

BSc (Computer Science) CBCS (w.e.f. 2017-18)

PSO1: Understand concepts of computer organisation, operating systems and computer networks.

PSO2: Design data structures and algorithms for real life problems.

PSO3: Develop data storage and retrieval techniques, build computer based application.

List of Courses for B.Sc. Computer Science Honors/General Degree program

Table I : Bachelor of Science (Honours) Core Papers

Semester	Course Codes	Title of paper	Credits (T+P)
I	CSC101	Programming Fundamentals using C	4T+2P
II	CSC102	Data Structures	4T+2P
III	CSC103	Database Management System	4T+2P
IV	CSC104	Computer Organization and Operating Systems	4T+2P
V	CSC105	Computer Networks	4T+2P
	CSC106	Object Oriented Programming	4T+2P
	CSC107	Software Engineering	4T+2P
VI	CSC108	Mobile Application Development	4T+2P
	CSC109	Full Stack Web Development	4T+2P
	CSC110	Internet of Things	4T+2P
	CSP101	Project	4

Table II : The General Electives for BA, BSc.

Semester	Course Code	Title of paper	Credits(T+P)
I	CSG101	Computer Fundamentals and Emerging Technologies	3T+1P
	CSG102	Cyber Space & Cyber Security	3T+1P
II	CSG103	IT Fundamentals	3T+1P
	CSG104	Multimedia and Web Design	3T+1P

Table III : Skill Based Papers

Semester	Course Code	Title of paper	Credits (T+P)
III	CSS103	Programming in Python	3T+1P
IV	CSS104	Web Application Development using Flask	3T+1P

Table IV : Discipline Specific Elective

Semester	Course Code	Course Title	Credits
V	CSD101	Human Computer Interaction	3T+1P
	CSD104	Embedded Systems	3T+1P
VI	CSD107	Introduction to Data Analytics	3T+1P

B.Sc. In Computer Science : From 2017-2018 onward :
(General and Honors)

Semester I : Paper I

DSC1 : Programming Fundamentals Using C

Total Marks: 150 (4 credits for theory and 2 credits for practical)

Theory: 100

Practical: 50

Internal Assessment for theory : 20 marks ISA and 80 marks SEE

Workload :

Theory : 4 Lectures per week of 1 hours duration

Practical : 2 Practical sessions of 2 hours duration per week per batch . (correspond to total 4 lectures per week)

Theory : 60 Lectures

Course Objective: To provide an understanding of steps involved in problem solving and learning basics of programming in C. Includes practical skills to write programs for solving basic mathematical problems.

Learning Outcomes: By the end of this paper, the student will be able to list steps involved in problem solving, describe basic problem solving strategies, determine the best strategy for solving basic mathematical problems, design flowchart for basic mathematical problems, and model a flowchart into a C program.

Theory

1) Overview of programming :

(5 Lectures)

Introduction to computer based problem solving : Requirement of problem solving by computers, Problem definition, use of examples for problem solving, similarities between problems, problem solving strategies, Steps involved in problem solving

2) Program design and Implementation issues :

(5 Lectures)

Programs and algorithms, Top down design and stepwise refinement, construction of loops , basic programming constructs, implementations

3) Programming environment :

(3 Lectures)

Programming language classification, assemblers, examples of high level languages, compiler linking and loading, algorithms for problem solving.

4) Algorithms for Problem Solving :

(18 Lectures)

Examples : exchanging values of two variables, summation of set of numbers, decimal to binary , reversing of digits of integer, greatest common divisor of two numbers, to verify whether a integer is prime or not, organize a given set of numbers in ascending order, find a square root of a integer, factorial of a given number, generate Fibonacci numbers for n terms, to find the value of power of a number raised by a integer , reverse order elements of a array, find the largest number in a array, print elements of upper triangular matrix, multiplication of two matrices, compute roots of a quadratic equations

5) Overview of C :

(15 Lectures)

Structure of a C program, data types, Constants and variables, operators and expressions, operators : arithmetic, logical, relational, assignment, ternary, comma operators ; Control constructs: selection , iterative, branching statements; array constructs, Strings, basic I/O, functions, recursion. Macro, preprocessor directives

6) Pointers and structured data types**(9 Lectures)**

Pointers , structures and unions, enumerated data type

Implementation of arrays and structure using pointers , Sparse Matrices (Array and Linked Representation)

7) FILE HANDLING : Text and data file create, open , read and write . (5 Lectures)**Reference Books :**

- 1). Harsha Priya, R. Ranjeet, "Programming and problem solving through C language", Firewall Media
- 2). Jeri R. Hanly, Elliot B. Koffman, "Problem solving and program design in C", Pearson Addison Wesley.++
- 3). R. G. Dromey, "How to solve it by computer", PHI
- 4). E Horowitz, S Sahni, S Rajasekaran, "Fundamentals of computer algorithm", Galgotia.
- 5). Byron Gottfried, "Programming with C", Tata McGraw Hill
- 6). Forouzan, "A Structured Programming Approach using C", 2nd Edition, Cengage Learning India, 2008.

Practicals**DSC1P : Programming Fundamentals Using C (Lab)****Practical's : 60 Lectures**

Algorithms should be developed for solving a given problem. 'C' programs should be written based on the algorithms.

The following list of problems may be used as exercises :

1. Print the sum and product of digits of an integer.
2. Reverse a number.
3. Compute the sum of the first n terms of the following series $S = 1 + 1/2 + 1/3 + 1/4 + \dots$
4. Compute the sum of the first n terms of the following series $S = 1 - 2 + 3 - 4 + 5 - \dots$
5. Write a function that checks whether a given string is Palindrome or not. Use this function to find whether the string entered by user is Palindrome or not.
6. Write a function to find whether a given no. is prime or not. Use the same to generate the prime numbers less than 100.
7. Compute the factors of a given number.
8. Write a macro that swaps two numbers. WAP to use it.
9. Print a triangle of stars as follows (take number of lines from user):

```
*
***
*****
*****
*****
```

10. Perform following actions on an array entered by the user:
 - i) Print the even-valued elements
 - ii) Print the odd-valued elements
 - iii) Calculate and print the sum and average of the elements of array
 - iv) Print the maximum and minimum element of array
 - v) Remove the duplicates from the array
 - vi) Print the array in reverse order

The program should present a menu to the user and ask for one of the options. The menu should also include options to re-enter array and to quit the program.

11. Take the radius of a circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the value of area and circumference from the main() function.

- 12. Display Fibonacci series (i) using recursion, (ii) using iteration
- 13. Calculate Factorial of a number (i) using recursion, (ii) using iteration
- 14. Calculate GCD of two numbers (i) with recursion (ii) without recursion.
- 15 recursion
- 16 Pointers
- 17 File Handling

B.Sc. General Elective Papers
Semester – I GE1
(Credits : 04) - (3 + 1 Lab)

General Elective : GE-1 : IT Fundamentals

Course Objectives:

1. To provide an understanding in Information Technology Fundamentals.
2. To understand the basic use of IT in various fields.

Learning Outcome:

The students will be able to identify and appreciate the use of IT in daily life.

Theory

Introduction: Introduction to logical organization of computer, input and output devices (with connections and practical demo), keyboard, mouse, joystick, scanner, OCR, OMR, monitor, printer, plotter, primary memory, secondary memory, auxiliary memory. 15 L

User Interface: Operating system as user interface, system tools, utility programs 5 L

Database: Introduction to database, relational data model, Entity types, entity set, attribute and key 5 L

Networks: Definition of network, classification of network, LAN, MAN, WAN, distinction among the networks, Guided Media: Twisted pair, Coaxial cable, and Optical fiber. Unguided media: Microwave, Radio frequency propagation, Satellite, LAN Topologies: Ring, bus, star, mesh and tree topologies. 10 L

Internet Applications: Internet as a global network, Search Engine, Online education, Internet utilities – email, online banking, reservations etc. 6 L

Use of Computers in Education and Research: Data analysis, Heterogeneous storage, e-Library, Google Scholar, Domain specific packages such as SPSS, Mathematica etc. 4 L

Reference Books:

1. A. Goel, Computer Fundamentals, Pearson Education, 2010.
2. P. Aksoy, L. DeNardis, Introduction to Information Technology, Cengage Learning, 2006
3. P. K.Sinha, P. Sinha, Fundamentals of Computers, BPB Publishers, 2007

Practical: 30 lectures

Practical exercises based on Open Office/ MS Office tools using document preparation, spreadsheet handling packages and presentation software.

FYBA Semester I
Computer Fundamentals and Emerging Technologies
GE1 : Generic Elective as per CBCS

Credits: 3+1 Duration: 2 hrs. Marks: 100(Theory 75 + Practical 25)
Lectures: 45, Practical Lab 15 Sessions Batch Size: 10-15 per batch
One Theory Lecture = One Hour & One Practical Lab Session = Two Hours

Course Objectives: To provide an understanding of Fundamental Technology Concepts and Emerging Technologies in Computer Field. Includes practical skills in data capture, text editing with report formatting, effective presentation tools, efficient search techniques and online collaboration tools.

Learning outcome: The student will be able to recall types of computers and their usage. The student will remember different emerging technologies and be able to use Internet facilities. The student will be able to create blogs, collaborate on wikis, and create online data forms.

Unit I Lectures:15, Practical Lab:1 Sessions Marks(Theory-30, Practicals-1)

Computer Fundamentals (Theory)

Introduction: Introduction to computer system, uses, types.

Data Representation: Number system and Coding Schemes(ASCII and UNICODE).

Human Computer Interface: Relationship between Hardware and Software, Types of software, Operating system as user interface, utility programs.

Role of Computers in: Business, Manufacturing, Mobile Computing, Public Sector, Media, Defense Services.

Lab 1: Computer Fundamentals (Practicals)

- a. Features of MS Windows based OS and any of the Linux flavor, Setting up users and User rights in a computer, Adding a printer, Software Installation
- b. Troubleshooting basic computer connections.

Unit II Lectures: 15, Practical Lab: 10 Sessions Marks (Theory-10, Practicals-18)

Content/Data Management Tools (Theory)

User Generated Content: Blogs and Wikis.

Online Data Capture Tools: Types of data capture form templates (Personal, Work and Education). Question Formats for data capture (short answer, paragraph, multiple choice, check-box, drop-down, linear-scale, multiple choice grid). Data form design (Add new question, add section, add title/description/image/video). Data form distribution techniques (Send via email, publish on social media, send as link). Response management (Print responses, Export to spreadsheet, View analysis, Include analysis in word processing reports)

Text Formatting using Word Processing tools: Use of Templates, Working with document: Editing text, Find and replace text, Formatting, spell check, Autocorrect, Autotext; Bullets and numbering, Tabs, Paragraph Formatting, Indent, Page Formatting, Header and footer, section break, footnotes, bibliography and references. Tables: Inserting, filling and formatting a table; Inserting Pictures and Video; Managing Mail Merge: including linking with Database; Printing documents Creating Business Documents using the above facilities.

Data Presentation using Presentation tools: Slides, Fonts, Drawing, Editing; Inserting: Tables, Images, texts, Symbols, Media; Design; Transition; Animation; and Slide-show. Creating Business Presentations using above facilities

Lab 2.1 : Creating Content for the Web (Practicals)

- a. Create a blog using popular blogging tools like WordPress/ BlogSpot.
- b. Edit Wikipedia articles.

Lab 2.2 : Data Capture using Google Forms (Practicals)

- a. Create data forms to capture data for Event Registration, Event Feedback, Customer feedback/satisfaction on a product or service and Order Request.

Lab 2.3 : Report Formatting using Word Processing (Practicals)

- a. Draft an official letter for job interview invitation/ job appointment/ invitation to an event, use mail merge to input the recipients list linking with database.
- b. Given a project report in PDF format transfer to word processor software and format to include title page, specified Paragraph and Page Formatting (page size, orientation, line spacing, font type and font size, Indent, bullets, paragraph formatting) details, Acknowledgement page, Table of contents page, List of figures page, List of Tables page, bibliography, references, distinct headers for each chapter, page numbering in roman for initial pages and normal from first chapter. The document should be checked for spelling errors and corrected appropriately.
- c. Design a certificate in landscape orientation with a border around the document.
- d. Design a Garage Sale sign.

Lab 2.4 : Content Presentation using Presentation Software (Practicals)

- a. Preparing presentation in areas such as Impact of Social Media on Youth, Emerging trends in Mobile Technology include appropriate slide animation, slide transitions, sound recording, slide timings, customer feedback video.
- b. Export the presentation as video or save as slide show.
- c. Prepare handouts for audience.

Unit III Lectures: 5, Practical Lab: 3 Sessions**Marks(Theory -20, Practical -5)****Overview of Emerging Technologies**

Cloud Computing: Meaning, Features, & Service models – Infrastructure as a service, Advantages and disadvantages, Mobile Computing: Meaning, Business Applications of Mobile computing, Virtual reality & Augmented Reality : Meaning and applications , IOT - Internet of Things: Meaning & Application

Lab 3: Web Applications (Practicals)

- a. Scheduling tasks in Google Calendar
- b. Create/Upload documents / spreadsheets and presentations online.
- c. Share and Collaborate in real time
- d. Safely store and organize your work of Google Drive or OneDrive

Unit IV Lectures: 10, Practical Lab: 1 Sessions**Marks (Theory-15, Practical -1)****Computing Trends in Internet, Education and Research:**

Internet-role and importance, Web Server and Web clients like web browser or web app, IP addressing : Public Vs Private, Static Vs Dynamic, world wide web and related protocols, e-Library, Google Scholar.

Lab 4: Internet Applications (Practicals)

- a. Surfing the Internet, Using Email and Search Engines
- b. Advanced web search and translation services, Web search, image search, Search only for pages that contain (ALL the search terms contain the exact phrase you type, contain at least one of the words you type, do NOT contain any of the words you type, written in a certain language, created in a certain file format like ppt, pdf, rtf, doc, xls)
- c. Advanced search operators: Include search (“+” search), synonym search, OR search, Domain search, Numrange search, other advanced search features (Google, Local language, Technology Search, Date, Occurrences, Domains, Safe search),

Reference Books:

1. Introduction to Information Technology by ITL Education Solutions Limited, second edition.
2. ‘ O ’ Level made simple “introduction to ICT resources” by Satish Jain, Shashank Jain,

Shashi Singh & M. Geetha Iyer, BPB publication.

3. Computer Fundamentals fourth edition by Pradeep K. Sinha and Priti Sinha BPB publications
4. Information Technology The breaking wave by Dennis Curtin Tata McGraw-hill edition
5. Computer Fundamentals by A. Goel, Pearson Education, 2010.
6. Introduction to Information Technology by P. Aksoy and L. DeNardis, Cengage Learning, 2006

B.Sc. In Computer Science : From 2017-2018 onward :
(General and Honors)
Semester II

DSC4 : Data Structures

Total Marks: 150 (4 credits for theory and 2 credits for practical)

Theory: 100

Practical: 50

Internal Assessment for theory : 20 marks ISA and 80 marks SEE

Workload :

Theory : 4 Lectures per week of 1 hours duration

Practical : 2 Practical sessions of 2 hours duration per week per batch . (correspond to total 4 lectures per week)

Theory : 60 Lectures

Course Objective: To provide an understanding of data structures and functions that represent real world problems using programming in C. Includes practical skills to design data structures and writing functions for real world problems.

Learning Outcomes: By the end of this paper, the student will be able to define data structures, explain the applications of each data structure, apply data structures for real life problems, analyze real life problems for data and functionality, compare the efficiencies of different ways of solving problems, design recursive functions for special real life problems, and derive data structures and functions for real life problems using C programming.

Theory

Stacks (7 Lectures)

Implementing single / multiple stack/s in an Array; Prefix, Infix and Postfix expressions, Utility and conversion of these expressions from one to another; Applications of stack; Limitations of Array representation of stack

Linked Lists (10 Lectures)

Singly, Doubly and Circular Lists (Array and Linked representation); Normal and Circular representation of Stack in Lists; Self Organizing Lists; Skip Lists

Queues (6 Lectures)

Array and Linked representation of Queue, De-queue, Priority Queue.

Recursion (5 lectures)

Developing Recursive Definition of Simple Problems and their implementation; Advantages and Limitations of Recursion; Understanding what goes behind Recursion (Internal Stack Implementation)

Trees**(19 Lectures)**

Introduction to Tree as a data structure; Binary Trees (Insertion, Deletion, Recursive and Iterative Traversals on Binary Search Trees); Threaded Binary Trees (Insertion, Deletion, Traversals); Height-Balanced Trees (Various operations on AVL Trees).

Searching and Sorting**(7 Lectures)**

Linear Search, Binary Search, Comparison of Linear and Binary Search, Bubble sort, Selection Sort, Insertion Sort, Comparison of Sorting Techniques.

Hashing**(6 Lectures)**

Introduction to Hashing, Deleting from Hash Table, Efficiency of Rehash Methods, Hash Table Reordering, Resolving collision by Open Addressing, Coalesced Hashing, Separate Chaining, Dynamic and Extendible Hashing, Choosing a Hash Function, Perfect Hashing Function

Reference Books :

- 1). Aaron M. Tenenbaum, Moshe J. Augenstein, Yedidya Langsam, "Data Structures Using C and C++", Second edition, PHI, 2009.
- 2). Richard F. Gilberg, Behrouz A. Forouzan, "Data Structures : A Pseudocode Approach with C", Cengage Learning, 2 Edition (Paperback), 2007.
- 3). Ellis Horowitz, Sartaj Sahni, "Fundamentals of Data Structures in C", Universities Press, 2nd Edition, 2008.
- 4). Seymour Lipschutz: "Data Structures with C", Schaum's outlines, Tata McGraw-Hill, 2011

Practical's : 60 lectures

Suggested list of Practicals:

1. Write a program to search an element from a list. Give user the option to perform Linear or Binary search. Use Template functions.
2. WAP using templates to sort a list of elements. Give user the option to perform sorting using Insertion sort, Bubble sort or Selection sort.
3. Implement Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list and concatenate two linked lists (include a function and also overload operator +).
4. Implement Doubly Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
5. Implement Circular Linked List using templates. Include functions for insertion, deletion and search of a number, reverse the list.
6. Perform Stack operations using Linked List implementation.
7. Perform Stack operations using Array implementation. Use Templates.
8. Perform Queues operations using Circular Array implementation. Use Templates.
9. Create and perform different operations on Double-ended Queues using Linked List implementation.
10. WAP to scan a polynomial using linked list and add two polynomial.
11. WAP to calculate factorial and to compute the factors of a given no. (i)using recursion, (ii) using iteration(ii) WAP to display fibonacci series (i)using recursion, (ii) using iteration
12. WAP to calculate GCD of 2 number (i) with recursion (ii) without recursion
13. WAP to create a Binary Search Tree and include following operations in tree:
 - 15 WAP to convert the Sparse Matrix into non-zero form and vice-versa.
 - 16 WAP to reverse the order of the elements in the stack using additional stack.
 - 17 WAP to reverse the order of the elements in the stack using additional Queue.
 - 18 WAP to implement Diagonal Matrix using one-dimensional array.
 - 19 WAP to implement Lower Triangular Matrix using one-dimensional array.
 - 20 WAP to implement Upper Triangular Matrix using one-dimensional array.
 - 21 WAP to implement Symmetric Matrix using one-dimensional array.

- 22 WAP to create a Threaded Binary Tree as per inorder traversal, and implement operations like finding the successor / predecessor of an element, insert an element, inorder traversal.
- 23 WAP to implement various operations on AVL Tree.

General Elective :GE-2 : Multimedia and Web Design

Course Objectives:

- To introduce multimedia concepts.
- To introduce web designing concepts.

Learning outcomes: Students will be able to describe multimedia concepts and design web pages using hypertext markup language and cascaded style sheets.

Multimedia : Definition, Components, uses, applications	6L
Multimedia Input/Output Devices: scanner, camera, microphone, speaker, monitors, printers.	6L
Multimedia Storage Devices: CD ROMs, DVDs, Blue ray disk.	5L
Multimedia Tools: Sound editor, video editor, animator, authoring tools.	8L
Web Designing: Concept of website, website as a communication resource. Internet, intranet and extranet,.	6L
HTML: Introduction to hypertext markup language (html) document type definition, creating web pages, graphical elements, lists, hyperlinks, tables, web forms, inserting images, frames, use of CSS	7 L

Reference Books:

1. Scott Mitchell , Create your own website , SAMS Publication , 2008
2. Tay Vaughan, Multimedia : Making it work, Tata McGraw Hill, Seventh edition, 2006
3. J. Jeffcoate, Multimedia in Practice, Pearson Education, First Edition, 2007

Practical:

Practical exercises based on Open Office tools using presentation software, web design and development tools, image editing tools (Gimp) and animation tools such as Blender

1. Create an HTML document with the following formatting options:
 - I. Bold
 - II. Italics
 - III. Underline
 - IV. Headings (Using H1 to H6 heading styles) V. Font (Type, Size and Color)
 - VI. Background (Colored background/Image in background)
 - VII. Paragraph
 - VIII. Line Break
 - IX. Horizontal Rule
 - X. Pre tag

2. Create an HTML document which consists of:
 - I. Ordered List
 - II. Unordered List
 - III. Nested List
 - IV. Image

Optional

Implement the followings using Blender -

1. Create an animation using the tools panel and the properties panel to draw the following – Line, pe , oval, circle, rectangle , square, pencil , brush , lasso tool
2. Create an animation using text tool to set the font , size , color etc.
3. Create an animation using **Free transform tool** that should use followings-
Move Objects
Skew Objects
Stretch Objects
Rotate Objects
Stretch Objects while maintaining proportion
Rotate Objects after relocating the center dot
4. Create an animation using layers having following features- Insert layer, Delete layer, guide layer, Mask layer.
5. Modify the document (changing background color etc.)using the following tools
Eraser tool
Hand tool
Ink bottle tool
Zoom tool
Paint Bucket tool
Eyedropper tool
6. Create an animation for bus car race in which both starts from the same point and car wins the race.
7. Create an animation in which text Hello gets converted into GoodBye (using motion/shape tweening).
8. Create an animation having five images having fade-in fade-out effect.
9. Create an scene to show the sunrise (using multiple layers and motion tweening)

FYBA Semester II

Cyber Space and Cyber Security

GE2 : Generic Elective as per CBCS

Credits: 3+1

Duration: 2 hrs.

Marks: 100(Theory 75 + Practical 25)

Lectures: 45, Practical Lab 15 Sessions

Batch Size: 10-15 per batch

One Theory Lecture = One Hour & One Practical Lab Session = Two Hours

Course Objectives: To introduce computer networking, e-commerce and understand principles of cyber security, online threats and cyber laws and prepare students to adopt safe practices.

Learning outcomes : The student will be able to list cyber laws, describe computer networks, threats to cyber security, illustrate uses of Ecommerce and security measures for cyber safety

Unit I Lectures: 6, Practical Lab: 2 Sessions Marks (Theory -15, Practicals -2)

Basics of Computer Networking

Networking basics, Need for computer networks, Types of networks-LAN, MAN, WAN, Network Components – H/W, Software, Communication channels, Network Devices, Network topologies.

Lab1

- Basic Networking Setup of PC, Network commands like ipconfig, ping, traceroute, nslookup / dig etc, Setup of Home Router / Wifi Hotspot,
- Understanding of Firewall and Basic Firewall Setup, File and Printer Sharing,

connecting to share

- Setup of Email Clients like Outlook, FTP Clients and Upload / Download.
- Finding out public address, connection speeds etc.

Unit II Lectures: 10, Practical Lab: 04 Sessions Marks (Theory-15, Practicals-16)

E-Commerce

Definition, Hardware requirements, E-commerce and Trade Cycle, Electronic Markets, Electronic Data Interchange and Internet Commerce, Benefits and Risk, Types of E-commerce :Business to Business E-Commerce, Business to Consumer E-Commerce. Consumer to Consumer, Electronic Payment Systems: Smart Cards – Credit Cards – Wallets, Risks, E-Retail, Concept and Examples, E-Banking, Features and services , M-Commerce, Products and services

Lab 2

E-commerce

- Attempt to purchase a product online from any E-Commerce Site. Proceed till payment gateway. Check digital certificates (such as ebay.in and amazon.com)
- Write a review of an E-Commerce Site visited include: Site description, Site Design, ease in navigation , process for purchasing items, security, privacy, compare with competitors, customer service, best features of site etc.
- An E-commerce site case study: Include Target market/audience: who uses this service?

Revenue model: where does the money come from? Competitive environment: who else is competing in this market, or who might enter the market and threaten this company's position? Competitive advantage analysis: how is your case company attempting to gain an advantage: competing on cost? Differentiation? How are they promoting their products in the marketplace? ,How have they been doing - financial results if available?

Unit III Lectures: 15, Practical Lab: 02 Sessions Marks (Theory-25, Practicals-2)

Emerging threats in Cyber Space

Introduction to cyber space, Malware threats- Definition and types(Virus/ worms, Trojan, Rootkits, Spyware, Keyloggers). Social Engineering, Cyber Crimes – Definition, Types (DOS, Intellectual Property crimes, Unauthorized access to computer system or networks, Theft of information contained in electronic form, Cyber Stalking, Identity Theft, Forgery, E-mail Spoofing, E-mail bombing, Online gambling, Sale of illegal articles, Child pornography, Cyber Defamation, Salami attack, Phishing, Pharming, Data Diddling, Virus/ worm attack, logic bombs, Web jacking, Theft of computer system, physically damaging a computer system, Cyber warfare, Cyber terrorism.)

Lab 3

- Installation and Configuration of any free Antivirus Package eg. AVG/Avast etc., Using Antivirus Package for Threat Detection
- Browser security and Safety such as Understanding SSL and Certificates, checking URL of site for Phishing attempts
- Email Headers and Tracking, Identification of Phishing Emails

Unit IV Lectures: 7, Practical Lab 03 Sessions Marks (Theory-10, Practicals-2)

Online Privacy and Cyber Safety

Online Privacy – Introduction, Significance, Privacy Policy, Sensitive Personal Information, Social media – Usage, Safety. Online shopping – Introduction, Safety measures (Encryption of data authentication , SSL, Digital signatures, Digital Certificates) Online payments – Introduction, Types, Safe practices.

Lab 4

- Keeping passwords cyber secure-Choosing strong password,
- Privacy settings on Facebook, Social Media Safety
- Payment Systems Security concerns and Safe Practices

- Online Banking Security features, OpenPGP Tools.

Unit V Lectures: 7, Practical Lab 04 Sessions

Marks (Theory-10, Practicals-3)

Cyber Laws and Cyber Forensics

Cyber Laws: Evolution and Need for cyber law, The legal perspectives – Indian perspective, Global perspective, Information Technology Act(ITA) 2000, Provisions related to E-commerce, Provisions for cyber-crimes, Information Technology (Amendment)(ITAA) Act 2008, Adjudicating officer, CERT-IN- its role and powers.

Reporting Cyber Crimes, Cyber Forensics: Introduction, Evidence collection, Data Recovery, Cloning of Devices, Forensic Investigation phases – Acquisition and preservation, Authentication, Analysis, Documenting Evidence, Presentation of Evidence, Media sanitization.

Lab 5

- Use of Investigation tools such as Winhex for forensic investigation
- Data Recovery using winhex
- Use of Free data recovery tools like Recuva
- Mapping a given list of cyber-crimes to appropriate ITAA Act 2008 offence listed in http://www.naavi.org/ita_2008/index

Reference Books and web references

1. Rick Lehtinen and G. T. Gangemi, Computer Security Basics, O'Reilly Media, Inc.; 2nd Edition, 2006
2. Wall, David, (2007). Cyber Crime: The Transformation of Crime in the Information Age. Polity Publishing
3. Michael cross, Scene of the Cyber Crime, Syngress Publishing, Elsevier Publishing, 2nd Edition, ISBN 13: 978-1-59749-276-8
4. Chander, Harish, Cyber Laws and IT Protection, ISBN: 978-81-203-4570-6
5. Nina Godbole, SunitBelapure, "Cyber Security – Understanding Cyber Crimes, Computer Forensics and Legal Perspectives", Wiely India Pvt.Ltd.,ISBN - 978-81-265-2179-1
6. Frontiers of Electronic Commerce Ravi Kalakota & Andrew B Whinston, Pearson Education.
7. Cyber Laws, <http://deity.gov.in/content/cyber-laws>
8. www.cert.org

Frontiers of Electronic Commerce Ravi Kalakota& Andrew B Whinston, Pearson Education.

Second Year B.Sc.

Semester III

Computer Science - CSC103 : Database Management Systems

(Credits: Theory-04, Practicals-02) Theory : 60 Lectures

Practicals : 60 Lectures

Course Objectives :

- a) Provide a strong foundation in database concepts, technology, and practice.
- b) Practice SQL programming through a variety of database problems.
- c) Understand the use of concurrency and transactions in database

Learning Outcomes:

The students will be able to:

- Explain database concepts, technology and practice.
- Formulate SQL statements and queries using SQL programming.
- Explain the use of concurrency and transactions in database.

1. Introduction to Data Base Systems: File Systems versus a DBMS, The Relational Model, Levels of abstraction in a DBMS, Data independence, Queries in DBMS, Concurrent Access and Crash

Recovery, Structure of DBMS, Advantage of DBMS, People who deal with Databases.	4 L
2. Conceptual design and Entity Relationship model: Overview of Data Base Design, The ER model-features, Key Constraints, Participation Constraints, weak Entities, Class Hierarchies, Aggregation, Entity versus attribute, Entity versus relationship, Binary versus ternary relationship, aggregation versus ternary relationships.	6L
3. The Relational Model: Attributes and domains, Relations, Integrity Constraints, Key Constraints, Foreign Key Constraints, General Constraints, Enforcing Integrity constraints.	2L
4. Logical Database design ER to relational : Entity sets to tables, Relationship sets (without constraints) to tables, translating relationship sets with key constraints, translating relationship sets with participation constraints, translating weak entity sets, translating class hierarchies, translating ER diagrams with aggregation.	8L
5. Schema Refinement and Normal forms: Introduction, Why Schema Refinement? Functional Dependencies, Normal Forms: BCNF, Third Normal Form, Normalisation-Decomposition up to BCNF	5 L
6. Relational Algebra: Relational algebra operations- select, project, join, natural join, equijoin and their implementation	4 L
7. SQL: The Form of Basic SQL query, Condition specification, SQL Joins, Outer joins, Union, Intersect, Except, Nested queries, Aggregate Operators, Null values.	12L
8. SQL: Embedded SQL, Cursors, Dynamic SQL, Triggers and active databases	4 L
9. Transaction management : The concept and properties of transaction, transaction and schedule, Notion of consistency, Serializability, Isolation levels, Lock based concurrency control, concurrency control without locking, deadlocks	10L
10. Crash Recovery: Introduction to crash recovery, Recovery and atomicity, Log based recovery, Shadow paging	5 L

Text Books :

1). Abraham Silberschatz, Henry F. Korth, S. Sudarshan, "Database System Concepts", McGraw Hill Education, 6th Edition.

Reference Books :

- 1). Ramez Elmasri, Shamkant B. Navathe, "Fundamentals of Database Systems", Pearson Education, 7th Edition
- 2). Raghu Ramakrishnan, Johannes Gehrke, "Database Management Systems", McGraw Hill Education, 3rd Edition.

(Practicals)

List of Practical (a minimum of 15 Practical need to be completed)

1. Gathering information, Analysing data, ER Diagram, Reduction to Tables.
2. Creation/modification of database tables using DDL statements and GUI tools of the DBMS software.
3. SQL queries
4. SQL Joins
5. Stored Procedures, Triggers
6. Views and User management, granting/revoking privileges, roles.
7. Report Generation using a reporting tool
8. Database Design : Normalisation examples
9. Use of any front-end to develop forms on desktop/web based applications.
10. Database User Level Security for Databases for tables, Views.

Note :

- a. A minimum of 3 example sets covering all concepts should be done for topics under Serial No 1(ERDs), 3 (SQL) and 8 (Normalisation). (each is a separate practical)

b. Practicals should be done using a DBMS software like Oracle, SQL Server, MYSQL, POSGRES and a compatible Front-End Tool

Second Year B. Sc.	Semester III
Computer Science -CSS103 : Programming in Python (Credits: Theory-03, Practicals-01) Theory : 45 Lectures Practicals : 30 Lectures Pre-requisites : Basic working knowledge of Computers and Internet	
Course Objectives: 1. To introduce programming concepts using Python. To introduce object oriented programming concepts.	
Learning Outcome: Students will be able to write programs in python language and create an application.	
(Theory)	
1. Python Interpreter, Python Shell, strings, relational operators, logical operators, precedence of operators, bitwise operators, variables and assignment statements, script mode, functions, modules, command line arguments, control structures- if conditional statements, iteration for and while statements, break, continue and pass statements. 4L 2. Data types- Boolean, numbers, coercing integers to floats and vice versa, numerical operations, lists, creating a list, slicing a list, adding and removing items from a list, searching for values in a list, tuples, immutability property, converting tuples into a list, sets, set operations, dictionaries, strings, Unicode, formatting strings, docString, modules, packages, scope, recursion 12 L 3. Object Oriented Concepts- Classes, Objects, Abstract Data types, polymorphism, encapsulation, modifier, accessor methods, static method, adding methods dynamically, composition, inheritance, built-in functions for classes. 14 L 4. Files, Exceptions 4 L 5. Applications of Python - use of Python libraries such as Matplotlib, Pandas, using databases with python, collecting information from Twitter etc. (at least three applications to be covered). 11 L	
Text book : 1) Taneja Sheetal, Kumar Naveen , “Python Programming - A modular approach”, Pearson	
Reference book: 1). Guttag John V., “Introduction to Computation and Programming using Python”, MIT Press, 2 nd Edition.	
(Practicals)	
List of Practicals : (at least 8 practicals from the following) 1) a) Write a function that returns the sum of digits of a number, passed to it as an argument. b) Write a function that returns True or False depending on whether the given number of a palindrome. c) Take the radius of circle as input from the user, passes it to another function that computes the area and the circumference of the circle and displays the values. d) Write a function that finds the sum of the n terms of the following series: $1 - x^2/2! + x^4/4! - x^6/6! + \dots x^n/n!$	
2) Perform following actions on a list : Print the even-valued elements Print the odd-valued elements	

Calculate and print the sum and average of the elements of array

Print the maximum and minimum element of array.

Remove the duplicates from the array

Print the array in reverse order

3) a) Define a function which can generate and print a list where the values are square of numbers between 1 and 20 (both included). Then the function needs to print all values except the first 5 elements in the list.

b) Write a program which takes 2 digits, X,Y as input and generates a 2- dimensional array. The element value in the i-th row and j-th column of the array should be $i*j$.

4) a) Write a program that accepts sequence of lines as input and prints the lines after making all characters in the sentence capitalized.

b) Write a program that accepts a sentence and calculate the number of letters and digits.

c) Given an array of integers, find two numbers such that they add up to a specific target number.

5) a) Write a function that takes a list of values as input parameter and returns another list without any duplicates.

b) Write a program that takes a sentence as input from the user and computes the frequency of each letter. Use a variable of dictionary type to maintain the count.

6) a) Write a recursive function that multiplies two positive numbers a and b and return the result. Multiplication is to be achieved as $a+a+a$ (b times).

b) Write a recursive function that inserts the element x at every nth position in the given list and returns the modified list.

7) a) Given a list of strings, return the count of the number of strings where the string length is 2 or more and the first and last characters of the string are the same

b) Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. e.g. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']

8) Define a class Student that keeps track of academic record of students in a school. The class should contain the following data members:

rollnum - roll number of the student

name - name of the student

marksList - List of marks in 5 subjects

stream - A: Arts, C: Commerce, S: Science

percentage - percentage computed using marks

grade - grade in each subject computed using marks

division - division computed on the basis of overall percentage

The class should support the following methods:

a. `__init__` for initializing the data members

b. `setMarks` to take marks for five subjects as an input from the user

c. `getStream` for accessing the stream of the student.

d. `Percentage` for computing the overall percentage of for the student.

e. gradeGen that generates grades for each student in each course on the basis of marks.

Marks	Grade
≥ 90	A
< 90 and ≥ 80	B
< 80 and ≥ 65	C
< 65 and ≥ 40	D
< 40	E

f. division for computing division on the basis of the following criteria based on overall percentage of marks scored:

Percentage	Division
≥ 60	I
< 60 and ≥ 50	II
< 50 and ≥ 35	III

g. __str__ that displays student information.

9) Define a base class Vehicle, having attributes registration number, make, model and color. Also, define classes PassengerVehicle and CommercialVehicle that derive the class Vehicle. The PassengerVehicle class should have additional attribute for maximum passenger capacity. The CommercialVehicle class should have an additional attribute for maximum load capacity. Define __init__ method for all these classes. Also, get and set methods to retrieve and set the value of the data attributes.

10) Define classes Car, Autorickshaw and Bus which derive from the PassengerVehicle class mentioned in the previous question. The Car and Bus should have attributes for storing information about the number of doors, not shared by Autorickshaw. The Bus should have Boolean attribute doubleDecker not shared by Car and Autorickshaw. Define __init__ method for all these classes. Also define get and set methods to determine and set the value of the data attributes.

11) Develop a program to sort the employee data on the basis of pay of the employees using i) selection sort ii) bubble sort. iii) insertion sort. Consider a list L containing objects of class Employee having empNum, name and salary.

12) Write a function that takes two file names, file1 and file2 as input. The function should read the contents of the file file1 line by line and should write them to another file file2 after adding a newline at the end of each line.

13) Write a function that reads a file file1 and displays the number of words and the number of vowels in the file.

14) Write a function that reads the contents of the file Peom.txt and counts the number of alphabets, blank spaces, lowercase letters and uppercase letters, the number of words starting with a vowel and the number of occurrences of word “beautiful” in the file.

15) Write a function that takes two files of equal size as an input from the user. The first file contains weights of items and the second file contains corresponding prices. Create another file that should contain price per unit weight for each item.

Note : Testing and Debugging tools to be used during the practical sessions.

Second Year B. Sc.

Semester IV

Computer Science –CSC104: Computer Organization and Operating Systems

(Credits: Theory-04, Practicals-02) Theory : 60 Lectures

Practicals : 60 Lectures

Learning Objectives:

To give knowledge about fundamentals of Computer System Architecture and different types of Operating Systems.

Learning Outcomes:

The students will be able to:

- Explain Computer Organization and Operating Systems concepts and various OS technologies in use.
- Explain the various internal processes that occur within a given Operating System.
- Explain the use of concurrency and transactions in Operating Systems.
- Discuss the various techniques used in managing memory of a computer.

Explain I/O management and security management in Operating Systems.

Theory

1. Introduction to digital electronics Logic gates, Boolean algebra, combinational circuits, circuit simplification, registers, counters and memory units.	4 L
2. Data Representation and Basic Computer Arithmetic Number systems, complements, fixed and floating point representation, character representation, addition, subtraction, magnitude comparison.	4L
3. Basic Computer Organization and Design Computer registers, bus system, instruction set, instruction cycle, memory reference, input-output and interrupt, Register organization, arithmetic and logical micro-operations, stack organization, Instruction formats, addressing modes, instruction codes, machine language, assembly language, RISC, CISC architectures, pipelining and parallel architecture.	14 L
4. Input-Output Organization Input / Output, External Devices, I/O Modules, Programmed I/O, Interrupt-Driven I/O, Direct Memory Access.	8L
5. Introduction to Operating Systems Basic OS functions, resource abstraction, types of operating systems–multiprogramming systems, batch systems , time sharing systems; operating systems for personal computers & workstations, process control & real time systems, network operating system, mobile operating systems.	5 L
6. Operating System Organization Processor and user modes, kernels, system calls and system programs.	5L
7. Process Management	8 L

System view of the process and resources, process abstraction, process hierarchy, Process Scheduling, non-pre-emptive and preemptive scheduling algorithms; concurrent processes, deadlocks.

8. Memory Management 7 L
Physical and virtual address space, paging, segmentation, virtual memory, Cache memory.
9. File , I/O Management , Protection and Security 5 L
Directory structure, file operations, file allocation methods, device management. Authentication and Authorization.

Text Books :

- 1). M. Morris Mano, Rajib Mall, “Computer System Architecture”, Revised 3rd Edition, Pearson Education
- 2). Carl Hamacher, Zvonko Viranesic, Safwat Zaky, “Computer Organization”, 5th Edition, McGraw Hill
- 3). Abraham Silberschatz, Peter B. Galvin, Greg Gagne ,“Operating Systems Concepts”, International Student Version, Wiley Student Edition
- 4). Andrew S. Tanenbaum, Herbert Bos, “Modern Operating Systems”, 4th Edition, Pearson Education

Reference Books :

- 1). William Stallings, “Computer Organization and Architecture : Designing for Performance”, 9th Edition, Prentice Hall of India
- 2). Milan. Milenkovic, “Operating Systems Concepts and design”, 2nd Edition, Tata McGraw Hill.
- 3).Garry Nutt, “Operating Systems: A Modern Perspective”, 2nd Edition, Pearson Education

(Practicals)

List of Practicals

(At least 15 Practical

s from the following)

- 1). Introduction to 8086 architecture and instruction set and Writing assembly language programs in 8086 using MASM or compatible assembler either in windows or Linux.
- 2) Find the sum of $1 + 2 + 3 + \dots + n$
- 3). Display the multiplication table of a number
- 4) Store and retrieve numbers from memory
- 5). Sort the numbers stored in the memory
- 6). Installing Linux / Windows Operating System, Partitioning and formatting disk, Installing applications device drivers, working with files, mounting file systems, checking system space, creating, modifying and deleting user accounts
- 7). Study of Basic commands of Linux.
- 8). Study of Advance commands of Linux.
- 9). Shell Programming in Unix/Linux, arithmetic operations, loops, files
Ex. Write a BASH shell script prime which will accept a number b and display first n prime numbers in standard output.
- 10). Shell scripting using general-purpose utilities.
Ex. A) Write a menu driven shell script which will print the following menu and execute the given task to display result on standard output.
 - a) Display calendar of current month
 - b) Display today’s date and time
 - c) Display usernames those are currently logged in the system
 - d) Display your name at given x, y position
 - e) Display your terminal number
 - f) Exit
- 11). Shell programming using filters (including grep, egrep, fgrep)

- 12). Write a shell script to validate the entered date. (eg. Date format is : dd-mm-yyyy)
- 13). Write a shell script to check entered string is palindrome or not
- 14). WRITE A PROGRAM in C using fork() and/or exec() commands where parent and child
Execute :
- same program, same code.
 - same program, different code.
 - before terminating, the parent waits for the child to finish its task.
- 15). WRITE A PROGRAM in C to report behavior of Linux kernel including kernel version, CPU type and model. (CPU information)
- 16). WRITE A PROGRAM in C to report behavior of Linux kernel including information on configured memory, amount of free and used memory. (memory information)
- 17). WRITE A PROGRAM in C to print file details including owner access permissions and file access time, the file name is given as argument.
- 18). WRITE A PROGRAM in C to copy files using system call

Second Year B. Sc.

Semester IV

Computer Science -CSS104: Web Application Development using Flask

(Credits: Theory-03, Practicals-01) Theory : 45 Lectures Practicals : 30 Lectures

Pre-requisites : 1) Should be able to write code in Python
2) Knowledge of object oriented concepts and databases

Course Objectives:

- To learn how to create a basic web page using HTML and CSS.
- To perform basic database operations using Python Flask Framework

Learning Outcome:

Students will be able to create a basic web page using Html and CSS and perform basic database operations using python flask framework.

(Theory)

- Introduction to world wide web, how the web works, Introduction to HTML5, anatomy of an HTML element, nesting elements, block versus inline elements, empty elements, attributes, Boolean attributes, anatomy of a HTML document, entity references, HTML comments, head, title, body, metadata, headings, paragraphs, lists, emphasis and importance, hyperlinks, anatomy of a link, block level links, URLs, absolute versus relative URLs, email links, description lists, quotations, abbreviations, superscript, subscript, date and time , image. 3 L
- Document and Website Structure, Structuring Content - semantic tags -header, navigation bar, main content, sidebar, footer, non-semantic wrappers- div and span, line breaks and horizontal rules, html table basics, span rows and columns, HTML table and advanced features and accessibility, designing form, fieldset, legend widgets, sending form data, form data validation, iframe. 4 L
- Introduction to CSS, how browsers affect CSS, internal and external style sheet, CSS syntax, selectors - simple selectors, attribute selectors, combinators, multiple selectors, pseudo-classes, pseudo-elements, cascade and inheritance, box model, fundamental text and font styling, values, units, colors, media queries, layout- static, liquid, adaptive and responsive, floats, positioning, flex box , grids. 14 L
- DOM, Introduction to JavaScript, statements, syntax, variables, functions, Event

handlers, Introduction to Bootstrap Framework.	4 L
5. Dynamic Pages v/s Static Pages , HTTP Request/Response Model, HTTP methods get and post, Installing Flask, Basic Flask Application : __init__.py package, Using python decorators to modify the function that follows it. Routing : route() decorator, creating URL routes, passing variables, URL Binding : url_for() function, Flask-HTTP methods.	4 L
6. Jinja2 Templating Engine : Separating code and User interface, render template() function, Conditional Statements, Loops, Template Inheritance.	2 L
7. Flask Extensions, Installing flask-wtf extension, Flask WTF : Disadvantages of HTML Forms; flexible wtf forms, rendering and validation library; Standard Form Fields in WTF : TextField, BooleanField, IntegerField, RadioField, SelectField, TextareaField, PasswordField, SubmitField; Validator class : Length, NumberRange, URL; a Form example : Form class, Templates (HTML), Views, Receiving form data, Field Validations, Generating Links.	5 L
8. SQLAlchemy a ORM for many relational databases, installing flask extension for SQL Alchemy, Flask-SQLAlchemy configuration for SQLite/PostgresSQL/MYSQL database, Database Models, Creation of a Migration Repository, Database Migration, Database Upgrade and Downgrade, Database Relationships, Basic database operations using SQLAlchemy	5 L
9. Session Handling in Flask, Introduction to Flask-Login, The User Model for Flask-Login, Password Hashing, User Loader Function, Logging Users In, Logging Users Out, Showing the Logged in users in templates, User registration	4 L

Text Books :

1. Laura Lemay , Rafe Colburn , Jennifer Kyrnin, “Mastering HTML, CSS & JavaScript Web Publishing”, BPB Publications
2. Alex Libby, Gaurav Gupta, Asoj Talesra , “Responsive Web Design with HTML5 and CSS3 Essentials”, PACKT Publishing
3. Thomas Powell, “HTML & CSS: The Complete Reference”, Fifth Edition, MCGraw Hill
4. Miguel Grinberg, “Flask Web Development 2e”, OReilly

Reference Books :

1. Elisabeth Robson, Eric Freeman, Head First HTML with CSS & XHTML A Learner’s Companion to HTML, CSS and XHTML, O’Reilly Media
2. Jon Duckett, “Web Design with HTML, CSS, JavaScript and jQuery”, Wiley; Pck edition
3. Jack Stouffer , “Mastering Flask”, Packt Publishing Limited

(Practicals)

List of Practicals

(at least 8 Practicals from the following)

1. a). .Create web pages using text, paragraphs, header tags, links, lists tags
b). Create web pages using table tags, column and row span
c). Creating tables using scope, id and header attributes
2. a). Design a form using widgets
b). Form validations
3. a). Use of different CSS selectors, pseudo-classes and pseudo-elements
b). Design a landing page layout
c). Use of CSS font style
4. Form validation using JavaScript
5. Use of Bootstrap
6. Installing Python3, Creating virtual environment , Installing Flask, Flask extensions flask-wtf, flask-bootstrap, flask-sqlalchemy, flask-migrate, flask-login
7. Using Jinja 2, template directory, render_template to display content from a Python

- Dictionary. Use Jinja2 conditional statements, loops, template inheritance
8. Create a Login Form using flask-wtf and flask-bootstrap
 9. Create a User Registration Form using flask-wtf and flask-bootstrap
 10. Develop a database application using Python Flask Framework

Course Code : CSC105		Course Title : Computer Networks		Semester : V	
Theory Marks : 100		Theory Credits : 04		Theory Lectures : 60	
Practical Marks : 50		Practical Credits : 02		Practical Lectures : 60	
Total Marks : 150		Total Credits : 06			
Course prerequisites : CSC101 Introduction to Programming using C					
Course objectives:					
<ul style="list-style-type: none"> • To provide a strong background of Network Concepts. • To be familiar with the components required to build and design different types of networks. • To explain the various protocols associated with the network layers 					
Learning Outcomes : Upon completion of the course, students should be able to:					
<ul style="list-style-type: none"> • Describe the network models and networks based on type and topology. • Categorize and use transmission media based on their characteristics and applications. • Detect and correct errors using various techniques. • Explain different protocols for data transmission at the DLL. • Be able to setup networks and also implement subnetting. • Be able to apply different transport and application layer protocols. 					
Course content:					
Unit		Topic			
#	Title	#	Content	Lectures	
I	Data Communication	A	Introduction: Beginnings of Networking and data communication, ARPAnet	10L	
		B	Networks: Components and Categories, Types of Connections, Topologies, Transmission Modes		
		C	Switching: Circuit switching, Message switching, Packet switching,		
		D	Protocols and Standards: Layered Architecture, OSI model, TCP/IP model;		
		E	Applications of Networks		

II	Physical Layer	A	Functions of Physical layer	8L	
		B	Data Encoding: Manchester, Differential Manchester		
		C	Transmission Media: Twisted pair, Coaxial Cable, Fiber Optics, Wireless Media		
		D	Physical layer Devices: Hub, Repeater		
III	Data Link Layer	A	Functions of Data link layer	14L	
		B	Data Framing techniques: Character Count, Character Stuffing, Bit Stuffing		
		C	Error detection and correction: Parity, CRC, Hamming code		
		D	Elementary Data Link Protocols: Stop and wait, Sliding window protocols - Go back-N: ARQ, Selective repeat ARQ		
		E	MAC Sublayer, Random Access Protocols: ALOHA, CSMA, CSMA/CD, CSMA/CA, Collision free protocols		
		F	Network Standards: IEEE 802.3 (Ethernet) frame format , Categories of standard ethernet – 10BaseT, 10BaseF, Bridged ethernet, separating collision domains , Switched ethernet , Fast ethernet IEEE 802.11 Architecture, frame structure		
		G	Data Link layer devices: Bridges, Switches		
IV	Network Layer	A	Functions of Network layer	12L	
		B	Network Service types: Virtual Circuits, Datagrams		
		C	Routing Algorithms: Shortest path routing, Flooding, Distance Vector routing,		

			Link State routing; Hierarchical Routing		
		D	Congestion Control: Algorithms & Congestion Prevention Policies		
		E	Internet Protocols: IP Frame Format, IP Addressing, Subnets, Internet Control Protocols: ICMP, ARP, RARP, DHCP		
		F	Internetworking, Network layer device : Routers		
V	Transport Layer and Application Layer	A	Functions of Transport layer	10L	
		B	Transport Services: Connectionless, Connection-oriented, Transport service primitives Berkley sockets, Gateways		
		C	Transport layer Protocols: User Datagram Protocol, Transmission Control Protocol; Quality of Service parameters		
		D	Functions of Applications layer	06L	
		E	Electronic Mail; Domain Name System		

Text Book:

Behrouz A. Forouzan; Data Communications and Networking , McGraw Hill Education; Fifth Edition

References:

Andrew S. Tanenbaum; Computer Networks, Pearson Education India;5th Edition

Suggested List of Practical :

1. Create scenario and study the performance of network with star topology through NS-2 simulation.
2. Implementation of framing using Bit stuffing and Character stuffing
3. Simulation of Stop and Wait Protocol using NS-2.
4. Implementation of Dijkstra algorithm for Shortest Path Routing.
5. Configuring TCP/IP on a desktop.
6. Using diagnostic Network Commands: ping, traceroute, netstat, nslookup.
7. Simulate a Mobile Adhoc network (MANET) using NS2
8. Using Network protocol analyzer tool like ethereal (wireshark) or tcpdump to analyze network traffic

9. IP address manipulation (Extract network id and Host id given netmask)
10. Simulation of Congestion Control Algorithms using NS2
11. Implementation of IP fragmentation and reassembly.
12. Simple TCP client and server application (Single server-single client) – String manipulation and arithmetic operations.
13. Simple UDP client and server application (Single server-single client) – String manipulation and arithmetic operations.

Note :

Languages/Tools: C/C++, NS-2, Wireshark/TCPDump should be used for practicals.

Course Code : CSC106	Course Title : Object Oriented Programming	Semester : V
Theory Marks : 100	Theory Credits : 04	Theory Lectures : 60
Practical Marks : 50	Practical Credits : 02	Practical Lectures : 60
Total Marks : 150	Total Credits : 06	

Course Pre-requisite : CSC101 Introduction to programming using C

Course objectives:

- To present the object oriented method, in viewpoint of software engineering — of the methods, tools and techniques for developing quality software in production environments.
- To study how practicing software developers, in industrial as well as academic environments, can use object technology to improve the quality of the software they produce
- Introduce Java Programming Environment and Design Patterns

Learning Outcomes : Upon completion of the course students should be able to:

- Use the characteristics of an object-oriented programming language in a program.
- Use the basic object-oriented design principles in computer problem solving.
- Use the basic principles of software engineering in managing complex software project
- Write Java programs using classes and object
- Implement Design Patterns in Java Programs

Course content:

Unit		Topic		
#	Title	#	Content	Lectures
I	CRITERIA OF OBJECT ORIENTATION		<ul style="list-style-type: none"> • On the criteria • Method and language • Implementation and environment • Libraries 	4

II	TOWARDS OBJECT TECHNOLOGY		<ul style="list-style-type: none"> • The ingredients of computation • Functional decomposition • Object-based decomposition • Object-oriented software construction • Issues 	5	
III	THE STATIC STRUCTURE: CLASSES		<ul style="list-style-type: none"> • Objects are not the subject • Avoiding the standard confusion • The role of classes • A uniform type system • A simple class • Basic conventions • The object-oriented style of computation • Putting everything together 	6	
IV	THE RUN-TIME STRUCTURE: OBJECTS		<ul style="list-style-type: none"> • Objects • Objects as a modelling tool • Manipulating objects and references • Creation procedures • More on references • Operations on references • Attachment: reference and value semantics • Dealing with references: benefits and dangers 	6	
V	MEMORY MANAGEMENT		<ul style="list-style-type: none"> • What happens to objects? • The casual approach • Reclaiming memory: the issues • Programmer-controlled deallocation • The component-level approach • Automatic memory management • Reference counting • Garbage collection • Practical issues of garbage collection 	6	
VI	INTRODUCTION TO INHERITANCE		<ul style="list-style-type: none"> • What is inheritance? • Overriding and Polymorphism • Typing for inheritance • Dynamic binding • Deferred features and classes • The meaning of inheritance • The role of deferred classes 	6	

VII	MULTIPLE INHERITANCE		<ul style="list-style-type: none"> • Examples of multiple inheritance • Feature renaming • Flattening the structure • Repeated inheritance 	3	
VIII	EXCEPTION HANDLING		<ul style="list-style-type: none"> • Basic concepts of exception handling • Handling exceptions • An exception mechanism • Exception handling in Java 	3	
IX	GENERICITY		<ul style="list-style-type: none"> • Horizontal and vertical type generalization • The need for type parameterization • Generic classes • Arrays • Generics and collection framework in Java 	9	
X	DESIGN PATTERNS : INTRODUCTION		<ul style="list-style-type: none"> • What is a Design Pattern? • Describing Design Patterns. • How Design Patterns solve Design Problems • How to select a Design Pattern • How to Use a Design Pattern 	2	
XI	CREATIONAL PATTERNS		<ul style="list-style-type: none"> • Factory Method • Prototype • Singleton 	2	
XII	STRUCTURAL PATTERNS		<ul style="list-style-type: none"> • Adaptor • Composite • Decorator • Façade • Proxy 	4	
XIII	BEHAVIORAL PATTERNS		<ul style="list-style-type: none"> • Chain of Responsibility • Command • Iterator • Observer • State • Strategy 	4	

Text Books:

1. Bertrand Meyer , Object Oriented Software Construction, Prentice Hall; Second edition
2. Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides, **Design Patterns : Elements of Reusable Object-Oriented Software**, Pearson
3. Khalid A. Mughal and Rolf W. Rasmussen, A Programmer's Guide to Java SCJP Certification, Addison-Wesley

Reference Books

1. Kathy Sierra, Bert Bates , Head First Java - A Brain-Friendly Guide, O'Reilly, Second Edition
2. Kathy Sierra, Elisabeth Freeman, Head First Design Patterns - A Brain-Friendly Guide, O'Reilly

Suggested list of practical :

Programs covering the following topics may be done :

1. Use of command line environment and run-time environment in Java (javac and java)
2. Creating classes
3. Constructors and overloading
4. Object composition using references
5. Use of standard libraries like Math, String, util.*
6. Inheritance
7. Overriding, polymorphism and dynamic binding
8. Abstract class, interfaces and multiple interface inheritance
9. Use of static keyword
10. Exception handling
11. Arrays
12. Collection framework – ArrayList, Maps
13. Minimum one exercise on each design pattern

Tools like Notepad, Eclipse may be used to do the practical.

Course Code : CSC107		Course Title : Software Engineering		Semester : V	
Theory Marks : 100		Theory Credits : 04		Theory Lectures : 60	
Practical Marks : 50		Practical Credits : 02		Practical Lectures : 60	
Total Marks : 150		Total Credits : 06			
Course prerequisites : -					
Course objectives: To study various methods used for software development with a stress on Agile Software Development					
Learning Outcomes : Upon completion of the course students should be able to::					
<ul style="list-style-type: none"> • Explain Evolution and fundamentals of software engineering methods • Apply Agile software development method – Scrum • Apply refactoring techniques • Perform software testing using various quality assurance methods • Explain Source Control Tools 					
Course content:					
Unit		Topic			
#	Title	#	Content	Lectures	
I	Introduction to Software Engineering		Introduction to Software Engineering, Software Development phases(Requirements, Analysis, design and implementation, testing and maintenance), SDLC, Waterfall methodology, Prototyping and Iterative, Reverse engineering, reengineering	8	
II	Introduction to Source Control tools		Introduction to Source Control tools - versioning, check-in/checkout, commit, branching, merging, synchronization	4	
III	Agile Approach		Agile Approach: Agile Framework, Agile Manifesto, Agile Principles, Extreme Programming, Scrum	8	
IV	Software Project Management using Scrum		Software Project Management using scrum : User stories, Estimation using story points, sprint, backlog(product and sprint), Scrum team, scrum artifacts, scrum ceremonies	20	

V	Design and Implementation using XP	Design and implementation using XP: TDD, refactoring(code smells and refactoring techniques) , Unit testing, Pair Programming	10	
VI	Quality Assurance	Quality assurance (Verification & Validation): Testing approaches, Types of testing, testing tools- JUnit, Selenium, Build tools, Iteration and Release planning, Introduction to Continuous Integration	10	

Text Books:

1. Pankaj Jalote , Integrated Approach to Software Engineering, Narosa Publishing House
2. Chris Sims and Hillary Louise Johnson , Elements of Scrum, Dymaxicon, LLC
3. Martin Fowler, Refactoring, Addison Wesley; 2nd edition

Reference Books:

1. Ken Schwaber, Mike Beedle , Agile Software Development with Scrum, Pearson Education
2. S. Kenneth Rubin, Essential Scrum: A Practical Guide to the Most Popular Agile Process, Pearson Education
3. Kent Beck, Extreme Programming Explained: Embrace Change, Addison Wesley, 2nd Edition

Suggested List of Practical :

1. Git
2. User stories, Estimation
3. Burndown charts, Scrum board, Trello,
4. JUnit, Selenium
5. Refactoring exercises(pair programming)
6. Debugging and defect tracking using Bugzilla
7. Maven Build
8. Javadoc

The above practical should be done using a mini project using scrum and by performing refactoring exercises using pair programming. The tools Eclipse, Git, Selenium, Bugzilla, Trello are suggested.

Course Code : CSC108	Course Title : Mobile Application Development	Semester : VI
Theory Marks : 100	Theory Credits : 04	Theory Lectures : 60
Practical Marks : 50	Practical Credits : 02	Practical Lectures : 60
Total Marks : 150	Total Credits : 06	

Course prerequisites :

- 1). CSC106 Object Oriented Programming
- 2). CSC103 Database Management Systems

Course objective : Introduce mobile application development for the Android platform using XML and Java/Kotlin. Include developing simple applications that could run on Android phones and tablets. Cover Android application terminologies, components and coding.

Learning Outcomes : Upon completion of the course students should be able to :

- Describe the anatomy of a mobile app.
- Use Android components in designing simple mobile applications.
- Identify the significance of each of the Android basic building blocks and determine when to use which component.
- Discuss the data storage options available on android platform and perform basic CRUD operations on persistent data.
- Design complete Android app by integrating the android building blocks and using firebase as backend tool.

Course content:

Unit		Topic		
#	Title	#	Content	Lectures
I	Introduction to Mobile Apps and Android	A	Introduction: Need for Mobile Apps. Different types of Mobile Apps. Android vs. Other mobile platforms Open Handset Alliance (OHA) Features of Android Android Limitations	03
		B	Mobile Navigation: Basic patterns, Pros and Cons Screen independent design - Resolution and density independence (px, dip, dp, sip, sp)	02

		C	Android Pre-requisites: Java/Kotlin Programming fundamentals Introduction to XML Introduction to Build System (Gradle)	05	
		D	Android Architecture: Overview of Android Architecture Internals Linux Kernel, Libraries, Android Runtime, Application Framework, Dalvik VM and .apk files Emulator – Android Virtual Device: Definition, Features, Examples	03	
		E	Android Basic Building blocks: Activities, Services, Broadcast Receivers and Content Providers UI Components: Views and notifications	02	
II	Activities and UI	A	Introduction: Activities, Views, layouts (LinearLayout and RelativeLayout and WebView) Responsive UI with ConstraintLayout Significance of application manifest file	04	
		B	Creating User Interface (UI): Common UI components (TextView, EditText, Button, Checkbox, RadioButton , ToggleButton, Spinner, Pickers) Activity lifecycle Understanding the exception handler	04	
		C	Event Handling: onClick(), onLongClick(), onFocusChange(), OnKey(), onTouch(), onCreateContextMenu(), onCreateOptionsMenu()	04	
		D	Intents: Intents uses, intent types (Implicit & Explicit), Passing data(Direct, Bundle & Parcelable) Implicit Intents – Intent Filters and Intent Resolution Process, Pending intents	04	
		E	Advanced UI: Building Layouts with an Adapter (GridView & ListView), Custom Adapters, Menus (Options menu & Context menu), Toast, Custom Toast, Dialogs, Status bar Notifications.	04	
III	Broadcast Receivers and Services	A	Broadcast Receivers (BR): Broadcast receiver registration (Static & Dynamic), Broadcast Receiver Classes, Sticky & non-sticky BR, BR Security ,Understanding Broadcast action, category and data, Sending & Receiving Broadcast	03	

		B	Services: Overview of services in Android, Implementing a Service, Service lifecycle, Inter Process Communication (AIDL Services).	03	
		C	Web Services and WebView : Consuming web services, Receiving HTTP Response (XML, JSON), Parsing JSON and XML, Using WebView	03	
IV	Multithreading	A	Multithreading: Background processing in android	02	
		B	Threads running on UI thread: Handlers & Runnable, AsyncTask	02	
V	Data Storage	A	Introduction to data storage: Shared Preferences: Introduction, Preferences types, operating modes Android File System: Internal storage, External storage.	04	
		B	SQLite: Basics of SQLite Database, Data Types, SQLite Connections, SQLiteOpenHelper class	03	
		C	SQLite Queries: Working with cursors, Inserting, updating, and deleting contents of SQLite	03	
		D	Firestore: Introduction, Firestore services (Authentication, Cloud messaging, Database query)	02	

Reference Books :

1. Dawn Griffiths and David Griffiths, Head First Android Development: A Brain-Friendly Guide, Shroff/O'Reilly; Second edition, 2017.
2. John Horton, **Android Programming for Beginners**, Packt Publishing Limited , 2015.
3. Erik Hellman, **Android Programming: Pushing the Limits (MISL-WILEY)** , Wiley,2013
4. Iyanu Adelekan, Kotlin Programming by Example, Packt Publishing Limited,2018.
5. Valentino Lee, Heather Schneider, and Robbie Schell, Mobile Applications: Architecture, Design, and Development, Prentice Hall, 2004.
6. Rajiv Ramnath, Roger Crawfis, and Paolo Sivilotti, Android SDK 3 for Dummies, Wiley, 2011.

Web References

1. <https://developer.android.com/>
2. <http://www.tutorialspoint.com/android/>
3. <http://www.appinventor.org/>
4. <http://www.smashingmagazine.com/guidelines-for-mobile-web-development/>
5. <https://www.smashingmagazine.com/2017/05/basic-patterns-mobile-navigation/>

Suggested List of Practical :

1). Setup Android Studio Environment

- Install or update Android Studio to its latest version.
- Ensure that your Android app:
 - Targets API level 16 (Jelly Bean) or later
 - Uses Gradle 4.1 or later
- Create Android Studio Environment,
- Explore Android Studio IDE
- Set up a device or emulator for running your app (Android studio's Emulator, Genymotion, Koplayer, Memu)
- Sign into Firebase using your Google account.
- Connect your Android app to Firebase

Testing and Debugging Android Application (Use of Dalvik Debug Monitor Server (DDMS), Use of Step Filters, Breakpoints, Suspend and Resume, use of LogCat (Verbose, Debug, Info, Warn, Error, Assert), Use of Perspectives

2). Create Simple Calculator Application

Layout design with constraint layout, Implement app, Debug and find errors, Installation of .apk into your android mobile

3). Create Tic tac toe Game

Layout design with TableLayout , Add colors, Buttons call event, Find winner, Play with device.

4). Create Zoo app

- ListView layout design, Load listView with Data, Load different views in ListView, ListView events, Add or Remove item to listView

5). Create Restaurant App

Design GridView Layout, Load gridView with list of foods, Show Food details

6). Web services - Find City sunrise time app

HTTP calls, JSON and XML, UI layout design, HTTP calls and JSON read

7). Sqlite database - My notes

Design Add notes layout, Use Menu bar and Intent, Styles, Add notes to Sqlite database, List notes from Sqlite database, Delete notes from Sqlite database, Update notes in Sqlite database

8). Complete Alarm App

• Layout design, Broadcast Receiver and Service concept, Broadcast Receiver and Set alarm time SharedPreferences and start app with OS

9). Use Firebase Realtime Database, to build a mobile system

- User login with phone number, Menus for contact and main activity, Dummy contact list data, Pick contacts from phone. Save my trackers in Shared preferences, Firebase signIn anonymously, Save user info into Firebase, Save my trackers in Realtime- database, People who I find using dummy data, People who I find using Firebase data, Load user's contacts, Send phone location to the server, Find missing phone location
- Services Vs Broadcast receiver
- Service, Run App in background

Note :

Languages/Tools: Java/Kotlin, XML, Android Studio, AVD (Android studio's Emulator, Genymotion, Koplayer, Memu), Firebase may be used for practical.

Course Code : CSC109		Course Title : Full Stack Web Development		Semester : VI	
Theory Marks : 100		Theory Credits : 04		Theory Lectures : 60	
Practical Marks : 50		Practical Credits : 02		Practical Lectures : 60	
Total Marks : 150		Total Credits : 06			
Course prerequisites : - Working knowledge of HTML,CSS, JavaScript					
Course objective:					
<ul style="list-style-type: none"> • To know the core concepts of Node js and React for server side and client side coding . • To gain insight and understand the working of MVC architecture with MERN • To introduce the no-sql database - MongoDB 					
Learning Outcomes : Upon completion of the course students should be able to::					
<ul style="list-style-type: none"> • Explain the significance of each of the MERN components. • Develop a CRUD application using MongoDB. • Develop applications using NODEjs, React to understand the different aspects of these technologies. • Design and implement a full-fledged application using all the components of the MERN Stack 					
Course content:					
Unit		Topic			
#	Title	#	Content	Lectures	
I	JavaScript	A	JavaScript basics: Introduction, Syntax and Statements, Comments, Operators, Variables, Z`Assignment, Loops, If and Switch conditions, break & Continue, Data Types: Number & Number Methods, Strings and String Methods,	03	

			Functions, Callbacks, Arrays, Array Methods, Looping through an array (Array Iteration)		
		B	JavaScript advanced features Let and Const, Let inside loops, String Templates For of Loops, map, reduce, filter, Arrow Functions Class, Class properties and methods, object, this operator, spread function, Class Constructor, Class Inheritance, Modules import and export, Form validation using validation API	06	
		C	AJAX Ajax - request object creation, forwarding the request, accepting response object and display on webpage, JSON syntax, XMLHttpRequest Object	02	
II	Node Js	A	Introduction Advantages of Node JS , Node.js Process Model, Working in REPL, Node JS Console.	02	
		B	Node JS Modules Functions, Buffer, Module, Module Types : Core Modules, Local Modules, Module.Exports	03	
		C	Node Package Manager NPM, Installing packages Locally, Adding dependency in package.json, Installing packages globally, Updating packages.	02	
		D	Web Server Creating Web Server, Handling HTTP Requests, Sending Requests.	03	
		E	File System Fs.readFile, Writing a File, Writing a file asynchronously, Opening a file, Deleting, Other IO Operations.	03	
		F	Events EventEmitter Class, Returning event emitter, Inheriting events	03	

III	React	A	React elements and JSX What is react? advantages and disadvantages Overview of JSX, creating elements with JSX, JSX components, properties in JSX, condition statements in JSX, Rendering an Element into the DOM, Naming Conventions	05	
		B	Components & Component Life Cycle Overview of Components, Props, State, component composability, Life Cycle Methods, Reusable Components	06	
		C	Forms Submitting the form data to server using react component and updating state, Validating Props, Call back events.	06	
IV	Express Js	A	Introduction Introduction to Express, MVC pattern, initial node server setup, adding data to the server(ready data from mockaroo)	02	
		B	Routing create a basic route, add a static route for file server, routing with express : routing parameters, routing handler, routing common methods, routing chaining,	04	
		C	HTTP Interaction: Handling Form Data, Handling Query Parameters, Cookies and Sessions	03	
		D	RESTful Services Creating and Consuming RESTful Services, Using Templates	03	
V	MongoDb		Introduction to MongoDB, Document-oriented database, key features, databases and collections , CRUD operations, Text search operations	04	

Reference Books :

1. Basarat Ali Syed , Beginning Nodejs, Apress
2. Vasan Subramanian , **Pro MERN Stack Full Stack Web App Development with Mongo, Express,**

React, and Node, Appress, 1st Edition

3. Eddy Wilson, Iriarte Koroliova, **MERN Quick Start Guide: Build web applications with MongoDB, Express.js, React, and Node**, Packt

Web References

1. <https://docs.mongodb.com>
2. <https://medium.com/poka-techblog/>
3. <https://javascript.info/>
4. <https://reactjs.org/>

Suggested List of Practical :

1. JavaScript
 - a) Demonstration of const, let, string templates, callbacks, arrow functions, class, class-properties, methods.
 - b) Handling asynchronous request using AJAX and JSON with a simple server script that just returns JSON data.
2. Nodejs - Installation and Nodejs Core
 - a) Installing Nodejs
 - b) Use of global object
 - c) Argument variables with process.argv
 - d) Standard input and standard output
3. Nodejs Modules
 - a) Core Modules
 - b) Collecting information with Readline
 - c) Handling events with EventEmitter
 - d) Exporting custom modules
 - e) Creating child process with exec/spawn
4. The File system
 - a) Listing directory files
 - b) Reading files
 - c) writing and appending files
 - d) Directory creation
 - e) Readable file streams
 - f) Writable file streams
5. The Http Module

- a) Making a request
 - b) Building a web server
 - c) serving files
 - d) serving JSON data
 - e) Collecting POST data
6. The Node Package manager
- a) Installing node NPM
 - b) Initializing a package-json file
 - c) Adding node packages
 - d) Managing global directory
 - e) Updating a package
 - f) Removing a package
 - g) File servers with httpster
7. Introduction to React Elements
- a) Installing create-react-app
 - b) Generate a project
 - c) Create react elements
 - d) Refactor elements using JSX
8. React components
- a) Create a react component
 - b) Add component properties
 - c) Creating a component(ex Book) with data
 - d) Adding custom methods
 - e) creating function components.
9. Props and states
- a) Compose components
 - b) Display child components
 - c) Introducing state
 - d) Using setstate
 - e) pass state as props
 - f) Conditional rendering
10. Additional react features
- a) Component life cycle
 - b) Fetching and rendering JSON data

c)Using forms with react.

11.Mongodb

a) installation

b)collection and basic operations(find, create, update ,delete, drop)

c)Reading and writing to MongoDB database using APIs

12. Developing a simple CRUD application using the MERN stack.

Note :

The Tools visual Studio code/sublime may be used for practical

Course Code : CSC110	Course Title : Internet of Things	Semester : VI
Theory Marks : 100	Theory Credits : 04	Theory Lectures : 60
Practical Marks : 50	Practical Credits : 02	Practical Lectures : 60
Total Marks : 150	Total Credits : 06	
Course prerequisites : - Basic programming knowledge		
Course objective : <ul style="list-style-type: none">• To Introduce concepts for internet of things and the different devices involved in IOT.• To introduce cloud concepts and its use in IOT• Gain hands on experience of working with different sensors/actuators and their use in IOT projects.• To gain knowledge of Arduino, NodeMcu, Raspberry pi Boards and to develop IOT projects by integrating these boards with a cloud platform.		
Learning Outcomes : Upon completion of the course students should be able to : <ul style="list-style-type: none">• Explain the requirements and components of an IOT system.• Develop different IOT projects using cloud technology• Develop IOT Projects using the Arduino, NodeMcu, Raspberry pi Boards and a cloud platform such as Nodered or similar.		
Course content:		
Unit	Topic	

#	Title	#	Content	Lectures
I	IOT Concepts	A	Introduction Definition, modern day IoT applications, Baseline technologies-M2M,WoT, IOT categories- industrial and consumer, IOT components	03
		B	Sensors and Actuators sensors, transducers, sensor features, resolution, analog sensors, digital sensors, scalar sensors, vector sensors, sensor types. Actuators-types-hydraulic, pneumatic, electrical, thermal/mechanical ,motors-DC, Servo, Stepper, relays, motor drivers for interfacing	06
		C	IOT Networks IoTWF Standardized Architecture, Connectivity Protocols- MQTT,SMQTT; communication protocols- IEEE 802.15.4,802.11,LORA wireless protocol ,ZigBee	06
II	IOT Boards	A	Arduino Introduction to Arduino Programming-features of arduino, Arduino IDE, sketch, sketch structure, supported data types, Arduino function libraries, operators, control statements, arrays, String functions, Interrupts, sensor interface with Arduino, DHT sensor library, types of motor actuators, Arduino servo library	10
		B	Raspberry Pi and comparative study Introduction to Raspberry Pi – specifications, GPIOs, Features of EsP8266 ,comparative studies of Arduino uno, raspberry pi, nodemcu boards and their applications	06
III	Cloud Technology	A	Introduction to cloud computing definition, characteristics, components, service models-IaaS, Pass, SaaS, Deployment models-public, private, hybrid, open source and commercial clouds-examples, facilities offered	10
		B	Cloud computing case studies Microsoft Azure-features, Azure as PaaS, Azure as IaaS; OPenStack-components and features, Firebase cloud service features	06
		C	Visual tool for wiring IOT NodeRed, its features, installing on Raspberry pi	04
		D	Wireless sensor networks definition, limitations; Sensor cloud-definition, difference with WSN, Actors in sensor cloud, architecture	05
		E	Fog computing Introduction, why use fog computing, when to use fog	04

		computing, architecture of fog, fog nodes, working of fog, applications of fog	
Reference Books :			
<ol style="list-style-type: none"> 1. Arshdeep Bagha, Vijay Madiseti , Internet of Things: A Hands-On Approach, Orient Blackswan Private Limited 2. Adrian Mcewen , Designing The Internet of Things, Wiley 3. Hanes David,Salguiero Gonzalo , IoT Fundamentals: Networking Technologies, Protocols and Use Cases for the Internet of Things , Pearson Education 			
Suggested List of Practical :			
<ol style="list-style-type: none"> 1. Blink an LED . Traffic lights using Arduino board. 2. Night security light using PIR motion sensor and photo resistor. 3. Arduino weather station with temperature, humidity, pressure date and time. 4. Controlling a server motor using arduino. 5. Setting up Raspberry pi and blinking LED 6. Capturing an image using Raspberry Pi 7. DHT22 interfaced with Raspberry Pi to record the temperature. 8. Setting up server on Raspberry Pi and sending the recorded temperature to the server. 9. Installing NOdeRed, creating and testing a simple flow in NodeRed. 10. Controlling an LED with NodeRed. 11. Use of Digital Smoke and gas sensor to detect gas/smoke with ESP8266, MQTT and NodeRed. 12. Controlling lamps and outlets using Arduino and MQTT 			
Note :			
Tools like Arduino IDE, python editor may be used			

Discipline Specific Electives

Course Code : CSD101	Course Title : Human Computer Interaction	Semester : V
Theory Marks : 75	Theory Credits : 03	Theory Lectures : 45
Practical Marks : 25	Practical Credits : 01	Practical Lectures : 30
Total Marks : 100	Total Credits : 04	
Course objectives:		
<ul style="list-style-type: none"> • To introduce the foundations of Human Computer Interaction, design technologies and user interface design and development. 		

- Learn the foundations of Human Computer Interaction
- Be familiar with the design technologies for individuals and persons with disabilities
- Learn the guidelines for user interface design and development
- Be aware of mobile HCI

Course Outcomes : Upon completion of the course students should be able to::

- Develop meaningful user interface
- Assess the importance of user feedback
- Design effective HCI for individuals and persons with disabilities
- Develop persona, conduct interview
- Develop storyboard and design prototype
- Design GUI, Web UI and Reports.
- Perform Heuristic Evaluation of the design

Course content:

Unit		Topic		
#	Title	#	Content	Lectures
I	FOUNDATIONS OF HCI		The Human: I/O channels, Memory, Reasoning and problem solving; The computer: Devices, Memory, processing and networks; Interaction: Models, frameworks, Ergonomics, styles, elements, interactivity, Paradigms	8
II	DESIGN – RULES AND TECHNIQUES		Interactive Design basics: process, scenarios, navigation, screen design, Iteration and prototyping. Usability engineering, Prototyping in practice, design rationale. Design rules: principles, standards, guidelines, rules. Evaluation Techniques, Universal Design.	8
III	MODELS AND THEORIES		Cognitive models, Socio-Organizational issues and stake holder requirements; Communication and collaboration models-Hypertext, Multimedia and WWW	8
IV	MOBILE HCI		Mobile Ecosystem: Platforms, Application frameworks, Types of Mobile Applications: Widgets, Applications, Games; Mobile Information Architecture, Mobile 2.0, Mobile Design: Elements of Mobile Design, Tools.	8
V	WEB INTERFACE DESIGN		Designing Web Interfaces: Drag & Drop, Direct Selection, Contextual Tools, Overlays, Inlays and Virtual Pages, Process Flow	8
VI	CONTEMPORARY INTERFACE DESIGN TECHNOLOGY		Future Domains, IHCI and Case Studies	5

Reference Books :

1. Alan Dix, Janet Finlay, Gregory Abowd, Russell Beale; Human Computer Interaction; Pearson Education, 2004 (UNIT I,II and III), 3rd Edition.
2. Brian Fling; Mobile Design and Development , OReilly Media Inc., 2009 (UNIT –IV)
3. Bill Scott and Theresa Neil ; Designing Web Interfaces; OReilly, 2009 (UNIT V), First Edition

Suggested list of practical :

1. Paper Prototyping using templates
2. Story boarding
3. Conducting survey interview and summarizing the result
4. Persona- conducting contextual interview and developing persona
5. GUI design- form design, menu design, help, error messages
6. Web UI design- pages, navigation, controls, (Ajax)
7. Report designs
8. Heuristic evaluation

Course Code : CSD104		Course Title : Embedded Systems		Semester : V	
Theory Marks : 75		Theory Credits : 03		Theory Lectures : 45	
Practical Marks : 25		Practical Credits : 01		Practical Lectures : 30	
Total Marks : 100		Total Credits : 04			
Course prerequisites :					
Course objectives:					
<ul style="list-style-type: none"> • Conceptualize the basics of Embedded systems • Understand fundamentals of Real Time Operating Systems 					
Learning Outcomes :					
<ul style="list-style-type: none"> • Explain the function and use of embedded system hardware and Interfacing I/O devices. • Identify various sensors, actuators and their use 					
Course content:					
Unit		Topic			
#	Title	#	Content	Lectures	

I	Introduction	Introduction to embedded systems, Microprocessors and Microcontrollers, Components of Embedded system & its classification, characteristics of embedded system.	5L
II	Interrupts	Microprocessor Architecture, Interrupt Basics, shared Data problem, Interrupt latency	8L
III	Survey of Software Architecture	Round Robin, Round Robin with interrupt, Function-Queue-Scheduling Architecture, Real time OS Architecture	8L
IV	Introduction to RTOS System	Task and Task states, Task and Data, Semaphores and shared Data	6L
V	More OS services	Message Queues, Mailboxes and pipes, Timer functions, Events, Memory Management, Interrupt routines in RTOS Environment	8L
VI	Embedded software Development Tools	Host and Target machines, Linkers/Locators for Embedded Systems, Getting Embedded software into the Target system	10L

Text Book:

1). David E Simon, “An Embedded Software Primer”, Pearson India, 1st Edition

References

1. [Tony Givargis Frank Vahid](#) ; Embedded System Design: A Unified Hardware / Software Introduction, Wiley; Student edition
2. D. Patranabis, “Sensors and Transducers”, PHI learning Private Limited

Suggested list of practical :

1. Interfacing sensors
2. Interfacing output devices
3. Interfacing input devices
4. Interfacing actuators
5. Programming with Raspberry Pi
6. Blink an LED, Traffic lights using Arduino Board
7. Monitoring Data over Cloud
8. Building Web app to control devices
9. A mini Project

Note :

Programs to be executed on some of the Embedded boards like Arduino, Intel Edison, Raspberry Pi, Bolt, etc

that covers the above tasks.

Course Code : CSD107	Course Title : Introduction to Data Analytics	Semester : VI		
Theory Marks : 75	Theory Credits : 03	Theory Lectures : 45		
Practical Marks : 25	Practical Credits : 01	Practical Lectures : 30		
Total Marks : 100	Total Credits : 04			
Course prerequisites: Students are expected to have basic knowledge of algorithms and reasonable programming experience and some familiarity with basic linear algebra and basic probability and statistics.				
Course objectives: Become familiar with methods of Data Science and their practical usefulness				
Learning Outcomes : Upon completion of the course students should be able to :				
<ul style="list-style-type: none"> • Describe what Data Science is and the skill needed to be a data analyst. • Explain in basic terms what statistical inference means. Identify probability distributions commonly used as foundations for statistical modeling. Fit a model to data. • Use Python to carry out basic statistical modeling and analysis. 				
Course content:				
Unit		Topic		
#	Title	#	Content	Lectures
I	Introduction to Data Science		Data Science, Big Data, significance of data science, Datafication, Current landscape of perspectives, Skill sets needed , The Data Science Process	4
II	Statistical Inference		Populations and samples, statistical modelling, probability distributions, fitting a model	6

III	Exploratory Data Analysis and the Data Science Process		Basic tools(plots, graphs and summary statistics) of Exploratory Data Analysis, Philosophy of EDA	6	
IV	Feature Generation and Feature Selection (Extracting Meaning From Data)	A	Motivating application: user (customer) retention, Feature Generation (brainstorming, role of domain expertise, and place for imagination) , Feature Selection algorithms	5	
V	Basic Machine Learning Algorithms		Classification and Clustering algorithms, Linear Regression, Logistic regression, k-Nearest Neighbors (k-NN), k-means , Decision Trees, Random Forests	10	
VI	Mining Social-Network Graphs		Social networks as graphs, Clustering of graphs, Direct discovery of communities in graphs, Partitioning of graphs, Neighborhood properties in graphs	6	
VII	Data Visualization		Basic principles, ideas and tools for data visualization, Examples of inspiring (industry) projects,	4	
VIII	Data Science and Ethical Issues		Discussions on privacy, security, ethics, A look back at Data Science, Next-generation data scientists	4	

Text Books:

1. Cathy O’Neil and Rachel Schutt.; “Doing Data Science, Straight Talk From The Frontline”, O’Reilly, 2014.

Reference Books :

1. Jure Leskovek, Anand Rajaraman and Jeffrey Ullman, Mining of Massive Datasets v2.1, Cambridge University Press, 2014 (free online)
2. Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, 2013.
3. Foster Provost and Tom Fawcett, Data Science for Business: What You Need to Know about Data Mining and Data-analytic Thinking, 2013.
4. Trevor Hastie, Robert Tibshirani and Jerome Friedman. Elements of Statistical Learning, Second Edition, 2009 (free online)
5. Avrim Blum, John Hopcroft and Ravindran Kannan. Foundations of Data Science.
6. Mohammed J. Zaki and Wagner Miera Jr, Data Mining and Analysis: Fundamental Concepts and Algorithms, Cambridge University Press, 2014.

Suggested list of practical :

1. Predict the class of a flower based on the available attributes for Iris data set.
2. Dream Housing Finance company deals in all home loans. They have presence across all urban, semi urban and rural areas. Customer first apply for home loan after that company validates the customer eligibility for loan. Company wants to automate the loan eligibility process (real time) based on customer detail provided while filling online application form. These details are Gender, Marital Status, Education, Number of Dependents, Income, Loan Amount, Credit History and others. To automate this process, they have given a problem to identify the customers segments, those are eligible for loan amount so that they can specifically target these customers.
3. Refer to the Wine Quality data set on the following link: <https://archive.ics.uci.edu/ml/machine-learning-databases/wine-quality/> Predict the quality of wine.

Note : All programs to be implemented using Python