as searching and sorting

Web Development Using HTML

1. Learning Objectives

- To provide a comprehensive overview of the two largest Web technologies, Hyper Text Markup Language (HTML), and Cascading Style.
- To learn through hands-on, practical instruction that will assist the students to tackle the real-world problems they face in building websites today—with a specific focus on HTML and CSS
- To develop an ability to design and implement a web site

2. Extension of the Course:

Practical : Web Development using HTML

3. Blue Print of the QuestionPaper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a) Theory (OR) b) Program	12.a)Theory (OR) b) Program	13.a)Theory (OR) b) Program	14.a)Theory (OR) b) Program	15.a) Theory (OR) b) Program
Section-C	16.Program	17.Theory/P rogram	18.Program	19.Theory/ Program	20.Theory/ Program

4. Course Outline**Unit - I: HTML Basics, Formatting Tags and Lists**

HTML Introduction – Web page: Static & Dynamic Page - Web Browsers - HTML Editors - Tags – Elements – Attributes - HTML Page Structure - HTML Basic tags: Head – Title – Body. Basic text formatting: Heading tags – Paragraph tag – hr tag - Line break – Pre formatted. Presentational Element - Phrase Elements. List Tags: Ordered List – Unordered List – Definition List.

Unit – II: Links, Images and Tables

Link: Basic link – creating links. Image and Object: Adding images in a website – Adding other objects – Using images as links.

Tables: Basic table elements and attributes – Advanced tables.

Unit – III: Frames and Forms

Frames: The Frameset, No Frame Element - Creating Link between Frames - Nested Frameset. Form: Text Fields - Password Field - Radio Button – Checkbox - Submit Button – Reset Button – Button – Select – option – text area.

Unit – IV: Cascading Style Sheet-I

Introduction – syntax – ID selector - Class selector – External CSS – Internal CSS – Inline CSS – font property: font family - font size – font weight - font style - font variant - font stretch - font size adjust. Text Formatting: Color, text-align, vertical-align, decoration – indent- shadow –transform- letter spacing –word pacing- white space - direction.

Unit - V: Cascading Style Sheet-II

Background: color – image – repeat – position – attachment. List: style type – style position – style image – marker offset. Table: table specific – border collapse – border spacing – caption side – empty cell – table layout. Outlines: outline width – outline style – outline color.

5. Teaching Resources TextBook

i. Text

1. Jon Ducktt. “Web Programming with HTML, CSS andJAVA SCRIPT”, Wiley Publishing, 2005.

Unit– I :Ch.1

Unit– II : Ch. 2, 3 &4

Unit- III : Ch.5,6

Unit– IV :Ch.7

Unit- V :Ch.8

ii. References

1. Joel Skylar. “Principles of Web Design”. Singapore : Thomson Asia Pvt. Ltd 2000
2. Powell , Thomas A. “Web Design – The Complete Reference”, Tata McGraw Hill Edition2000
3. Alexis Goldstein, Louis Lazaris, Estelle Weyl. “HTML5 & CSS3 for the RealWorld”.

iii. Web References

(i) OnlineTutorial

1. <http://www.w3schools.com/css>
2. <http://www.tutorialspoint.com/css>

(ii) **OnlineQuiz**

1. [Http://www.Indiabix.com/online-test/](http://www.Indiabix.com/online-test/)

6. Learning Outcomes

Upon Completing the Course, Students will able to:

- Use knowledge of HTML and CSS code and an HTML editor to create personal and/or business websites following current professional and/or industry standards.
- Use critical thinking skills to design and create websites.

PRACTICAL - I: Web Development Using HTML

HTML

1. Heading Elements
2. Phrase Tags
3. Presentational Tags
4. Lists
5. Links
6. Images
7. Tables
8. Forms
9. Frames

CSS

10. Cascading Style Sheet

DIGITAL COMPUTER FUNDAMENTALS

1. Learning Objectives

- To explore the Number System, Number Conversion from one Base to another Base and Complements.
- To understand the Logic Gates, Boolean Algebra and to design the Logical Circuits.
- To simplify the Boolean Functions using K-Map Method
- To Learn Combinational circuits as Adders and Subtractors, Encoders and Decoders.
- To Learn the different types of Flip-Flops such as SR Flip flop, JK Flip flop, T Flip flop and D Flip flop .

2. Blue Print of the Question Paper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a)Theory (OR) b) Theory	12.a)Theory (OR) b) Theory	13.a)Theor y (OR) b) Theory	14.a)Theory (OR) b) Theory	15.a) Theory (OR) b) Theory
Section-C	16.Theory	17. Theory	18. Theory	19.Theory	20. Theory

3. Course Outline**Unit – I: Number System and Binary Arithmetic's**

Digital Computer and Digital System - Number Systems: Number Systems - Decimal, Binary, Octal, Hexadecimal - Conversion from one to another.

Characters and Codes: BCD, ASCII, 2421 Code, Excess-3 Code, Gray Code.

Binary Arithmetic's: Binary Addition, Subtraction, Multiplication, Division.

Complements: n's and n-1's Complements.

Unit – II: Logic Gates and Boolean Algebra

Logic Gates: AND, OR, NOT, NOR, NAND, XOR, XNOR Gates - Logic Circuits.

Boolean Algebra and Boolean Laws and Theorems - De Morgan's Theorems – Duality Theorem.

Unit – III: Map Simplification

Simplification of Sum of Product and Product of Sum Expressions - Karnaugh Map and Simplifications: Three Variable Maps, Four Variable Maps - Don't Care Condition.

Unit – IV: Combinational Circuits

Combinational Circuits: Half and Full Adders – Half Subtractor and Full Subtractor - Encoders and Decoders – Multiplexers – De-multiplexers.

Unit – V: Flip Flops and Sequential Circuits

Sequential Logic Design: Flip-Flops - SR, JK, D and T Flip-Flops – Edge Triggered Flip-Flop – Master-Slave Flip-Flop – Flip-flop Excitation table.

4. Teaching Resources

i. TextBooks

1. Morris M Mano, “Digital Logic and Computer Design”, Prentice Hall of India Pvt. Lmt., New Delhi 2001.

Unit - I : Chap. 1.1 to 1.8

Unit - II : Chap. 2.1 to 2.7

Unit - III : Chap. 3.1 to 3.3, 3.5, 3.8

Unit - IV : Chap. 4.3, 4.4, 5.5, 5.6

Unit - V : Chap. 6.1 to 6.3 & 6.6

ii. References

1. Morris M Mano, “Computer System Architecture”, Prentice Hall of India Pvt.Lmt., New Delhi, 1991.
2. Donald P. Leach and Albert Paul Malvino, “Digital Principles and Application”, Fifth Edition, Tata McGraw-Hill Publishing Company Ltd., New Delhi, 2003.
3. Thomas C. Bartee, “Computer Architecture and Logic Design”, McGraw Hill International Edition, New Delhi, 1991.

iii. Web References

i) Online Tutorial

1. <https://www.geeksforgeeks.org/introduction-of-logic-gates/>
2. https://www.tutorialspoint.com/computer_logical_organization/logic_gates.ht

m

ii) Online Quiz:

1. <https://www.avatto.com/computer-science/test/mcqs/digital-electronics/questions/90/1.html>
2. <https://www.geeksforgeeks.org/digital-logic-number-representation-gq/>

5. Learning Outcomes

- Perform conversions among different number systems, to be familiar with basic logic gates,
- Draw the Logic circuits and truth table for Boolean functions
- Simplify Boolean functions by using k-map method and Boolean Laws and Theorems.
- Design of combinational circuits such as Adder, Subtractor, Multiplexer, Encoder and Decoder etc.
- Understand the design of sequential Circuits such as Flip-Flops, Edge-trigger and master slave flip flops.

Programming Using C

1. Learning Objectives

- To enhance analyzing and problem-solving skills and use the same for writing programs in C.
- To develop logics which will help them to create programs, applications in C.
- To use the comparisons and limitations of the various programming constructs and choose the right one for the task in hand.
- To enter the program on a computer, edit, compile, debug, correct, recompile and run it.

2. Extension of the Course:

Practical: Programming using C

3. Blue Print of the Question Paper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a) Theory (OR) b) Program	12.a) Theory (OR) b) Program	13.a) Theory (OR) b) Program	14.a) Theory (OR) b) Program	15.a) Theory (OR) b) Program
Section-C	16.Theory/ Program	17.Program	18.Program	19.Theory / Program	20.Program

4. Course Outline**Unit – I: Data types, Operators and Structures**

History of C - Structure of a C program – Constants and Variables - Basic data types (int, float, char, double, void) – operators and expressions (arithmetic operators, relational operators, logical operators, assignment operator, Increment and decrement operator, conditional operator, bitwise operators, mapping input output operator) – Control Constructs (if, if/else, switch, while, do...while, for), break and continue, exit() function, goto and label, The ?:operator.

Unit – II: Arrays and Functions

Arrays (declaration, one and two dimensional arrays) - Character Arrays and Strings. Function Fundamentals (General form, Function Definition, Function arguments, return value) – Parameter passing: call-by-value and call-by-reference – Recursion – Passing Arrays to Function – Passing Strings to Function.

Unit – III: Storage Classes, Structures and Unions

Scope rules (Local variables and global variables, scope rules of functions) -Type modifiers and storage class specifiers.

Structures – Basics of Structure – Declaring of Structure – Referencing Structure elements - Array of Structures – Nesting of Structures - Passing Structures to function – Pointers and Structures - Unions.

Unit – IV: Pointers

Understanding Pointers – Accessing the Address of a Variable – Declaring the Pointer Variables – Initialization of Pointer Variables – Accessing a Variable through its Pointer – Pointer Expressions – Pointers and Arrays – Pointers and Character Strings – Array of Pointers – Pointers as Function Arguments – Functions returning Pointers – Pointers to Functions.

Unit – V: File Management in C

Introduction – Defining and Opening a File – Closing a File – Input / Output Operations on Files –

5. Teaching Resources

i. Text

1. E.Balagurusamy, “Programming in ANSI C”, Seventh Edition, McGraw Hill Education Private Limited, NewDelhi:2016.

Unit - I : Chap. 2to 7

Unit - II : Chap. 8 to 10

Unit - III : Chap. 12

Unit - IV : Chap. 10& 11

Unit - V : Chap. 13

ii. References

1. YashavantKanetkar, “Let us C”, BPB Publications, Tenth Edition - New Delhi:2010
2. Ashok N.Kamthane, “Programming in C”, Second Impression, Pearson:2012.

iii. Web References

(i) Online Tutorial

1. <http://www.c4learn.com/?gclid=COK1y6nHk7wCFcUA4godmlg>

AKA/

2. <http://www.cprogramming.com/tutorial/c-tutorial.html/>
3. <http://www.tutorialspoint.com/cprogramming/>

(ii) OnlineQuiz

1. <http://www.indiabix.com/online-test/c-programming-test/>
2. <http://www.pskills.org/c.jsp/>

(iii) OnlineCompiler

1. https://www.tutorialspoint.com/compile_c_online.php

6. LearningOutcomes

After course completion the students will have the following learning outcomes:

- Understanding a functional hierarchical code organization.
- Ability to define and manage data structures based on problem subject domain.
- Ability to work with textual information, characters and strings.
- Ability to work with arrays, structures, pointers and files.

VII. DISTRIBUTION OF CREDITS FOR B.SC. (CS) PROGRAMME

Part	Subject	Credits Distribution (Semester Wise)						
		1	2	3	4	5	6	Total
I	Language I: Tamil	3	3	3	3	-	-	12
II	Language II: English Theory	2	2	2	2	-	-	8
	Language II: English Practical	1	1	1	1	-	-	4
III	Allied I: Mathematics	5	5	-	-	-	-	10
	Allied II: Physics Theory	-	-	3	3	-	-	6
	Allied II: Physics Practical	-	-	-	1	-	-	1
	Main : Theory	7	7	7	7	15	16	59
	Main : Theory – Elective	-	-	-	-	4	4	8
	Main : Practical	2	2	2	2	6	6	20
	Project Work	-	-	-	-	-	4	4
	Non Major Elective	-	-	-	-	1	1	2
IV	Skill Elective & Value Education	03	03	02	02	-	-	10
V	Co – Curricular	-	-	-	04	-	-	4
VI	Additional Credits	-	-	-	-	-	-	-
Total Credits								148

VIII. LEGEND OF THE CIPHER USED FOR THE COURSES

The cipher attributed for each course/subject in [L:P:C – CA:SE] shall be interpreted as [Lecture hours : Practical hours : Credits – Continuous Assessment Maximum mark : Semester Examination Maximum mark].

IX. REGULATIONS FOR THEORY COURSES

5. Each theory course will have a maximum of 100 marks.
6. For a theory course, Continuous Assessment (CA) is 50 marks and Semester Examination is 50 marks.
7. There is no passing minimum for CA
8. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
8.	First CA	15
9.	Second CA	15
10.	Attendance	5
11.	MCQ Test Using MOODLELMS	5
12.	Assignment/Open Book Test /Problem Solving	5
13.	Other Components (Seminars, Library reference works, Group Discussions, Field Visits and Quiz)	5
TOTAL		50

5. Question Paper Pattern for Continuous Assessment Tests

Time: 2 Hrs.

The Question Paper shall consist of three sections

S.No	CA Question Paper Pattern	Mark
1	Part- A (6 x 2 =12) Answer all Questions.	12
2	Part – B (3 x 6 =18) 3 Questions with internal choice (either or type)	18
3	Part – C (2 x 10 =20) Answer any two questions out of 3 questions.	20
Total		50

6. Question Paper Pattern for Semester

Time: 3 Hrs.

The Question Paper shall consist of three sections

S.No	Semester Question Paper Pattern	Marks
1	Part - A (10 x 2 = 20) Answer all Questions. Two questions from each unit.	20
2	Part - B (5 x 7 = 35) 5 Questions with internal choice (either or type). One question from each	35
3	Part - C (3 x 15 = 45) Answer any three questions out of 5 questions. One question from each unit.	45
Total		100

14. *Blue Print of Theory Courses for Semester Examinations*

Section	Description Type and Choice	Marks	Unit wise distribution					Total Questions in each Section
			Unit I	Unit II	Unit III	Unit IV	Unit V	
A	Short Answer Questions NO CHOICE	Each Question Carries 2 Marks	2	2	2	2	2	10
B	Medium Answer Questions EITHER OR TYPE	Each Question Carries 7 Marks	1 Pair	1 Pair	1 Pair	1 Pair	1 Pair	5 Pairs
C	Long Answer Questions ANY THREE	Each Question Carries 15 Marks	1	1	1	1	1	5
Total Number of Questions			3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	15 + 5 Pair
Total Marks			26	26	26	26	26	130

X. REGULATIONS FOR PRACTICAL COURSES

7. Each practical course will have a maximum of 100 marks.
8. For a practical course, CA is 50 marks and Semester Examination is 50 marks.
9. There is no passing minimum for CA
10. The duration of semester practical examination is three hours. The student should submit a bonafide record of the experiments done at the time of the semester examination. The student shall not be allowed to appear for the semester examination without the bonafide record. The bonafide record should contain a certificate, program

list and source code listing of all the programs with outputs

11. The features of every programming language are listed in the syllabus; however, the students are expected to carry out several exercises in each feature of the programming language.

12. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
5.	CA For each practical paper, only one CA test will be conducted.	20
6.	Performance in the practical session Every practical session will carry a maximum of 10 marks and it is divided as follows: Initial Preparation & Observation : 5 marks. Debugging & Execution of Program : 5 marks. The students must prepare for the practical exercises by writing programs in the observation notebook. The observation notebook should be submitted for evaluation. Marks will be deducted for late as well as incomplete or incorrect submission. Ten marks will be awarded for each exercise subject to the successful completion of the entire exercise as directed by the staff concerned	20
7.	Attendance	5
8.	Module Development / Viva Voce / MCQ Test Using MOODLE LMS	5
TOTAL		50

7. Question pattern for practical examination

Time:3Hrs

Max. Marks: 40+10(forRecord) = 50

The question paper pattern for continuous assessment test is same as that the semester practical examination.

Each student will get a single question to be answered. The question may have subdivisions.

No more than three candidates should get the same question in a batch of 30 students.

Hence a question paper in practical should have 10 questions.

8. Blue Print of Practical Courses for the continuous Assessment tests and Semester Practical examination

Section	Description Type and Choice	Marks	Total Questions in each Section
A	Programming Questions	Each Question Carries 20 Marks	1 Pair
B	Programming Questions	Each Question Carries 20 Marks	1 Pair
Total Number of Questions			2 Pair
Total Marks			40

In each section, one question will be asked from the list of exercise completed in the practical. Another question will be a general question covering the features of the programming language.

XI. REGULATIONS FOR PROJECT WORK COURSE

Learning Objective

The students are every given adequate exposure and opportunity to develop a full-pledged software according to his taste and ability.

Guidelines

- 14) A group of students can choose a problem related to application area or system software and solve it by presenting appropriate computer programs using any programming language.
- 15) During the course of V Semester, the student groups will be guided to select a topic for the project work that is scheduled for the VI semester.
- 16) A faculty member as a guide will be available to 2 groups of students. Maximum of 5 students can be placed in a group.
- 17) Students will be allotted a lot by the guide. Among the allotted students the group can be formed by the faculty.
- 18) Guidance will be given to plan the entire project and do system study. Actual coding and debugging using the machine must be done by the student's groups.
- 19) Evaluation of the project and allotting of marks and will be done under the components CA and semester.
- 20) Passing minimum is 40%. A failed candidate has to improve his project work and submit it.
- 21) Methodology of project work is to be taught in the V and the VI Semester.
- 22) Each student can submit a separate report for the evaluation purpose.
- 23) Two copies of the project report are to be prepared.
 - a. One will be kept by the candidate
 - b. The other copy is meant for the Library
- 24) Evaluation

CA Components			50 Marks
4.	First Review	10 Marks	
5.	Second Review	10 Marks	
6.	Implementation and Testing	30 Marks	
Total		50 Marks	
Semester Examinations			50 Marks
1.	Evaluation of Project Work Document	40 Marks	
2.	Viva – Voce	10 Marks	
Total		50 Marks	
Total			100 Marks

- 25) Project Report Contents

Requirement Analysis	10 Marks
Design	10 Marks
Implementation	10 Marks

Testing and Documentation
Total

10 Marks
40 Marks

Internal examiner and the external examiner will evaluate the project report separately and average will be calculated as the final semester mark of the student.

26) Project Plan

Project area	Work product
Project Management	Project Proposal
	Project Plan
	Project Review Record -1
Requirement analysis	System Study (SSD)
	Vision Document (VSD)
	Use Case Specification
	Project Review Record – 2
Design	Design Document
	Architectural Design
	Database Design
	Interface Design
	Procedural Design
	Test Case Design
First Review	Project Review Record -3
	Draft Report (Combination of all the above work products)
	PPT for project presentation
Implementation	Project Presentation
	Overview of the Project
	Pseudo Code (Algorithms)
Test	Project Review Record – 4
	Test Case Document
	Unit Testing
	Integration Testing
Second Review	System Testing
	Project Review Record – 5
	Draft Copy of the Project Report
	PPT for Project Presentation
Application Demo	Project Presentation
	Application Demo

XII. DETAILED SYLLABI

[SEMESTER-III]

[3:0:3-50:50]

COMPUTER ORGANIZATION AND ARCHITECTURE

1. Learning Objectives

- To understand the basics of Computer Organization.
- To know the relationship between computer instruction and the Machine code execution.
- To know about the various types of CPU Organization and Addressing Modes.

- To recognize the need of interface between CPU and Input / Output devices.
- To think critically, independently, and quantitatively about Computer Memory.

2. Blue Print of the Question Paper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a)Theory (OR) b) Theory	12.a)Theory (OR) b) Theory	13.a)Theory (OR) b) Theory	14.a)Theory (OR) b) Theory	15.a) Theory (OR) b) Theory
Section-C	16.Theory	17. Theory	18. Theory	19.Theory	20. Theory

3. Course Outline

Unit I. Computer Organization and Design

Instruction Codes - Computer Registers - Computer Instructions – Timing and Control – Instruction Cycle - Memory Reference Instructions.

Unit II. Programming the Basic Computer

Introduction - Machine language - Assembly language - The assembler - Program loops - Programming arithmetic and logical operation – Subroutines - Input-output programming.

Unit III. Central Processor Unit

Introduction – General Register Organization – Stack Organization – Instruction Formats – Addressing Modes.

Unit - IV: Input / Output Organization

Peripheral Devices – I/O interface – Asynchronous Data Transfer – Modes of Transfer - Direct Memory Access .

Unit - V: Memory Organization

Memory Hierarchy – Main Memory - Associative Memory – Cache Memory – Virtual Memory.

4. Teaching Resources

i. Text

1. Morris Mano M. “Computer System Architecture”. New Delhi: Prentice Hall of India Private Limited, 2011

Unit- I : Ch. 5.1 –5.6

Unit- II : Ch. 6.1 –6.8

Unit- III : Ch. 8.1 –8.5

Unit- IV : Ch. 11.1 – 11.4 & 11.6

Unit- V : Ch. 12.1, 12.2 & 12.4 -12.6

ii. References

1. William Stallings. “Computer Organization and Architecture”. 8th edition. Pearson Publication, 2010

2. Morris Mano. “Digital Login and Computer Design”. New Delhi: Prentice Hall of India Private Limited, 2001.

iii. Web References

(i) Online Tutorial

1. www.onlinevideolecture.com/computer.../computer-architecture

2. www.computer-pdf.com/architecture/

3. www.tutorialspoint.com/computer_logical_organization

(ii) Online Quiz

1. <https://www.pritee.org/index.php/knowledge-base-articles/computer-organisation-and-architecture/30-computer-organization-and-architecture-quiz-1>

2. <https://www.geeksforgeeks.org/computer-organization-and-architecture-gg/>

3. <https://www.sanfoundry.com/1000-computer-organization-architecture-questions-answers/>

5. Learning Outcomes

- Study basic computer organization, design and micro-operations.
- Prepare machine code from the instructions
- Understand CPU organization and different types of addressing modes.
- Understand how the Input/ Output devices communicate with the computer
- Learn various methods and techniques of memory organization.

Data Structures and Algorithms Using C

1. Learning Objective

1. To provide the knowledge of basic data structures and their implementations.
2. To understand importance of data structures in context of writing efficient programs.
3. To develop skills to apply appropriate data structures in problem solving

2. Blue Print of the Question Paper

	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.Theory (OR) Theory	12.Theory (OR) Theory	13.Theory (OR) Theory	14.Theory (OR) Program	15.Theory (OR) Program
Section-C	16.Theory (OR) Program	17.Theory (OR) Program	18.Theory (OR) Program	19.Theory (OR) Theory	20.Theory (OR) Theory

3. Course Outline**Unit - I: Arrays and Linked List**

Arrays: Characteristics of Array-One dimensional Array-Operation with Array: Insertion, Deletion and Sorting-Manipulation of using pointer-Representation of Sparse matrix

Linked list: Important terms-Implementation of linked List-Memory allocation and De-allocation-Operation on linked list-Singly Linked list: Insertion, concatenation, Splitting- Circular linked list-Doubly linked list.

Unit - II: Stack and Queue

Stack: Related terms-stack implementation-Operation on stack-Pointer and Stack-Representation of Arithmetic expression: Infix, Prefix, and Postfix notations-Application of Stack.

Queue: Various positions of Queue-Queue implementation-Operation on Queue-Disadvantages of Simple Queues-Dynamic implementation (Pointer),Insertion and Deletion operation-Types of Queues-Application of Queues.

Unit - III: Trees

Trees:Basic Terms-Binary Trees-Binary Tree Representation-Operation on Binary Tree-Traversal of a Binary Tree-Binary Search Tree.

Unit - IV: Searching and Sorting

Searching Techniques: Searching- Linear (Sequential) Search-Binary Search.

Sorting Techniques: Sorting-Insertion sort-Selection sort-Bubble Sort-Quick sort.

Unit - V: Graph

Graphs:Terminologies of Graphs-Graphs Representation-Traversal of Graphs-Breadth First Search-Depth First Search.

4. Teaching Resources

i. Text

1. Ashok N.Kamthne, “ Introduction to Data Structure in C “ Pearson Education:2005(Singapore)

Unit-I : Ch. 2.1-2.5, 2.10, 2.11-2.16, 6.1-6.4, 6.6, 6.12-6.24, 6.26, 6.27

Unit-II : Ch. 4.1-4.10, 5.1-5.9

Unit-III : Ch. 8.1-8.3, 8.7-8.12

Unit-IV :Ch.11.1-11.4, 10.1-10.6

Unit-V : Ch. 9.1-9.6

ii. References

1. SeymoreLipshutz. “Theory problems of Data structure”. Schaum’s outline series, New Delhi:McGraw Hill Book Company,1986
2. Horowitz E and Shani S. “Fundamentals of Data structure in C”, Hyderabad: UNIVERSITIES Press(India)Pvt.Ltd.,2008

iii. Web References

(i) Online Tutorial

1. [http://www. Cprogramming.com/algorithms-and-data-structures.html](http://www.Cprogramming.com/algorithms-and-data-structures.html)
2. <http://www. Tutorialspoint.com>
3. <http://www. ece.uwaterloo.ca/~dwharder/aads/Lecture materials/>

(ii) Online Quiz

1. <http://www.tcyonline.com/tests/data-structure-test>
2. <http://www.pskills.org/c.jsp>

(iii) Online Compiler

1. <https://www.onlinegdb.com/Sy-fU7gJW>

5. Learning outcomes

Upon Completing the Course, Students will able to:

1. Learn the basic types for data structure, implementation and application.
2. Know the strength and weakness of different data structures.
3. Use the appropriate data structure for a given problem.
4. Develop programming skills required to solve a given problem.

PRACTICAL - III: DATA STRUCTURES AND ALGORITHMS USING C

1. Matrix representation and Manipulation
2. Sparse Matrix representation and Transpose
3. Stack Representation and Manipulation
4. Queue Representation and Manipulation
5. Linked List Representation and Manipulation
6. Doubly Linked List Representation and Manipulation
7. Binary Tree Representation and Manipulation.
8. Sorting Algorithms
9. Searching Algorithms
10. Graph Representation and Traversals

SOFTWARE ENGINEERING

1. Learning Objectives

Upon completion of this course, students should be able to:

- Understand the principles of large scale software systems, and the processes that are used to build them.
- Acquire ability to the software-development process, including requirements analysis, design, programming, testing and maintenance.
- Understand the Communication issues in large, complex software projects.
- Understand purpose and importance of the project management from the perspective of planning, tracking and completion of project.

2. Course Outline**Unit - I: Software Process**

The Software Engineering – Software Process – Process Model – Prescriptive Models – Specialized Models – Unified Process – Personal Software Process – Team Software Process – Agile Process – Extreme Programming.

Unit - II: Modeling I

Requirement Engineering – Establishing –Eliciting Requirements – Developing use cases – Building Requirements Model – Negotiating and Validating Requirements – Requirement Analysis- Scenario Based Modeling – UML Models – Data Modeling concept – Class Based Modeling – Requirement Modeling – Flow oriented Modeling – Behavioral Model – Design Process – Design Models.

Unit - III: Modeling II

Software Architecture – Architecture Styles – Architectural Design – Architectural Mapping using Data Flow – Component – Designing class based component – Using traditional components – User Interface Design – The Golden Rules - User interface Analysis and Design – Interface Analysis – Design Steps.

Unit - IV: Quality Management

Software Quality – Achieving software Quality – Software Quality Assurance, Tasks, Goals and Metrics – Software Reliability – Software Testing Strategies: A Strategic Approach – Strategic Issues – Test Strategies for Conventional Software – System Testing- Validation Testing – The Art of Debugging – Software testing fundamentals –White box testing: Basis Path Testing – Control structure Testing – Black box testing – Model based testing.

Unit - V: Managing Software Projects

The Management Spectrum - People – The Product – Process – The Project – The W5HH Principle – Critical Practices – Basic Concepts – Project Scheduling – Defining a Task Network Scheduling – Software Risk – Risk Identification – Risk Projection – Risk Refinement – Risk Mitigation, Monitoring and Management – The RMMM Plan.

3. Teaching Resources**i. Text**

1. Pressman, Roger S. Software Engineering a practitioner's Approach 7th Edition. New York: McGraw Hill International Edition, 2010.

Unit-I : Ch. 1.1-1.6, 2.1-2.6, 3.1-3.4.

Unit-I :Ch 5, 6, 7, 8.2, 8.4.

Unit-III: Ch. 9, 10.1-10.3, 10.5-10.7, 11.1-11.4.

Unit-IV: Ch. 14, 16.1-16.3, 16.6, 17.1-17.3, 17.6-17.8, 18.1-18.7.

Unit-V: Ch. 24, 27, 28.

ii. References

1. Rajib Mall. Fundamentals of Software Engineering. New Delhi PHI Learning Pvt Ltd., 2009
2. James K.L Software Engineering New Delhi: PHI Learning Pvt Ltd., 2009.

iii. Web References

(i) Online Tutorial

1. <http://www.scribd.com/doc/27252883/Software-Engineering-Notes>
2. <http://www.People.cs.missouri.edu/~duanye/cs4320/lectures.html>
3. <http://www.Engineeringppt.blogspot.in/2011/12/pressman-software-engineering-ppt-pdf.html>.

4. Learning Outcomes.

Upon completion of this course, students should be able to:

- Plan and deliver an effective software engineering process, based on knowledge of widely used development lifecycle models.
- Employ group working skills including general organization, planning and time management and inter-group negotiation.
- Capture, document and analyze requirements.
- Translate a requirements specification into an implementable design, following a structured and organized process.
- Make effective use of UML, along with design strategies such as defining a software architecture, separation of concerns and design patterns.
- Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing.
- Evaluate the quality of the requirements, analysis and design work done during the module.

RELATIONAL DATABASE MANAGEMENT SYSTEMS

1. Learning Objectives

- To understand the basic principles of Databases and Data Models.
- To know about the Relational Data Structures and Relational Algebra.
- To understand the concepts of Functional Dependency and Normalization.
- To learn the features and to write Queries using SQL.
- To explore the organization and to acquire skills in developing programs using PL/SQL.

2. Blue Print of the Question Paper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a) Theory (OR) b) Theory	12.a) Theory (OR) b) Theory	13.a) Theory (OR) b) Theory	14.a) Program (OR) b) Theory	15.a) Program (OR) b) Theory
Section-C	16.Theory	17.Theory	18.Theory	19.Program	20.Program

3. Course Outline**Unit - I: Basic Concepts and Data Models**

Basic concepts and definition – Data Dictionary – Database System – Database Administrator – File Oriented System Vs Database System: Advantage and Disadvantage. Three level Database Architecture – Data Independence – Data Model: Physical Data model - Hierarchical Data model – Network Data Model.

Unit - II: Relational Model

Structure of Relational Model – Relational Algebra - Entity Relationship Model: Basic E-R Concepts - ER Diagram Symbols.

Unit - III: Relational Database Design

Functional Dependency: Functional Dependency Diagram and Example – Full Functional Dependency. Decomposition: Lossy-Join Decomposition – Lossless-Join Decomposition. Normalization: Normalization - First Normal Form – Second Normal Form –Third Normal Form – Boyce Codd Normal Form.

Unit - IV: Structured Query Language (SQL)

Creating, Dropping and Altering Tables – Create Table – Drop Table – Alter Table – Inserting Rows – Querying the Database – Simple Select Statement Sub-Selects – Aggregate Functions – String, Number and Date Functions – SET Operations – Views – Create View – Drop View – Modifying the Database – Insert – Update – Delete Statements.

Unit - V: Procedural Language – SQL (PL/SQL)

Data Types and Variables – Program Control Statements – Null Statement – Assignment Statement – Conditional Statements – Loops – Program Structure – Anonymous Blocks – Procedures and Functions – Stored Procedures and Functions – Packages – Triggers – Database Access using Cursors.

4. Teaching Resources

i. Text

1. S.K. Singh, “Database Systems - Concept, Design and Applications”, Dorling Kindersley (India) Pvt. Ltd., Second Impression,2008.

Unit - I : 1.1 – 1.8(1.8.1, 1.8.2, 1.8.5, 1.8.6)& 2.3 – 2.7(2.7.3, 2.7.4, 2.7.5)

Unit - II : 4.1- 4.4& 6.1 - 6.5

Unit - III : 9.1 - 9.3& 10.1 – 10.4

2. RajeshkharSunderraman. Oracle 8 Programming A Primer. New Delhi :Addition - Wesley publication, 2000.

Unit - IV : 2.1 – 2.6

Unit -V : 4.1 – 4.8

ii. References

1. Bipin C Desai, “An Introduction to Database Systems”, Galgotia Publications, New Delhi, 1999.
2. Abraham Siberscha, et al. Database System Concepts. McGraw Hill.
3. Ramez Elmasriand Navathe, Shamkant B. Fundamentals of Database Systems.Pearson Education.

iii. Web References

i) Online Tutorial

1. <https://www.javatpoint.com/dbms-tutorial>
2. <https://www.tutorialspoint.com/dbms/index.htm>
3. <http://www.w3schools.com/sql/>

ii) Online Quiz

1. <https://www.avatto.com/computer-science/test/mcqs/questions-answers/database/71/1.html>
2. <https://www.geeksforgeeks.org/dbms-gq/er-and-relational-models-gq/>
3. <https://www.geeksforgeeks.org/dbms-gq/sql-gq/>
4. <https://www.geeksforgeeks.org/dbms-gq/database-design-normal-forms-gq/>

5. LearningOutcomes

- Gain a good understanding of the architecture functioning of database management systems as well as associated tools and techniques.
- Implement the Entity Relationship Diagram using various E-R Diagram Symbol.
- Develop a good database design using normalization techniques.
- Understand the use of structured query language & PL/SQL, its syntax, its working and its scope.
- Acquire a good understanding of database systems concepts and to be in a position to use and design databases for different applications.

PRACTICAL - IV: RELATIONAL DATABASE MANAGEMENT SYSTEMS

SQL

1. Creating, Altering and Dropping aTable.
2. Manipulating a Table with Selection, Projection andOrdering.
3. Manipulating a Table with Aggregate, Numeric, String and Date Functions.
4. Creating, Manipulating and Dropping withViews.
5. Manipulation of Nested Queries andSub-Queries.

PL/SQL

6. Program ControlStatements.
7. Functions andProcedures.
8. Triggers.
9. Packages.
10. Cursors.

XIII. LEGEND OF THE CIPHER USED FOR THE COURSES

The cipher attributed for each course/subject in [L:P:C – CA:SE] shall be interpreted as [Lecture hours : Practical hours : Credits – Continuous Assessment Maximum mark : Semester Examination Maximum mark].

XIV. REGULATIONS FOR THEORY COURSES

9. Each theory course will have a maximum of 100 marks.
10. For a theory course, Continuous Assessment (CA) is 50 marks and Semester Examination is 50 marks.
11. There is no passing minimum for CA
12. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
15.	First CA	15
16.	Second CA	15
17.	Attendance	5
18.	MCQ Test Using MOODLELMS	5
19.	Assignment/Open Book Test /Problem Solving	5
20.	Other Components (Seminars, Library reference works, Group Discussions, Field Visits and Quiz)	5
TOTAL		50

5. Question Paper Pattern for Continuous Assessment Tests

Time: 2 Hrs.

The Question Paper shall consist of three sections

S.No	CA Question Paper Pattern	Mark
1	Part- A (6 x 2 =12) Answer all Questions.	12
2	Part – B (3 x 6 =18) 3 Questions with internal choice (either or type)	18
3	Part – C (2 x 10 =20) Answer any two questions out of 3 questions.	20
Total		50

6. Question Paper Pattern for Semester

Time: 3 Hrs.

The Question Paper shall consist of three sections

S.No	Semester Question Paper Pattern	Marks
1	Part - A (10 x 2 = 20) Answer all Questions. Two questions from each unit.	20
2	Part - B (5 x 7 = 35) 5 Questions with internal choice (either or type).One question from each	35
3	Part - C (3 x15 = 45) Answer any three questions out of 5 questions. One question from eachunit.	45
Total		100

21. *Blue Print of Theory Courses for Semester Examinations*

Section	Description Type and Choice	Marks	Unit wise distribution					Total Questions in each Section
			Unit I	Unit II	UnitIII	UnitIV	Unit V	
A	Short Answer Questions NO CHOICE	Each Question Carries 2 Marks	2	2	2	2	2	10
B	Medium Answer Questions EITHER OR TYPE	Each Question Carries 7 Marks	1 Pair	1 Pair	1 Pair	1 Pair	1 Pair	5 Pairs
C	Long Answer Questions ANY THREE	Each Question Carries 15 Marks	1	1	1	1	1	5
Total Number of Questions			3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	3 + 1Pair	15 + 5 Pair
TotalMarks			26	26	26	26	26	130

XV. REGULATIONS FOR PRACTICAL COURSES

13. Each practical course will have a maximum of 100 marks.
14. For a practical course, CA is 50 marks and Semester Examination is 50 marks.
15. There is no passing minimum forCA
16. The duration of semester practical examination is three hours. The student should submit a bonafide record of the experiments done at the time of the semester examination. The student shall not be allowed to appear for the semester examination without the bonafide record. The bonafide record should contain a certificate, program

list and source code listing of all the programs with outputs

17. The features of every programming language are listed in the syllabus; however, the students are expected to carry out several exercises in each feature of the programming language.

18. Evaluation Scheme for Continuous Assessment

S.NO.	CA Component	Marks
9.	CA For each practical paper, only one CA test will be conducted.	20
10.	Performance in the practical session Every practical session will carry a maximum of 10 marks and it is divided as follows: <p style="text-align: center;">Initial Preparation & Observation : 5 marks. Debugging & Execution of Program : 5 marks.</p> The students must prepare for the practical exercises by writing programs in the observation notebook. The observation notebook should be submitted for evaluation. Marks will be deducted for late as well as incomplete or incorrect submission. Ten marks will be awarded for each exercise subject to the successful completion of the entire exercise as directed by the staff concerned	20
11.	Attendance	5
12.	Module Development / Viva Voce / MCQ Test Using MOODLE LMS	5
TOTAL		50

7. Question pattern for practical examination

Time:3Hrs

Max. Marks: 40+10(forRecord) = 50

The question paper pattern for continuous assessment test is same as that the semester practical examination.

Each student will get a single question to be answered. The question may have subdivisions.

No more than three candidates should get the same question in a batch of 30 students.

Hence a question paper in practical should have 10 questions.

8. Blue Print of Practical Courses for the continuous Assessment tests and Semester Practical examination

Section	Description Type and Choice	Marks	Total Questions in each Section
A	Programming Questions	Each Question Carries 20 Marks	1 Pair
B	Programming Questions	Each Question Carries 20 Marks	1 Pair
Total Number of Questions			2 Pair
Total Marks			40

In each section, one question will be asked from the list of exercise completed in the practical. Another question will be a general question covering the features of the programming language.

XVI. REGULATIONS FOR PROJECT WORK COURSE

Learning Objective

The students are every given adequate exposure and opportunity to develop a full-pledged software according to his taste and ability.

Guidelines

- 27) A group of students can choose a problem related to application area or system software and solve it by presenting appropriate computer programs using any programming language.
- 28) During the course of V Semester, the student groups will be guided to select a topic for the project work that is scheduled for the VI semester.
- 29) A faculty member as a guide will be available to 2 groups of students. Maximum of 5 students can be placed in a group.
- 30) Students will be allotted a lot by the guide. Among the allotted students the group can be formed by the faculty.
- 31) Guidance will be given to plan the entire project and do system study. Actual coding and debugging using the machine must be done by the student's groups.
- 32) Evaluation of the project and allotting of marks and will be done under the components CA and semester.
- 33) Passing minimum is 40%. A failed candidate has to improve his project work and submit it.
- 34) Methodology of project work is to be taught in the V and the VI Semester.
- 35) Each student can submit a separate report for the evaluation purpose.
- 36) Two copies of the project report are to be prepared.
 - a. One will be kept by the candidate
 - b. The other copy is meant for the Library
- 37) Evaluation

CA Components			50 Marks
7.	First Review	10 Marks	
8.	Second Review	10 Marks	
9.	Implementation and Testing	30 Marks	
Total		50 Marks	
Semester Examinations			50 Marks
1.	Evaluation of Project Work Document	40 Marks	
2.	Viva – Voce	10 Marks	
Total		50 Marks	
Total			100 Marks

38) Project Report Contents

Requirement Analysis	10 Marks
Design	10 Marks
Implementation	10 Marks

Testing and Documentation
Total

10 Marks
40 Marks

Internal examiner and the external examiner will evaluate the project report separately and average will be calculated as the final semester mark of the student.

39) Project Plan

Project area	Work product
Project Management	Project Proposal
	Project Plan
	Project Review Record -1
Requirement analysis	System Study (SSD)
	Vision Document (VSD)
	Use Case Specification
	Project Review Record – 2
Design	Design Document
	Architectural Design
	Database Design
	Interface Design
	Procedural Design
	Test Case Design
First Review	Project Review Record -3
	Draft Report (Combination of all the above work products)
	PPT for project presentation
Implementation	Project Presentation
	Overview of the Project
	Pseudo Code (Algorithms)
Test	Project Review Record – 4
	Test Case Document
	Unit Testing
	Integration Testing
Second Review	System Testing
	Project Review Record – 5
	Draft Copy of the Project Report
	PPT for Project Presentation
Application Demo	Project Presentation
	Application Demo

XVII. DETAILED SYLLABI
[SEMESTER V]

[4:0:4-50:50]

PROGRAMMING USING JAVA

1. Learning Objectives

- To acquire the programming skills in core java applications.
- To learn the art of GUI programming with Applet.
- To write interface with Applet Controls.
- To understand the Layouts of Applets.
- To establish database connectivity.
- To learn the Interaction between AWT control and Data Base.

2. Course Outline

Unit – I: Foundation, Essentials, Control Statement and Classes & Objects

Stage of Java – origin of Java – challenges - features - Object-Oriented Programming; **Java Essentials:** Elements - API - variables - primitive data types – String Class - operators –combined assignment operators - conversion –scope – comments - keyboard input; **Control Statements:***if, if-else*, nested *if&if-else-if* statements – logical operators – comparison – conditional operator – *switch* – increment and decrement – *while, do-while&for* loops – nested loops – *break* and *continue*; **Classes and Objects:** classes and objects -modifiers - passing arguments– constructors - package & import - static class members –method overloading– constructor overloading – returning objects – *this* variable – recursion – nested & inner classes – abstract classes & methods.

Unit – II: Arrays, String Handling, Inheritance, Interface and Packages

Introduction –processing array – passing arrays – returning arrays – String arrays – two Dimensional Arrays - Arrays with Three or More Dimensions; **String Handling :** String class – concatenation – comparison – substring – methods – other methods– *StringBuffer, StringBuilder&StringTokenizer* classes;**Inheritance:** basics –inheriting and overriding superclass methods – calling superclass constructor – polymorphism – inherit from different classes – abstract classes – final Class; **Interfaces:**Basics – multiple Interfaces – multiple inheritance using interface – multilevel interface – **Packages** – Create and access packages in NetBeans IDE – static Import and package class – access specifiers.

Unit – III: Exception Handling, I/O and File Handling and Multithreading

Introduction - *try* and *catch* block - multiple *catch* block - nested *try* - finally Block – *throw* Statement – exception propagation – *throw* Clause - custom exception – built-in exception; **Multithreading:**Introduction – threads – thread creation – life cycle – joining a thread – scheduler & priority – synchronization – inter-thread communication – thread control – thread Pool – thread group – daemon thread; **Files and I/O Streams:** *file* Class – streams – byte streams – filtered byte streams – *RandomAccessFile* class – character streams.

Unit – IV: Applet and GUI Part I

Fundamentals – applet class – life cycle – steps for applet program – passing values through parameters – graphics – event handling; **GUI I:**GUI – creating windows – dialog boxes – layout managers – AWT component classes – Swing component classes – applications of AWT controls.

Unit – V: GUI Part II and Java Database Connectivity

Event handling – AWT components – AWT graphics classes – Swing controls – application using Swing and AWT; **Java Database Connectivity:** types of drivers – JDBC architecture – JDBC classes & interfaces – steps in JDBC applications – creating a new Database and table with JDBC.

3. Teaching Resources

i. Text

1. S.Sagayaraj, R.Denis, P.Karthik&D.Gajalakshmi, “Constructive Java Programming“, Universities Press, 2021

Unit - I	: Ch. 1.1 – 1.5, 2.1 – 2.11, 3.1 – 3.15 & 4.1- 4.13
Unit - II	: Ch. 5.1 – 5.8, 6.1 – 6.9, 7.1 - 7.7 & 8.1 – 8.8
Unit - III	: Ch. 9.1 – 9.10, 10.1 - 10.12 & 11.1 – 11.6
Unit - IV	: Ch. 12.1 – 12.7 & 13.1 – 13.7

ii. References

1. Patrick Naughton and Herbert Schildt. The Complete Reference JAVA 2. 3rd Edition. Tata McGraw-Hill Edition, 1999.
2. Muthu C. Programming with JAVA. 2nd Edition. Vijay Nicole Imprints, 2011.
3. Ken Arnold Gosling and Davis Holmen. The Java Programming Language. 3rd Edition. Addition Wesley Publication.

iii. Web References

(i) Online tutorials

1. <http://www.roseindia.net/java/>
2. www.tutorialspoint.com/java

(ii) Online quiz

1. www.bullraider.com/quiz/core-java-quiz
2. www.javatpoint.com/examaccess.

4. Learning Outcomes

Upon completion of this course, students should be able to:

- Understand the concept of OOP as well as the purpose and usage principles of inheritance, polymorphism, encapsulation and method overloading.
- Identify classes, objects, members of a class and the relationships among them needed for a specific problem.
- Create Java application programs using sound OOP practices (e.g., interfaces and APIs) and proper program structuring (e.g., by using access control identifies, and create user define package for specific task,(reusability concepts) error exception handling)
- Develop programs using the Java standard class library.
- Develop software in the Java programming language, (using applet, AWT controls, and JDBC)

WEB DEVELOPMENT USING XML

1. Learning Objectives

- To know how to represent data over the Web using XML.
- Understanding of the XML Document Object Model.
- Understanding xml DTD and its uses.
- Understanding xml schema and its uses.
- Understanding JSON and its uses.

2. Course Outline**Unit - I: Fundamentals of XML**

SGML - The Beginnings of XML – Benefits of XML - Advantages of XML over SGML, HTML, Databases and Flat Files - Drawbacks of XML. XML Syntax - Document Structure - Declaration - Markup and Content - Elements - Attributes - Entities - Comments - Processing Instructions - Rules of XML Structure – Well Formed and Valid Documents - Applying CSS Style to XML.

Unit - II: Validating XML with the DTD

Document Type Definitions -Some Simple DTD Examples - Structure of a Document - Type Definition-DTD Attributes-DTD Entities-DTD Directives-DTD Drawbacks and Alternatives

Unit – III: XML Schema

Schema Recommendation - Document - Schema for XML Document - Creating XML Schemas - Declaring Attributes - Declaring Elements - Declaring Complex Elements - Declaring Simple Types - Refining Simple Types Using Facets - Anonymous Type Declarations - Specifying Mixed Content for Elements - Annotating Schemas - Model Groups - Attribute Groups - Targeting Namespaces - "Inheriting" from Other Schemas.

Unit – IV: X-path, X-link and Xml for the Web

XPath - Operators and Special Characters - XPath Syntax – Axes – Predicate – XPath Function. XPointer - Points - Ranges - Abbreviating XPointer Notation - XLink - Simple Links - Extended Links. **JSON:** JSON Introduction - JSON Syntax - JSON Data types - JSON Objects - JSON Schemas - JSON Comparison with XML.

Unit - V: XML DOM

What Is DOM, Anyway? - What DOM Is Not-Why Do I Need DOM?-Disadvantages of Using DOM - DOM Levels - DOM Core: Parents, Children, and Siblings - DOM Interfaces - Java Bindings - Walking Through an XML Document -Creating an XML Document -DOM Traversal and Range: Traversal - Range.

3. Teaching Resources

i. Text

1. Ron schmelzer. et al. XML and Web Services Unleashed. Sams Publishing, 2002.

Unit 1 : Ch. 1 & 2

Unit 2 : Ch. 3

Unit 3 : Ch. 4

Unit 4 : Ch. 5

Unit 5 : Ch. 7

ii. Reference

1. David Chappell and Tyler Jewell. Java Web Services. 1st Edition. O'Reilly, 2002.

iii. Web References

(i) Online Tutorial

1. <http://www.w3schools.com/xml/>
2. <http://www.scribd.com/doc/29110068/XML-and-Web-Services>
3. <http://msdn.microsoft.com/en-us/library/ms996507.aspx>

(ii) Online Quiz

1. <http://www.indiabix.com/online-test/>
2. <http://www.pskills.org/xml.jsp>

(iii) Online Compiler

1. <http://compileonline.com/>

4. Learning Outcomes

Upon completion of this course, students should be able to:

- Describe how namespaces are used in XML.
- Follow XML syntax rules.
- Validate XML using DTD.
- Construct XSLT style sheets for transforming HTML.
- Construct XPath expressions for use within XSLT style sheet templates.
- Be able to write the schema for the given XML documents in both DTD and XML Schema languages.
- Be able to parse XML documents by using DOM.

PROGRAMMING USING PHP

1. Learning Objectives

- To learn about PHP is a server scripting language, and a powerful tool for making dynamic and interactive Web pages
- To Understand File handling concepts
- Understanding PHP code to connect, access, and update a MySQL database
- Understanding PHP using XML

2. Course Outline**Unit – I: Fundamentals of PHP**

Web server – Apache - PHP Intro- PHP Install -PHP Syntax -PHP Variables-PHP Echo / Print -PHP Data Types- PHP Strings -PHP ConstantsPHP Operators- Control structures - PHP Functions - Directory Functions - File System Functions -PHP ArraysPHP Sorting ArraysPHP Super global - String Functions - Date and Time Functions-Mathematical Functions - Miscellaneous Functions.

Unit – II: PHP Forms

Basic Form Processing (GET and POST Method) - PHP Form Handling - PHP Form Validation- PHP Form Required– URL - E-mail- PHP Form Complete.

Unit - III: PHP Advanced

PHP Arrays Multi-PHP Date and Time- PHP Include-PHP File Handling-PHP File Open/Read- PHP File Create/Write- PHP File Upload-PHP Cookies- PHP Sessions- PHP Filters- PHP Filters Advanced- PHP Error Handling- PHP Exception-COM-DOM - CURL-SOAP.

Unit – IV: PHP with MySQL Database

PHP MySQL Functions -Connect- Create DB -Create Table- Insert Data- Get Last ID- Insert Multiple- Prepared-Select Data- Delete Data- Update Data- Limit Data - Table join - Database driven application.

Unit - V: PHP - XML

PHP XML Parsers - PHP Simple XML Parser- PHP Simple XML - GetPHP XML ExpatPHP XML DOM.

3. Teaching Resources**i. Text**

1. Julie C.Meloni, Sams Teach yourself PHP, MySQL and Apache, Fourth edition, 2008 by sams publishing.

Unit - I : Ch. 3 – 8, 10

Unit - II : Ch. 11

Unit - III : Ch. 12-13

Unit - IV : Ch. 16

Unit - V : Ch. 28

ii. References

1. Nowicki, et al. Professional PHP, Wrox Press, 2000.

iii. Web References

(i) Online Tutorial

1. www.w3schools.com
2. www.php.net
3. www.phpclasses.org

4. Learning Outcomes

Upon completion of this course, students should be able to:

- Understand process of executing a PHP-based Script on a webserver.
- Understand basic PHP syntax for variables use and standard language constructs, such as conditional and loops.
- Storing data in arrays.
- Using PHP built-in functions and creating custom functions
- Understanding POST and GET in form submission.
- How to receive and process form submission data.
- Reading and writing cookies.
- Create a database in phpMyAdmin Read and process data in a MySQL database.

OPERATING SYSTEMS

1. Learning Objective

- To acquire the principles of Operating System, Process, its Description, Uniprocessor and Multiprocessor and its Scheduling Techniques.
- To understand the concept of Mutual Exclusion, Deadlock and its detection, prevention & avoidance.
- To learn the various Main Memory and Virtual Memory Management techniques.
- To explore the Organization and Management of I/O, Disk and File Managements.

2. Course Outline**Unit – I: Operating System Overview and Process Description**

Operating System Objectives and Functions - The Evolution of Operating Systems - Developments Leading to Modern Operating Systems - Process Description and Control: What is a Process? - Process States - Process Description - Process Control - Security Issues.

Unit – II: Uniprocessor, Multiprocessor and Real-Time Scheduling

Types of Processor Scheduling - Scheduling Algorithms - Multiprocessor Scheduling - Real-Time Scheduling.

Unit – III: Mutual Exclusion, Synchronization and Deadlock

Mutual Exclusion: Hardware Support – Semaphores : Message Passing – Readers / Writers Problem - Principles of Deadlock - Deadlock Prevention - Deadlock Avoidance - Deadlock Detection.

Unit – IV: Memory Management and Virtual Memory

Memory Management Requirements - Memory Partitioning – Paging – Segmentation - Security Issues – Virtual Memory: Hardware and Control Structures - Operating System Software.

Unit – V: I/O Management, Disk Scheduling and File Management

I/O Devices - Organization of the I/O Function - I/O Buffering - Disk Scheduling – File Management: Overview - File Organization and Access - File Directories - File Sharing – Record Blocking – Secondary Storage Management - File System Security.

3. Teaching Resources**i. Text**

1. William Stallings,” Operating Systems: Internals and Design Principles”, 7th Edition, Pearson Education Inc., Fourth Impression: 2016.

ii. Reference

1. Madnick S.E and Donovan J.J. Operating Systems. New Delhi: McGraw hill International Book Co, 1987.

4. Learning Outcomes

- To make students able to learn different types of operating systems along with concept of file systems and CPU scheduling algorithms used in operating system.
- To provide students knowledge of memory management schemes and I/O handling algorithms.
- At the end of the course, students will be able to implement various algorithms required for management, scheduling, allocation and communication used in operating system.
- Able to compare & constant various scheduling algorithm.

COMPUTER GRAPHICS

1. Learning Objectives

Upon completion of this course, students should be able to:

- Understand the Role and importance of Algorithms like Line drawing Algorithm, Circle drawing Algorithm, Character generating Algorithm.
- Understand 2D and 3D Transformations.
- Understand various Clipping Algorithms like point clipping, line clipping and polygon clipping.
- Understand the importance of the User Dialogue and various input functions.
- Understand the Visible Surface Detection Methods.

2. Course Outline

Unit - I: Overview of Graphics system

Raster scans display - Random scan display - Graphics software's - Output Primitives: Points and Lines - Line drawing algorithms: DDA Algorithm - Bresenham's Line Algorithm - Circle generating algorithms: Properties of Circles - Mid Point Circle Algorithm.

Unit – II: 2D Transformations

Two-Dimensional Transformation: Translation-Scaling-Rotation - Homogenous Representation - Inverse Transformation - Composite Transformation: Translation – Rotation - Scaling-Pivot point Rotation-fixed point scaling - Other Transformation: Reflection-Shear.

Unit – III: 2D viewing and Clipping

The viewing pipeline - Window to Viewport coordinate Transformation - Clipping operation - Point clipping - line clipping: Cohen Sutherland line clipping - Polygon clipping: Sutherland Hodgeman polygon clipping.

Unit – IV: Graphical User Interfaces and 3D Transformation

The User Dialogue - Input of Graphical Data: Locator, Stroke, String, Valuator, And Choice, Pick Devices - Interactive picture construction techniques - Three Dimensional Display Methods: 3D Transformation - 3D Viewing.

Unit – V: Visible Surface Detection Methods

Visible Surface Detection: Back Face Detection - Depth Buffer Method - A Buffer Method - Scan Line Method - Depth Sorting Method - Area Subdivision Method.

3. Teaching Resources

i. Text

1. Donald Hearn and Pauline Baker M. Computer Graphics C version 2nd Edition, Prentice Hall, 2011(Reprint).

Unit I :Ch 2.1, 2.7, 3.1-3.2, 3, 3.11, 3.14

Unit II :Ch 5.1-5.4

Unit III :Ch 6.1-6.7, 6.8

Unit IV :Ch 8.1-8.2, 9.1, 10.1, 11.1- 11.3

Unit V :Ch 13.1-13.8

ii. References

1. Neumann W.M. and Sproull R.F. Principles of Interactive Computer Graphics 2nd Edition Tata McGraw Hill International: 1979.
2. Edward Angel. Interactive Computer Graphics 5th Edition Pearson Education, 2009.

iii. Web References

(i) Online Tutorial

1. <http://www.svecw.edu.in/Docs%5CITCGLNotes2013.pdf>
2. <http://www.cprogrammingexpert.com/C/Tutorial/graphics.aspx>
3. http://www.opengl.org/archives/resources/code/samples/glut_examples/examples.html
4. [http://www. Openglsamples.sourceforge.net/](http://www.Openglsamples.sourceforge.net/)
5. <http://www.openglsamples.sourceforge.net/triangle.html>

4. Learning Outcomes

- To provide comprehensive introduction about computer graphics system, design and two dimensional transformations.
- To make the students familiar with techniques of clipping, three dimensional graphics and three dimensional transformations.
- Prepares the students for activities involving in design, development and testing of modeling, rendering, shading and animation.

DATA MINING AND WAREHOUSING

Learning objectives

- To understand data mining principles and techniques and Introduce DM as a cutting edge business intelligence
- To expose the students to the concepts of data warehousing architecture and implementation
- To study the overview of developing areas – web mining, text mining and ethical aspects of data mining
- To identify business applications and trends of data mining

UNIT- I: DATA MINING AND PREPROCESSING

Data Mining - Kinds of Data – Kinds of patterns –Used technology – Kinds of Applications – Issues in Data mining .Know Your Data: Data objects and Attributes Types –Basic Statistical Description of Data –Data Visualization – Measuring Data Similarity and Dissimilarity. Data Processing –Data Cleaning – Data Integration – Data Reduction –Data Transformation and data Discretization.

UNIT - II: DATA WAREHOUSING AND OLAP

Data Warehousing–Data Warehouse Architecture- Design and Usage –Data Warehouse Implementation – OLAP operations- ROLAP- MOLAP-Association Rules:AprioriAlgorithm-FP- tree Growth Algorithm.

UNIT - III: CLASSIFICATION TECHNIQUES

Classification: Basic Concepts – Decision Tree Induction – Bayes Classification Methods – Rule-Based classification –Model Evaluation and Selection – Techniques to Improve Classification Accuracy.

UNIT - IV: CLUSTER ANALYSIS

Cluster Analysis – Partitioning Methods – Hierarchical Methods – Density-Based Methods – Grid-Based Methods – Evaluation of Clustering- categorical clustering algorithms- STIRR-ROCK-COCTUS- Case Study: Back Propagation-Support Vector Machines.

UNIT - V: WEB MINING

Web Content Mining- Web Structure Mining- Web Usage Mining- Text Mining- Unstructured Text- Episode rule discovery for texts- hierarchy of categories- Text Clustering- Temporal Data Mining- rules- sequence mining- GSP algorithm- Event Prediction Problem- Genetic Algorithm-Time Series Analysis- Spatial Mining- tasks-clustering-trends.

Teaching Resources**i. Text**

1. JiaweiHan, MichelineKamber and Jian Pei, “Data Mining Concepts and Techniques”, Morgan Kauffmann Publishers, Third Edition, 2012
2. Arun K Pujari, “Data Mining Techniques”, Universities Press (India) private Limited, Fourth Edition, 2017.

ii. References

1. K.P. Soman, ShyamDiwakar and V. Ajay, “Insight into Data mining Theory and Practice”, Easter Economy Edition, Prentice Hall of India, 2006
2. G. K. Gupta, “Introduction to Data Mining with Case Studies”, Easter Economy Edition, Prentice Hall of India, 2006

3. Berson, Alex & Smith, Stephen J, “Data Warehousing, Data Mining, and OLAP”, TMH Pub.Co. Ltd, New Delhi, 2012
4. Pang-Ning Tan, Michael Steinbach and Vipin Kumar, “Introduction to Data Mining”, Pearson Education, 2007

iii. Web references

1. <http://www.scribd.com/doc/5710731/mining-background-literature-review>
2. <http://www.scribd.com/doc/104389040/Why-Mining>
3. <http://www.scribd.com/doc/6283008/Data-Integration-Data-Mining-Clinical-Research>
4. <http://www.selectbs.com/products-general/what-is-business-intelligence>
5. <http://www.scribd.com/doc/30346964/Business-intelligence>

Learning Objectives

- To introduce the decision making system, models and support
- To appraise the general nature and range of decision support and group support systems
- To impart about knowledge based system and advanced intelligent systems

UNIT - I: DECISION-MAKING SYSTEMS, MODELING, AND SUPPORT

Decision-Making: Introduction and Definitions, Systems, Models, Phases of the Decision-Making Process, Decision-Making: The Intelligence Phase, The Design Phase, The Choice Phase, The Implementation Phase, How Decisions Are Supported, Personality Types, Gender, Human Cognition, and Decision Styles, The Decision Makers

UNIT – II: DECISION SUPPORT AND GROUP SUPPORT SYSTEM

DSS Configurations, What Is a DSS?, Characteristics and Capabilities of DSS, Components of DSS, The Data Management Subsystem, The Model Management Subsystem, The User Interface (Dialog) Subsystem, The Knowledge-Based Management Subsystem, The User, DSS Hardware, DSS Classifications. **Group Support System:** Group Decision-Making, Communication, and Collaboration, Communication Support, Collaboration Support: Computer-Supported Cooperative Work, Group Support Systems, Group Support Systems Technologies, Group systems Meeting room and Online, The GSS Meeting Process, Distance Learning, Creativity and Idea Generation.

UNIT - III: KNOWLEDGE-BASED SYSTEMS

Concepts and Definitions of Artificial Intelligence, Evolution of Artificial Intelligence, The Artificial Intelligence Field, Basic Concepts of Expert Systems, Applications of Expert Systems, Structure of Expert Systems, How Expert Systems Work, Problem Areas Suitable for Expert Systems, Benefits and Capabilities of Expert Systems, Problems and Limitations of Expert Systems, Expert System Success Factors, Types of Expert Systems, Expert Systems on the Web.

UNIT – IV: KNOWLEDGE ACQUISITION, REPRESENTATION, AND REASONING

Concepts of Knowledge Engineering, Scope and Types of Knowledge, Methods of Knowledge Acquisition from Experts, Knowledge Acquisition from Multiple Experts, Automated Knowledge Acquisition from Data and Documents, Knowledge Verification and Validation, Representation of Knowledge, Reasoning in Rule-Based Systems, Explanation and Meta knowledge, Inferencing with Uncertainty, Expert Systems Development, Knowledge Acquisition and the Internet.

UNIT – V: ADVANCED INTELLIGENT SYSTEMS

Machine-Learning Techniques, Case-Based Reasoning, Basic Concept of Neural Computing , Learning in Artificial Neural Networks, Developing Neural Network-Based Systems, Genetic Algorithms Fundamentals, Developing Genetic Algorithm Applications, Fuzzy Logic Fundamentals, Developing Integrated Advanced Systems.

Teaching Resources

i. Text

1. Efraim Turban and Jay E. Aronson, Decision Support System and Intelligent Systems, Prentice Hall International, 7th Edition 2007.

ii. References

1. Janakiraman V. S and Sarukesi K, Decision Support Systems, Prentice Hall of India, 6th Printing 2006.

2. Lofti, Decision Support System and Management, McGraw Hill Inc, International Edition, New Delhi 1996.

3. Marakas, Decision Support System, Prentice Hall International, Paperback Edition, New Delhi, 2003

iii. Web references

1. ndwrcdp.werf.org/documents/WU-HT-03-35/DSS%20Tutorial.pdf

2. www.slideshare.net/sursayantan92/decision-support-systemdss
www.uky.edu/BusinessEconomics/dssakba/instmat.htm

3. <https://ceit.aut.ac.ir/~shiry/lecture/DSS/Introduction.ppt>

SOFTWARE TESTING AND QUALITY ASSURANCE**Learning objectives**

- To introduce various approaches, techniques, technologies, and methodologies used in software testing and quality assurance.
- To understand the role of testing in applications
- To learn to design the test cases
- To know the different levels of testing
- To study the state-of-the-art of software testing and quality assurance.

UNIT I: TESTING BASICS

Testing as an engineering activity – Role of Process in software quality – Testing as a process – Basic definitions – Software testing principles – The tester's role in a software development organization – Origins of defects – Defect classes – The defect repository and test design – Defect examples – Developer / tester support for developing a defect repository.

UNIT II: TEST CASE DESIGN

Introduction to testing design strategies – The smarter tester – Test case design strategies – Using black box approach to test case design – Random testing – Equivalence class partitioning – Boundary value analysis – Other black box test design approaches – Black box Testing and COTS – Using white box approach to test design – Test adequacy criteria – Coverage and control flow graphs – Covering code logic – Paths – Their role in white box based test design – Additional white box test design approaches – Evaluating test adequacy criteria.

UNIT III: LEVELS OF TESTING

The need for levels of testing – Unit test – Unit test planning – Designing the unit tests – The class as a testable unit – The test harness – Running the unit tests and recording results – Integration tests – Designing integration tests – Integration test planning – System test – The different types – Regression testing – Alpha, beta and acceptance tests.

UNIT IV: FUNDAMENTALS OF SOFTWARE QUALITY & QUALITY ASSURANCE

Software quality - Hierarchical models of Boehm and McCall - Quality measurement - Metrics measurement and analysis - Gilb's approach - GQM Model-Quality tasks - SQA plan - Characteristics - Implementation - Documentation - Reviews and audits.

UNIT V: QUALITY CONTROL AND RELIABILITY

Defect prevention and removal - Reliability models - Rayleigh model - Reliability growth models for quality assessment. Case Study: Tools for quality - Ishikawa's basic tools - CASE tools.

Teaching Resources**i. Text**

1. Ilene Burnstein, "Practical Software Testing", Springer International Edition, 2003. (Unit I, II and III)
2. Allan C. Gillies, "Software Quality: Theory and Management", Thomson Learning, 2003. (Unit IV)
3. Stephen H. Kan, "Metrics and Models in Software Quality Engineering", Pearson Education (Singapore) Pvt. Ltd., 2002. (Unit 5)

ii. References

1. Elfriede Dustin, "Effective Software Testing", Pearson Education, 2003.
2. RenuRajani and Pradeep Oak, "Software Testing – Effective Methods, Tools and Techniques", Tata McGraw Hill, 2003.
3. Mordechai Ben, Menachem and Garry S. Marliss, "Software Quality", Thomson

Asia Pvt. Ltd., 2003.

4. Kamna Malik and Praveen Choudry, "Software Quality: A Practitioner Approach", PHI, 2000.

iii. Web references

http://en.wikibooks.org/wiki/Software_Quality_Assurance

PRACTICAL - V: PROGRAMMING USING JAVA

1. Classes and objects
2. Inheritances & Interfaces
3. Package
4. String Handling
5. Exception Handling
6. File handling
7. Multithreading
8. Menu and Dialogue Box
9. Applet and AWT Controls
10. GUI application with JDBC

PRACTICAL - VI: WEB DEVELOPMENT USING XML

1. XML Document Structure
2. Rules of XML Structure
3. XML with XSLT/CSS
4. Namespaces in XML
5. Creating XML Schemas
6. XPath
7. XPointer
8. XLink
9. XHTML
10. XFORMS

PRACTICAL - VII: PROGRAMMING USING PHP

1. Data Types and Operators
2. Control Statements and Looping
3. Functions
4. Arrays
5. Form Processing (GET & POST)
6. Validation
7. File Uploading and Downloading
8. Cookies
9. Forms and Databases
10. XMLs

MOBILE APPLICATIONS DEVELOPMENT

1. Learning Objectives

Upon completion of this course, students should be able to:

- To develop a mobile application.
- To understand the concept of SQLite.

2. Course Outline**Unit - I: Introducing Android**

Introduction – History – Versions – Features – Understanding the Android market - Android software stack – Life cycle of an Android – The layers of Android – The Intent of Android development – Four kinds of Android components – Understanding the AndroidManifest.xml file – Mapping applications to processes – Android development environment – Introducing the Android SDK – Exploring the development environment – Building an Android application in Eclipse - Creating an Android Hello World Application – Using the Android emulator – Debugging your application.

Unit - II: Building Basic User Interfaces and Using Controls

User Interfaces – Understanding Android’s Common Controls – Adapters and List Controls – Understanding Layout Managers – Working with Menus and Action Bars - Working with views – Intents and Services – Toast.

Unit - III: Android Applications

Telephony – Exploring telephony background and terms – Accessing telephony information – Interacting with the phone – Working with messaging: SMS – Notifications and alarms – Introducing Toast – Placing your Toast message – Making custom Toast view – Introducing notifications – Making a custom notification view – Introducing alarms – Graphics and animation – Drawing graphics in Android – Creating animations with Android’s Graphics API – Multimedia – Introducing to Multimedia and Stage fright – Playing audio – Playing video – Capturing media.

Unit - IV: The Maturing Platform

Location – Simulating your location within the emulator – Using Location Manager and Location Provider – Working with Maps – Converting places and addresses with Geocoder – Bluetooth and sensors – Exploring Android’s Bluetooth capabilities – Interacting with the Sensor Manager – App Widgets – Drag and Drop – The drag-and-drop classes – Drag-and-drop operations – The shadow builder – Drag events – Starting drag operations – Listening for drag-and-drop events – Responding to drag-start operations – Handling drop operations.

Unit - V: Database Operations

Storing and retrieving data – Creating a SQLite Database – Migrating a Database – SQLite DB: CRUD Operations. Publishing Android Application: Export android application – Google play store registration.

3. Teaching Resources**i. Text Book**

1. W. Frank Ableson, Robi Sen, Chris King, C. Enrique Ortiz, “**Android in Action**”, Third Edition 2012.
2. Dave Maclean, Satya Komatineni, Grant Allen, “**Pro Android 5**”, Apress Edition 2015.

ii. Reference

1. Dave Smith and Jeff Friesen, “Android Recipes: A Problem – Solution Approach”, Rakmo Press (P) Ltd, New Delhi, 2011.

iii. Web Reference

1. Android Developer’s Guides - available at <http://developer.android.com/>

4. Learning Outcome

Upon completion of this course, students should be able to:

1. Describe the platforms upon which the Android operating System will run.
2. Create a simple application that runs under the Android operating system.
3. Access and work with the Android file system.
4. Create an application that uses multimedia under the Android operating system.
5. Access and work with database under the Android operating system.

1. Course objectives:

- Develop basic understanding of the basics of Python programming language.
- Learn core Python scripting elements such as data types and flow control structures.
- Design simple applications using Python.

2. Course outline**UNIT - I: OVERVIEW**

Introduction to Python: Features of Python - How to Run Python – Identifiers - Reserved Keywords - Variables - Comments in Python - Indentation in Python - Multi-Line Statements - Multiple Statement Group(suite) – Quotes in Python - Input, Output and Import Functions - Operators.

UNIT - II: DATA TYPES AND OPERATIONS

Data Types and Operations: Numbers- Strings - List- Tuples – Set- Dictionaries -Data type conversion.

UNIT - III: FLOW CONTROL

Flow Control: Decision Making- Selection Structures-Loops-Nested Loops-Types of Loops.

UNIT - IV: FUNCTIONS, MODULES AND PACKAGES

Functions: Function Definition-Function Calling - Function Arguments - Recursive Functions - Function with more than one return value- Modules and Packages: Built-in Modules - Creating Modules - import Statement - Locating Modules - Namespaces and Scope - The dir() function - The reload() function - Packages in Python - Date and Time Modules.

UNIT - V: FILE HANDLING AND EXCEPTION HANDLING

File Handling: Opening a File - Closing a File - Writing to a File – Reading from a File - File Methods - Renaming a File - Deleting a File - Directories in Python- Exception Handling: Built-in Exceptions - Handling Exceptions - Exception with Arguments- Raising Exception - User-defined Exception - Assertions in Python

3. Teaching Resources**i. Text**

1. Jeeva Jose and P. Sojan Lal, “Introduction to Computing and Problem Solving with Python”, Khanna Book Publishing Co. (P) Ltd., 2016.

ii. References

1. ChSatyanarayana, M Radhika Mani & B N Jagadesh, “Python Programming”, Universities Press, 2018.

iii. Web references

1. www.learnpython.org/
2. <https://www.codecademy.com/learn/python>
3. <https://www.Codementor.io>
4. <https://www.Python.org>

4. Course outcomes

After this course, the student will be able to

- Understand and apply Python's core data types while writing new programs.
- Express different decision making statements and functions
- Understand and summarize the different file handling operations

LINUX AND SHELL PROGRAMMING

1. Learning Objectives

- State the major components and describe the architecture of the UNIX operating system.
- To learn and understand UNIX commands.
- State how the shell functions at the user interface and command line interpreter.
- Create structured shell programming with flow control constructs.

2. Course Outline**Unit - I: Organization**

Salient Features of Unix – Unix System Organization – Types of Shells – Unix Commands – The Unix File System – Creating Files – Listing Files and Directories. - The Boot Block – The Super Block – The Inode Table – Data Blocks – How Does Unix Access Files – Storage of Files – Disk Related Commands.

Unit - II: UNIX Commands

Essential Unix commands: Password – Commands: cal, banner, touch – File Related Commands – Viewing Files – Taking Printouts – File Compression. I/O Redirection and Piping. vi editor – Modes of operation – The First Editing Session. Processes in Unix: What's Running Right Now – Still More Processes – Background Processes – The nohup command – Killing a process – Changing Process Priorities – Scheduling of Processes, Communication – Unix write and wall command - Basis of Unix Communication.

Unit - III: SHELL Programming - I

Interactive Shell Scripts – Shell Variables – Shell Keywords –Assigning Values to Variables – Positional Parameters – Passing Command Line Arguments – Setting Values of Positional Parameters – Displaying Date in Desired Format – Using Shift on Positional Parameters – Arithmetic in Shell Script, Taking Decisions: if-then-fi Statement – if-then-else-fi Statement – The test Command – Nested if-else – Form of if – Use of Logical Operators – else - if Equals elif – The Case Control Structure.

Unit - IV: SHELL Programming - II

Loop Control Structure: Loops – The While Loop – Reading from a file – The Until and for Loop – Creating Nested Directories – Generating Values for a for Loop – The Break and Continue Statement- Shell script using Command Line Arguments.

Unit - V: System Calls

System calls: Operational mode – Kernel mode – User mode. File Handling calls: open(), create(), open(), read(), write(), lseek(),close(). Directory Handling calls: mkdir(), rmdir(), chdir(), getcwd(), opendir(), readdir(), telldir(), seekdir(), rewaddir(), closedir(). Process related calls - exec(), fork(), wait(), exit(). Interrupted system calls – Error Handling: strerror() – perror().

3. Teaching Resources**i. Text**

1. YashavantKanetkar. Unix Shell Programming. New Delhi: BPB Publisher, 1996.

Unit – I : Ch. 1, 2, 3.

Unit – II : Ch. 4, 5, 6, 7, 8.

Unit – III : Ch. 9 - 10

- Unit – IV** : Ch. 11.
2. BM. Harwani. Unix and Shell programming. OXFORD University press.
Unit – V : Ch. 7.1, 7.2.1- 7.2.6, 7.3, 7.4, 7.5, 7.8.

ii. Reference

1. Kernighan. et al. The UNIX Programming Environment. 2nd Edition. New Delhi: Prentice Hall of the India, 1988.

iii. Web References

(i) Online tutorials

1. <http://www.cgl.ucsf.edu/Outreach/bmi219/slides/shell.html>
2. <http://www.cs.utk.edu/~huangj/cs360/360/notes/Syscall-Intro/lecture.html>

(ii) Online quiz

1. www.tcyonline.com/tests/unix-and-shell-scripts

(iii) Online compiler

1. www.compileonline.com/execute_bash_online.php/

4. Learning Outcome

Upon completion of this course, students should be able to:

- Understand the basic Unix command
- Understand the concepts piping and redirections.
- Create a shell script using VI editor.
- Able to develop using shell script to solve simple application problem.

MICROPROCESSOR USING 8086/88

1. Learning Objectives

- To Understand the basic architecture of the Microprocessor
- To learn the instruction sets of the processor
- To write applications using assembly level language program
- To study the input/output interfaces of the processor
- To understand the importance of interrupts in programming

2. Blue Print of the Question Paper

Section	I-Unit	II-Unit	III-Unit	IV-Unit	V-Unit
Section-A	1-2	3-4	5-6	7-8	9-10
Section-B	11.a)Theory (OR) b) Theory	12.a)Theory (OR) b) Program	13.a)Theory (OR) b) Program	14.a)Theory (OR) b) Theory	15.a) Theory (OR) b) Theory
Section-C	16.Theory	17. Program	18. Program	19.Theory	20. Theory

3. Course Outline**Unit - I: Software Architecture and Machine Coding**

Microcomputer : PC – Architecture – Microprocessor Evolution – micro architecture of the 8088/8086 - Software Model - Memory Address Space And Data Organization - Data Type - Segment Registers And Memory Segmentation - Dedicated And General Use Of Memory - Instruction Pointer - Data Registers - Pointer And Index Registers - Status Register - Generating A Memory Address - The Stack - I/O Address Space - Addressing Modes.

Unit - II: Microprocessor Programming - I

The Instruction Set Of 8086 - Data Transfer Instructions - Arithmetic Instructions - Logic Instructions - Shift Instructions - Rotate Instructions.

Unit - III: Microprocessor Programming - II

Flag Control Instructions - Compare Instructions - Control Flow and the Jump Instructions - Subroutines and Subroutine - Handling Instructions - The Loop and The Loop Handling

Instructions - Strings And String - Handling Instructions.

Unit - IV: I/O Interface of the 8086 Microprocessor

8088 and 8086 Microprocessors – Minimum mode and Maximum mode systems – Minimum mode Interface Signals – Maximum mode Interface Signals –Memory Control Signals – Read and Write cycles.

Unit - V: Interrupt Interface of the 8086

Interrupt Mechanism, Types, and Priority – Interrupt Vector Table - Interrupts Instructions - Enabling/Disabling Of Interrupts - External Hardware Interrupt Interface -External Hardware Interrupt Sequence - 8259A Programmable Interrupt Controller -Software Interrupt - Non-Maskable Interrupt – Reset - Internal Interrupt Functions - DMA.

4. Teaching Resources

i. Text

1. Triebel. et al. The 8088 And 8086 Microprocessors Programming, Interfacing Software, Hardware And Applications. 4th Edition. New Delhi: Prentice Hall Of The India, 2011.

Unit - I: Ch. 1.1 – 1.2, 2.1-2.16

Unit - II: Ch. 4.1-4.7

Unit - III: Ch. 5.1-5.7

Unit - IV: Ch. 6.1-6.5, 6.11-6.12

Unit - V: Ch. 8.1-8.8&8.10-8.13

ii. Reference

1. John Uffenbeck, The 8086/8088 Family, Design, Programming And Interfacing. 7th Edition. New Delhi: Prentice Hall of India, 2000.

iii. Web Reference

(i) Online Tutorial

1. <https://www.udemy.com/course/8086-microprocessor-architecture-in-one-video-in-easy-way/>
2. <https://www.geeksforgeeks.org/microprocessor-tutorials/>
3. <https://www.wisdomjobs.com/e-university/microprocessor-tutorial-2391.html>

(ii) Online Quiz

1. <https://gatetestseries.in/ec-ese-computer-engineering/8086-microprocessors/>

2. <https://www.goconqr.com/quiz/10973890/microprocessor-and-assembly-language-8085-8086>
3. <http://examradar.com/microprocessor-8086-mcqs-set-1/>

5. Learning Outcomes

At the end of the course, students should be able to:

- Identify the types of instructions and the organization of registers and memory
- Describe the translation model of assembly language to machine language.
- Understand the micro-program by mapping the instructions.
- Recognize the types of computer organizations.
- Accept the better ways of Parallel and Vector processing.

COMPUTER NETWORKS

1. Learning Objective

- To learn the basic concepts of Computer Networks.

2. Course Outline

Unit – I: Data Communications

Introduction: Data Communications – Networks – Network Models: Layers in the OSI Model – Addressing. Transmission Media: Guided Media – Unguided Media.

Unit – II: Data Link Layer

Error Detection and Correction: Introduction – Block Coding – Linear Block Codes – Cyclic Codes: Cyclic Redundancy check – Checksum. Data Link Control: Framing – Flow and Error Control – Protocols – Noiseless Channels – Noisy Channels.

Unit – III: Network Layer

Internet Protocol: Internetworking – IPv4 – IPv6 – Transition from IPv4 to IPv6 - Delivery, forwarding and Routing: Delivery- Forwarding.

Unit – IV: Transport Layer

Process-to-Process Delivery: User Datagram Protocol – TCP. Quality of service: Data Traffic – Congestion – Congestion Control – Quality of Service.

Unit – V: Application Layer

Domain Name System: Name Space – Domain Name Space – Distribution of Name Space – DNS in the Internet – Resolution – DNS Messages – Types of Records – Registrars – Dynamic Domain Name System – Encapsulation. Remote Logging – Electronic Mail – File Transfer.

3. Teaching Resources

i. Text

1. Behrouz A Forouzan, “Data Communication and Networking”, 4th Edition, Tata McGraw-Hill Publishing Company Limited, New Delhi: 2008.

ii. Reference

1. Andrew S Tanenbaum, ” Computer Networks”, 4th Edition, Pearson Education, New Delhi: 2003.

4. Learning Outcomes

- To explain how communication works in computer networks and to understand the basic terminology of computer networks
- To explain the role of protocols in networking and to analyze the services and features of the various layers in the protocol stack.
To understand design issues in Network Security and to understand security threats, security services and mechanisms to counter it.

SOFTWARE PROJECT MANAGEMENT**LEARNING OBJECTIVES**

- To provide sound knowledge in Project Management.
- To understand the importance of requirement gathering
- To explore different models in Software Development
- To know the workflow of a Project
- To identify various actors in the activity

UNIT I: INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT

Introduction: Project – Software Projects vs other types of Project – Activities Covered by SPM – Some Ways of Categorizing Software Projects – Stakeholders, Setting Objectives – The Business Case - Project Success and Failure - Management and Management Control. Project Evaluation: A Business Case – Project Portfolio Management – Evaluation of Individual Projects – Cost Benefit Evaluation – Risk Evaluation.

UNIT II: PROJECT PLANNING AND SELECTION OF PROJECT APPROACH

Project Planning - Introduction to Step Wise Project Planning – Step 0 to Step 10. Selection of an Appropriate Project Approach -Introduction – Build or Buy – Choosing Methodologies and Technologies – Software Processes and Process Models – Choice of Process Models – The Waterfall Model– Prototyping – other ways of categorizing prototype- Agile Methods – Extreme Programming - Selecting the Most Appropriate Process Model.

UNIT III: EFFORT ESTIMATION AND ACTIVITY PLANNING

Effort Estimation – Introduction –Estimates – Problems with Over and Under-estimate – Basis for Software Estimating – Effort Estimation Techniques – Bottom-up Estimating – Top-down Approach and Parametric Models – Expert Judgment - Estimating by Analogy – Albrecht Function Point Analysis – Function Mark II – COCOMO & COCOMO II – Cost Estimation – Staffing Pattern. Activity Planning –Introduction – Objectives of Activity Planning – When to plan – Project Schedules – Project and Activities – Sequencing and Scheduling Activities – Networking Planning Models – Formulating a Network Model– Activity on Arrow Networks.

UNIT IV: RISK MANAGEMENT, RESOURCE ALLOCATION AND MONITORING

Risk Management –Risk – Categories of Risk – A Framework for Dealing with Risk – Risk Identification – Risk Assessment – Risk Planning – Risk Management. Resource Allocation – Introduction – The Nature of Resources – Identifying Resource Requirements – Scheduling Resources. Monitoring –Creating the Framework – Collecting the Data – Review and Project Termination Review – Visualizing Progress – Cost Monitoring and Earned Value Analysis – Getting the Project Back to Target – Change Control – SCM.

UNIT V: MANAGING PEOPLE AND WORKING IN TEAMS

Managing People –Understanding Behavior – Organizational Behavior – Selecting the Right Person for the Job – Instruction in the Best Methods – Motivation – The Oldham-Hackman Job Characteristics Model – Stress – Health and Safety. Working in Teams –Introduction – Becoming a Team – Decision Making – Organization and Team Structures – Coordination Dependencies – Dispersed and Virtual Teams – Communication Genres – Communication Plans – Leadership.

Teaching Resources**i. Text**

1. BOB Huges, Mike Cotterell, Rajib Mall “Software Project Management”, McGraw Hill, Fifth Edition, 2011.

ii. References

1. Futrell, “Quality software Project management”, Pearson Education India.
2. Royce, “Software Project Management”, Pearson Education India.

iii. Web references

<https://www.lynda.com/Project-Management-training-tutorials/39-0.html>
www.rspa.com/spi/project-mgmt.html

SECURITY SYSTEMS**LEARNING OBJECTIVES**

- To introduce the classical encryption techniques
- To understand the concept of network security practices
- To know about system security and web security
- To impart knowledge on wireless network security
- To explore the types of cyber crimes

UNIT – I: CONVENTIONAL ENCRYPTION & CRYPTOGRAPHY

Classical Technique – Modern technique – Algorithms; Public Key Cryptography : Public Key Cryptography – Introduction to Number Theory – Message Authentication and Hash Function – HASH and MAC Algorithm – Digital Signature and Authentication protocol.

UNIT – II: NETWORK SECURITY PRACTICE

Authentication Application – Electronic Mail Security – IP Security Program Security and System Security: Secure programs – Nonmalicious program errors – viruses and Worms – Memory and address protection – control access to general objects – File protection mechanism – user authentication – Trusted operating system design and assurance – sandboxing -Intrusion Detection system.

UNIT – III: SYSTEM SECURITY AND WEB SECURITY

Intruders,– Firewall - Managing Access – Password management - Web Security requirements – SSL and TLS – SET; Client Side Security : Using SSL – Active Content – Web Privacy. Database Security: The Database as a Networked Server – Securing database-to-database communication – Reliability and Integrity of database – sensitive data – inference – multilevel databases

UNIT – IV: WIRELESS NETWORK SECURITY

Mobile Security – Encryption Schemes in WLANs – Basic approach to WLAN security and Policy Development – WLAN intrusion process – WLAN security solutions. Digital Watermarking and Steganography: Models of Watermarking – Basic Message Coding – Watermark Security – Content Authentication – Steganography.

UNIT – V: CYBER CRIMES

Introduction – computer crime and cyber crimes; Classification of cyber crimes, Cyber crime and Related Concepts: Distinction between cyber crime and conventional crimes, Reasons for commission of cyber crime, Cyber forensic : Cyber criminals and their objectives, Kinds of cyber crimes – cyber stalking; cyber pornography; forgery and fraud; crime related to IPRs; Cyber terrorism; computer vandalism, Regulation of cyber crimes: Issues relating to investigation, Issues relating to Jurisdiction, Issues relating to Evidence , Relevant provisions under Information Technology Act, 2000, Indian Penal Code, Pornography Act and Evidence Act etc.

Teaching Resources**i. Text**

1. Charles P. Pfleeger, Shari Lawrence Pfleegner, “Security in Computing”, Prentice Hall of India, 2011.
2. William Stallings, “Cryptography and Network Security”, 5th Edition, Pearson.

ii. References

1. John W. Rittinghouse, James F. Ransome, “Wireless Operational Security”, Elsevier, 2004.
2. Ron Ben Natan, “Implementing Database Security and Auditing”, Elsevier, 2005.
3. Lincoln D. Stein, “Web Security”, Addison Wesley, 1999.

4. Ingemar J. Cox, Matthew L. Miller, Jeffrey A. Bloom, Jessica Fridrich, Ton Kalker, "Digital Watermarking and Steganography", 2nd Edition, Elsevier.
5. Dr. R. K. Tiwari, P. K. Sastri, K. V. Ravikumar, "Computer Crime and Computer Forensics", 1st Edition, Selective Publishers, 2002.

iii. Web references

www.cryptographyworld.com/

www.ti89.com/cryptotut/home.htm

nptel.ac.in/courses/106105031/

williamstallings.com/Extras/Security-Notes/

<http://study.com/academy/lesson/what-is-cyber-crime-definition-types-examples.html>

COGNITIVE COMPUTING**1. Learning Objectives**

- To learn the basics of Cognitive Science with focus on acquisition, representation, and use of knowledge by individual minds, brains, and machines
- To study the mind and intelligence, embracing psychology, artificial intelligence, Neuroscience and linguistics
- To understand the role of neuro science in the cognitive field

2. Course Outline**UNIT – I: BASICS OF COGNITIVE SCIENCE**

The Cognitive view – Some Fundamental Concepts – Computers in Cognitive Science – Applied Cognitive Science – The Interdisciplinary Nature of Cognitive Science – Artificial Intelligence: Knowledge representation – The Nature of Artificial Intelligence – Knowledge Representation – Artificial Intelligence: Search, Control, and Learning.

UNIT - II: COGNITIVE PSYCHOLOGY

Cognitive Psychology – The Architecture of the Mind – The Nature of Cognitive Psychology – A Global View of The Cognitive Architecture – Propositional Representation – Schematic Representation – Cognitive Processes, Working Memory, and Attention – The Acquisition of Skill – The Connectionist Approach to Cognitive Architecture.

UNIT - III: COGNITIVE NEUROSCIENCE

Brain and Cognition Introduction to the Study of the Nervous System – Neural Representation – Neuropsychology – Computational Neuroscience – The Organization of the mind – Organization of Cognitive systems – Strategies for Brain mapping – A Case study: Exploring mindreading.

UNIT - IV: LANGUAGE ACQUISITION, SEMANTICS AND PROCESSING MODELS

Milestones in Acquisition – Theoretical Perspectives – Semantics and Cognitive Science – Meaning and Entailment – Reference – Sense – Cognitive and Computational Models of Semantic Processing – Information Processing Models of the Mind – Physical symbol systems and language of thought – Applying the Symbolic Paradigm – Neural networks and distributed information processing – Neural network models of Cognitive Processes .

UNIT - V: HIGHER-LEVEL COGNITION

Reasoning – Decision Making – Computer Science and AI: Foundations & Robotics – New Horizons – Dynamical systems and situated cognition – Challenges – Emotions and Consciousness – Physical and Social Environments – Applications.

3. Teaching Resources**i. Text**

1. José Luis Bermúdez, Cognitive Science: An Introduction to the Science of the Mind, Cambridge University Press, New York, 2014
2. Carolyn Panzer Sobel and Paul Li, Cognitive Science: An Interdisciplinary Approach, 2013
3. J. Friedenber and G. Silverman, Cognitive Science: An Introduction to the Study of Mind, 2011

PRACTICAL-VIII:MOBILE APPLICATIONS DEVELOPMENT

- 1.Hello World Application
- 2.Android's Common Controls
- 3.Adapters and List Controls
- 4.Menus and Action Bars
- 5.Telephony and SMS
6. Notification and Multimedia
7. Location and Map
8. Drag and Drop
9. CRUD Operations
10. Publishing Android Apps

PRACTICAL-IX: PROGRAMMING USING PYTHON

1. Types of Operators
2. Numbers
3. Strings
4. List & Dictionaries
5. Tuples & Set
6. Flow Control
7. Functions
8. Modules and Packages
9. File Handling
10. Exception Handling

**PRACTICAL - X: UNIX AND SHELL PROGRAMMING & MICROPROCESSOR
USING 8086/88**

Part – I: Programming With Shell Script and System Calls

1. Shell Script - Sequential structure
2. Shell Script – Iterative Control Structure
3. Shell Script – Strings
4. Shell Script – Files
5. Shell script - Command line arguments
6. System Call - Printing the command line arguments.
7. System Call - read(), write(), open(), creat()
8. System Call - execlp(), execvp() perror() system calls
9. System Call - Use of fork(), wait() & exit()
10. System Call - Child process , generated interrupt & lseek()

Part - II: MICROPROCESSOR USING 8086/88

1. 8 and 16 Bit Arithmetic Operation and Logical Operation.
2. Finding the Largest Element in an Array.
3. Sum of the numbers in an array.
4. Computation of Factorial.
5. Sorting – Two methods.
6. Searching - Two methods.
7. Code Conversion from BCD to HEX, ASCII to BCD.
8. String Manipulation.
9. Illustration of loop instruction.
10. Demonstration of procedures.