B.Sc. Computer Science

Syllabus

(For students admitted from 2018-2019 onwards)

Department of Computer Science
Bishop Heber College (Autonomous)
Nationally Re-accredited at the 'A' by NAAC with a CGPA of 3.58 out of 4
Recognized by UGC as " College of Excellence"
Tiruchirappalli 620017

Bishop Heber College (Autonomous) Tiruchirappalli – 620 017

B.Sc., Computer Science

(Applicable to Candidates admitted From the Academic Year (2018-2019) onwards

Sem	Part	Course	Course Title	Course Code	Hours / Week	Credits	Marks		
							CIA	ESE	Total
I	I	Tamil I	செய்யுள், இலக்கிய வரலாறு, உரைநடை, மொழிப்பயிற்சியும் படைப்பாக்கமும்	U18TM1L1	6	3	25	75	100
	II	English I	English Communication Skills - I	U16EG1L1	6	3	40	60	100
	III	Core I	Programming Principles and Abstractions using C	U18CS101	6	5	25	75	100
		Core Practical I	Programming Principles and Abstractions using C Lab	U18CS1P1	3	2	40	60	100
		Allied I	Operation Research	U14MAZ11	5	4	25	75	100
		Env. Studies	Environmental Studies	U16EST11	2	2	25	75	100
	IV	Val.Edu.	Value Education (VLO - RI / MI)	U15VL1:1/ U15VL1:2	2	2	25	75	100
II	I	Tamil II	செய்யுள், இலக்கிய வரலாறு, சிறுகதைத்தரட்டு, மொழிப்பயிற்சி ரு	U18TM2L2	6	3	25	75	100
	II	English II	English Communication Skills - II	U16EG2L2	6	3	40	60	100
	III	Core II	Exploring Data Using Python	U18CS202	5	5	25	75	100
		Core Practical II	Exploring Data Using Python Lab	U18CS2P2	3	2	40	60	100
		Allied II	Numerical Methods	U16MAZ22	4	4	25	75	100
		Allied III	Probability and Statistics	U16MAZ23	4	4	25	75	100
	IV	SBEC I	Business Communication and Collaboration Tools	U18CSPS1	2	2	40	60	100
III	I	Tamil III	செய்யுள், காப்பியங்கள், இலக்கிய வரலாறு, நாவல் மொழிப்பயிற்சி	U18TM3L3	6	3	25	75	100
	II	English III	English for Competitive Examinations	U16EG3L3	6	3	40	60	100
	III	Core III	Object Oriented System Design	U18CS303	6	5	25	75	100
		Core Practical III	Object Oriented System Design Lab	U18CS3P3	3	2	40	60	100
		Applied I	Electricity, Magnetism and Electromagnetism	U13PHZ34	4	3	25	75	100
		Applied Practical I	Applied Physics Practicals	U13PHZP1	3	---	---	---	---
	IV	NMEC I	Basics of Computer Programming	U18CS3E1	2	2	25	75	100

Sem	Part	Course	Course Title	Course Code	Hours / Week	Credits	Marks		
							CIA	ESE	Total
IV	I	Tamil IV	செய்யுள், நாடகம், இலக்கிய வரலாறு, மொழிப்பயிற்சி	U18TM4L4	5	3	25	75	100
	II	English IV	English through Literature	U16EG4L4	5	3	40	60	100
	III	Core IV	Database Design	U18CS404	6	5	25	75	100
		Core Practical IV	Database Design Lab	U18CS4P4	3	2	40	60	100
		Applied II	Solid State Devices and Microprocessor	U13PHZ45	4	4	25	75	100
		Applied Practic II	Applied Physics Practicals	U13PHZP1	3	3	40	60	100
	IV	NMEC II	Fundamentals of Information Technology	U18CS4E2	2	2	25	75	100
		Soft Skills	Life Skills	U16LFS41	2	1	---	---	100
	V	Extension Activities	NSS, NCC, Rotract, Lioclub, etc…,	U16ETA41	---	1	---	---	---
V	III	Core V	Database – Driven Web Design	U18CS505	5	5	25	75	100
		Core VI	Principles of Operating Systems	U18CS506	5	5	25	75	100
		Core VII	Computer Graphics	U18CS507	5	5	25	75	100
		Core Practical V	Database – Driven Web Design Lab	U18CS5P5	6	4	40	60	100
		Elective I	1a) Software Engineering	U18CS5:1	5	5	25	75	100
			1b) Internet of Things	U18CS5:2					
			1c) Cloud Computing	U18CS5:3					
	IV	SBEC II	Project Proposal	U18CSPS2	2	2	40	60	100
		SBEC III	Technical Communication for Computer Scientists	U18CSPS3	2	2	40	60	100
VI	III	Core VIII	Computer Networking	U18CS608	6	5	25	75	100
		Elective II	2a) Multimedia Technologies	U18CS6:1	6	5	25	75	100
			2b) Web Applications Development	U18CS6:2					
			2c) XML Basics	U18CS6:3					
		Core Practical VI	Digital Electronics and Microprocessors Lab	U18CS6P6	6	5	40	60	100
		Elective III	3a) Business Analytics	U18CS6:4	6	5	25	75	100
			3b) Foundations of Enterprise Computing	U18CS6:5					
			3c) Mobile Application Development	U18CS6:6					
		Core / Project	Project Implementation	U18CS6PJ	6	5	---	---	100
	V	Gender Studies	Gender Studies	U16GST61	---	1	---	---	100
Total Credits						**140**			

Programming Principles and Abstractions using C

Course Objectives

This course introduces students basic concepts, syntax and control structures in programming. Students will learn how to develop applications in a step-wise problem solving fashion. They will also learn various data structures and select an appropriate data structure for the chosen problem.

Learning Outcomes

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

- Declare variables, constants, arrays, pointers, structures and unions.
- Initialize variables, arrays, pointers, structures with values.
- Manipulate values of variables, arrays, pointers, structures and files.
- Define functions that can receive variables, arrays, pointers and structures and return values.
- Create, open, read, manipulate, write and close files.
- Select and use appropriate data structures for the given problem.

Unit I: Basics of C and Decisions

Data types, Variables, Assignment statements – Printf, puts, scanf statements, Format specifiers, Escape specifications – Arithmetic operators, Arithmetic expressions, Order of precedence, Evaluation of arithmetic expressions – Logical, relational and bitwise operators – Decision statements: if, if-else, if-else-if and nested if statements – Switch statement, break statement – Constants , C keywords, C character set , Type conversions – Switch statement.

Unit II: Loops, Arrays and Functions

For loop, Continue statement – While loop – Do-while loop – 1-D Arrays: Definition, initialization and Array index –2-D Arrays: Definition, initialization and Array index – Multi-dimensional arrays – Functions: definition, arguments, return statement and calling – Local variables and global variables. Functions with arrays – Functions with decision and looping statements. – Recursion.

Unit III : Pointers, Structures and Union

Strings processing: Standard string library functions – Pointers: declaration, assigning to pointer variable, using a pointer, printing pointer, null pointer – Arithmetic operation with pointers – Pointers and arrays, pointers and strings – Pointers and functions, variable length arrays – Array of pointers, Storage class – Structures: declaration, initialization, array of structures – Pointer to structures, Structures and functions – Typedef, Enumerated datatypes, Unions.

Unit IV : **File Processing**

Files: introduction and files functions – Writing and reading in Text mode – Simple application: Display the contents of a file. Write data to a file. Append data to existing file – Writing and reading in Binary mode – Simple application: Display the contents of a file. Write data to a file. Append data to existing file – File IO – Reading and writing structures – Command line arguments.

Unit V : Data Structures

Stack: LIFO concept, Stack operations, Array implementation of stack – Queue: FIFO concept, Queue operations, Array implementation of queue – Singly Linkedlist: concepts, operations – Doubly Linkedlist: concepts, operations – Trees: General trees, Binary trees – Tree traversal algorithms: inorder, preorder and postorder – Binary search trees: concepts, operations – Graphs: concepts, graph implementations using adjacency matrix and adjacency list – Graph traversals: Breadth first search – Graph traversals: Depth first search.

Text Book(s):

1. Ashok N Kamthane, ***"Programming and Data Structures"***, Pearson, 2009.
2. E.Horowitz, S.Sahni and Susan Anderson Freed, ***"Fundamental Data Structures in C",*** 2ed, Orient BlackSwan Publisher, 2009.

References Book(s):

1. Steve Oualline, ***"Practical C Programming",*** 3ed, Oreilly Publishers, 1997.
2. Reema Thareja, ***"Data Structures using C",*** Oxford University Press, 2014.

Programming Principles and Abstractions using C Lab

Course Objectives

This lab course teaches students the art of programming using elementary algorithms and data structures. Students will learn how to identify algorithmic steps, develop code using C language, debug errors and test the code with various inputs.

Learning Outcomes

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

- Declare variables, constants, arrays, pointers, structures and files
- Initialize variables, arrays, pointers, structures with values
- Manipulate values of variables, arrays, pointers, structures and files
- Define functions that can receive variables, arrays, pointers and structures and return values
- Create, open, read, manipulate, write and close files
- Select and use appropriate data structures for the given problem

1. [Starter Applications]: Read and print name, Convert temperature from F to C and vice versa, Find biggest of 3 numbers, Compute area of shapes: rectangle, circle and cylinder.
2. [Sorting student marks]: Sort the given marks of students in ascending order and descending order
3. [Arithmetic Calculator]: Develop menu-driven arithmetic calculator application using functions. Each function should receive values and return the calculated output.
4. [Sorting a stack]: Given a stack, sort it such that the top of the stack has the largest element. Read the size of the stack, followed by the numbers to be pushed to stack. The output should be the popped elements from the sorted stack.
5. [Car washing queue]: A car washing garage has 2 gates one to enter another to exit and can wash 3 cars at a time. Cars can enter the garage in one gate and exit in another gate. For this application, ask user which operation one wants such as '1-insert car', '2-remove car', '3-show queue' or '4-quit'. Perform the following sequence of operations: 1 1; 1 2; 1 3; 2; 3; 1 4; 3; 2; 1 5; 1 6; 2; 3; 4
6. [Binary search tree of filenames]: Given a set of file names (such as hello.c), create a binary search tree. Read 10 file names and display the names based on inorder traversal.
7. [Detecting cycles in a graph]: Given a directed graph, your task is to detect if there is a cycle in the graph or not. Read two integers 'N' and 'M' which denotes the no of vertices and no of edges respectively. Read 'M' pairs u and v denoting that there is an uni-directed edge from u to v. Print if there is a cycle or not.
8. [Intersection of two linked lists]: Given two linked lists, your task is to complete the function findIntersection(), that returns the intersection (ie., common elements) of two linked lists. The function takes 2 arguments, reference pointer to the head of the first linked list (head1) and reference pointer to the head of the second linked list (head2). The function should return reference pointer to the head of the new list that is formed by the intersection of the two the lists.
9. [Sorting employees]: Each employee record contains name and salary. Read name and salary of 10 employees into structure. Sort the records based on salary, if same then sort on name
10. [Mark List Generation]: Read a file that contains details of 5 students. Each record contains student name, mark1, mark2 and mark3. Prepare a mark list that displays student name, marks and average mark of each student.

Exploring Data Using Python

Course Objectives

This course introduces Python as a language for solving data analysis problems instead of a language to understand any algorithms or abstractions. The main emphasis is given to data-oriented examples and exercises and not just number-oriented exercises, exploiting real world use cases and simple examples using Python.

Learning Outcomes

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

- Apply language features including strings, lists, tuples, dictionaries, regular expressions and files.
- Develop classes using OO features.
- Import other packages such as urllib to develop applications.

Unit I: Python Basics and Functions

Variables – Operators – Statements – Getting Inputs – Boolean conditions – Alternative, chained and nested conditions – Catching exceptions – Function calls – Built-in functions – Type conversion functions and math functions – Creating new functions, parameters and arguments – Need for functions.

Unit II: Loops and Strings

While statement – Infinite loops – Continue statement – For loops – Counting and summing loops – Maximum and minimum loops – Traversal through strings – String slice – Looping and counting in strings – The in operator – String comparison – String methods – parsing strings – Format operator.

Unit III: Files and Lists

Opening files – Text files – Reading files – Searching through files – Selecting files names from user – Writing files – Traversing list – List operations – List slice – List methods – Deleting elements – Built-in list functions – Objects, value and aliasing – List arguments.

Unit IV: Dictionaries, Tuples and OOP

Dictionaries – Files and dictionaries – Looping and dictionaries – Advanced text processing – Tuples – Comparing tuples – Tuple assignments – Dictionaries and tuples – Tuples as keys in dictionaries – Creating objects – Encapsulation – Classes as types – Object lifecycle – Instances – Inheritance.

Unit V: Internet Programming

Regular expressions – Character matching – Extracting data – Escape character – Designing simple web browser using sockets – Retrieving images using HTTP – Retrieving web pages using urllib – Reading binary files using urllib.

Text Book(s):

1. Charles R. Severance, Python for Everybody: "***Exploring data using Python 3"***, Schroff Publishers, 1ed, 2017, ISBN 978-9352136278.

References Book(s):

1. Allen Downey, Think Python: "***How to think like a computer scientist"***, Schroff / O'Reilly Publishers, 2ed, 2016, ISBN 978-9352134755.

SEMESTER: II **HOURS/WEEK : 3**
Code: U18CS2P2 **CREDITS : 2**

Exploring Data Using Python Lab

Course Objectives

This lab course aims to provide students programming experience in Python to solve data analysis problems. The main emphasis is given to data-oriented examples and exercises utilizing real world use cases.

Learning Outcomes

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

- Develop simple applications using Python interpreter and Jupyter Notebooks and Python editors such as JEdit.
- Use built-in functions and to develop their own functions.
- Create and import files, split file contents and process tokens also using regular expressions.
- Design classes OO features and import other packages such as urllib.

I. Simple Applications

1). Practice using the Python interpreter as a calculator:

A). The volume of a sphere with radius r is $4/3\pi r^3$. What is the volume of a sphere with radius 5?

B). Suppose, cover price of a book is INR 24.95, but a bookstore offers a 40% discount. Shipping costs INR 3.00 for the first copy and 75 Paise for each additional copy. What is the total wholesale cost for 60 copies?

C). If I leave my house at 6:52 am and run 1 km at an easy pace (8:15 per km), then 3 km at tempo (7:12 per km) and 1 km at easy pace again, what time do I get home for breakfast?

2). Develop an application to prompt for a score between 0.0 and 1.0. If the score is out of range, print an error message. If the score is between 0.0 and 1.0, print a grade using the following table:

Score	Grade	Score	Grade	Score	Grade	Score	Grade	Score	Grade
>= 0.9	A	>= 0.8	B	>= 0.7	C	>= 0.6	D	< 0.6	F

3). Develop an application that repeatedly reads numbers until the user enters "done". Once "done" is entered, print out the total, count, and average of the numbers. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number.

4). The **datetime** module provides time objects that are similar to the Time objects, but they provide a rich set of methods and operators. Write a program that takes a birthday as input and prints the user's age and the number of days, hours, minutes and seconds until their next birthday.

5). 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 chars of the string are the same.

6). Develop a function front_x(). Given a list of strings, return a list with the strings in sorted order, except group all the strings that begin with 'x' first. Eg. ['mix', 'xyz', 'apple', 'xanadu', 'aardvark'] yields ['xanadu', 'xyz', 'aardvark', 'apple', 'mix']. Hint: this can be done by making 2 lists and sorting each of them before combining them.

7). Develop a function sort_last(). Given a list of non-empty tuples, return a list sorted in increasing order by the last element in each tuple. E.g. [(1, 7), (1, 3), (3, 4, 5), (2, 2)] yields [(2, 2), (1, 3), (3, 4, 5), (1, 7)]. Hint: use a custom key= function to extract the last element form each tuple.

8). Develop a function remove_adjacent(). Given a list of numbers, return a list where all adjacent same elements have been reduced to a single element, so [1, 2, 2, 3] returns [1, 2, 3]. List [2, 2, 3, 3, 3] returns [2, 3]. List [ ] returns [ ]. You may create a new list or modify the passed in list.

9). Write a function verbing(). Given a string, if its length is at least 3, add 'ing' to its end. Unless it already ends in 'ing', in which case add 'ly' instead. If the string length is less than 3, leave it unchanged. Return the resulting string. So ‘hail’ yields: hailing; ‘swimming’ yields: swimmingly; ‘do’ yields: do.

10). Develop a function not_bad(). Given a string, find the first appearance of the substring 'not' and 'bad'. If the 'bad' follows the 'not', replace the whole 'not'...'bad' substring with 'good'. Return the resulting string. So 'This dinner is not that bad!' yields: This dinner is good!

II. Advanced Applications

11) Develop an application that prompts the user for a list of numbers and prints out the maximum and minimum of the numbers at the end when the user enters “done”. If the user enters anything other than a number, detect their mistake using try and except and print an error message and skip to the next number. The program should store the numbers the user enters in a list and use the max() and min() functions to compute the maximum and minimum numbers after the loop completes.

12) Develop an application that reads a file and prints the *letters* in decreasing order of frequency. Your program should convert all the input to lower case and only count the letters a-z. Your program should not count spaces, digits, punctuation, or anything other than the letters a-z. Also, print the top N (say N=10) words.

13) Develop an application to open the file (say, romeo.txt) and read it line by line. For each line, split the line into words using the *split* function. For each word, check to see if the word is already in a list. If the word is not in the list, add it to the list. When the program completes, sort and print the resulting words in alphabetical order.

14). Develop an application to read through the email data and when you find line that starts with “From”, you will split the line into words using the split function. We are interested in who sent the message, which is the second word on the From line.

From stephen.marquard@uct.ac.za Sat Jan 5 09:14:16 2008

You will parse the From line and print out the second word for each From line, then you will also count the number of From (not From:) lines and print out a count at the end.

15) Develop an application to look for lines of the form, *(say, “New Revision: 39772”)* and extract the number from each of the lines using a regular expression and the findall() method.
Compute the average of the numbers and print out the average.

SEMESTER: II **HOURS/WEEK : 2**
Code: U18CSPS1 **CREDITS : 2**

SBEC – I Business Communication And Collaboration Tools

Course Objectives

This course involves the use of MS Word and MS Excel to perform various formatting tasks and calculations.

Learning Outcomes

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

- Apply the various formatting techniques to textual data
- Create and design tables
- Apply the mail merge concept
- Create charts for the given data

1. Prepare a Bio-Data to apply for job in IT Company using the text manipulation techniques like font style, font size, bullets, footers and headers.
2. Create a mark sheet for 10 students containing the following fields: name, roll no., mark1, mark2 and mark3 and total.
3. Design an effective Class Time Table.
4. Create mathematical formulae.
5. Write a letter to invite your friends for a get together using Mail Merge option.
6. Tabulate a list of 10 students and grade them based on their seminar performance and then sort them according to their grades
7. Prepare a payroll for 10 employees of a company using the following details.
    - If the basic salary is between 15000 and 20000, then 20% of basic salary is given to HRA, 25% to DA, 12% to LIC and 15% to PF.
    - If the basic salary is less than 15000, then 25% is given to HRA, 30% to DA, 11% to LIC and 16% to PF.
    - If the basic salary is more than 20000, then 15 % is given to HRA, 40% to DA, 14% to LIC and 15% to PF Compute the GrossPay, Deductions and NetPay.
8. Prepare a Electricity Bill for the following table:

	1-500 Units	501-800 Units	>800 Units
House	6	8	11.50
Business Enterprise	9	12	17.50
Education Institution	7	10	13.25

Compute the amount to be paid to the Electricity Board.

9. Prepare a Bar chart for analyzing the admission of students in your college for the past five years.
10. Prepare a pie chart for scheduling your daily activities.

SEMESTER: III **HOURS/WEEK : 6**

Object Oriented System Design

Course Objectives

This course enables students to develop a solid foundation in a modern programming language, concepts of programming such as variables, data types, functions, I/O, parameter-passing, assignment, classes, arrays, conditionals, loops and Object Oriented Programming (OOP) using Java. Students will understand techniques of analysis, design, testing, documentation, coding, and debugging.

Learning Outcomes

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

- Declare, initialize and process variables, constants and arrays
- Read and print values from keyboard using Scanner and Dialog boxes
- Create statements for decisions and loops
- Define functions and return values
- Create classes, objects and constructors
- Understand and apply OO design concepts
- Create, open, manipulate and close files using Streams
- Create applets for drawing shapes and playing audio clips

Unit I : Java Fundamentals and decisions

Parts of Java program, println and print methods, Escape sequences - Variables, literals and primitive data types - Arithmetic operators, expressions, precedence and evaluation of expressions - Type conversion, Math class, String class and comment statements - Reading keyboard input: Scanner class and its methods - Reading keyboard input: Message dialog box - Reading keyboard input: Input dialog box - Converting String inputs to numbers - Relational operators, if, if-else, if-else-if statements - Logical operators and its precedence, String comparison and Conditional operator.

Unit II : Multi point selection and Loops

Switch statement, String cases for switch statement - Arithmetic calculator application - Format specifications using printf method, Increment and decrement operators - While loop, indefinite loop, while loop for input validation - Do-while loop - For loop, variable declaration in for loop, multiple statements in initialization - Running totals and sentinels - Nested loops - Break and continue statement, Deciding which loop to use - Exception handling using try-catch-finally statements.

Unit III : Methods and Arrays

Method Definition: Void and value-returning methods, Calling a method, Method calling another method - Passing single argument to a method - Passing multiple arguments to a method, Passing object references to a method, Local variables - Defining and calling value returning method - Returning Boolean value, Returning a reference to an object - Problem solving with methods : Divide and conquer technique - Arrays: declaration, accessing elements, inputting and outputting array contents, Bounds checking, Array initialization - Processing array elements: Array length, Enhanced For loop, Copying arrays - Passing arrays as arguments, Returning arrays from methods, Array of objects - 2D arrays: Initialization, Displaying elements, Summing rows and columns, Passing 2D arrays to methods, Multi-dimensional arrays.

Unit IV : Object Oriented Programming using Java

Objects, classes, Writing a simple class and its methods, Showing access specification in UML diagram - Instance variables and methods, Constructors, default constructor - Overloading methods and constructors - Static variables, Static methods - Passing objects as arguments to methods, Returning objects from methods – Aggregation - Inheritance: Generalization & specialization, IS-A relationship - Calling superclass constructor, Overriding superclass methods, Protected members - Abstract classes and abstract methods – Interfaces.

Unit V : Data structures, Files and Applets
Stack - String Tokenizer, Hashtable - Writing data to file using PrintWriter class - Reading text and primitive data from file using Scanner's nextLine method - FileOuputStream and FileInputStream classes - FileWriter and FileReader classes - Creating and running Applets - Drawing Lines, Rectangles and Ovals in Applets - Drawing Arcs and Polygons in Applets - Playing Audio clips in Applets.

Text Book(s):

1. Tony Gaddis. "***Starting out with Java: From control structures through objects"***, 5ed, Pearson, 2013

References Book(s):

1. Stuart Reges and Marty Stepp. ""***Building Java programs***: Back to basics approach, 3ed, Pearson, 2014.
2. Robert Sedgewick and Kevin Wayne. "***Introduction to Programming in Java: An Interdisciplinary Approach***", 1e, Pearson, 2014

Object Oriented System Design Lab

Course Objectives

This lab course teaches students how to analyze, design, code, test and debug simple and advanced object oriented applications using Java language.

Course Outcomes

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

- Declare and initialize variables and arrays
- Develop applications using decision and loop statements
- Apply modular programming concepts using functions
- Develop simple OO system using OOP features
- Select and use appropriate data structures
- Store data in files and manipulate files
- Develop GUI applications using Applets

Part-I: Simple Applications

1. Read salaries (*double* values) of three employees and display the highest salary
2. Calculate the grade for a given mark of a student. Grade is A if mark >= 80, B if mark in between 60 and 79, C if mark in between 40 and 59, F if mark less than 40)
3. Check an year leap or not
4. Print the multiplication table of order **NxN**
5. Generate the first***n*** Prime numbers
6. Calculate Factorial of a given number
7. Create an array ***temp*** and read into this array 7 temperature values (in Celsius) of a city, representing the weatherfrom Monday to Sunday and display which day was the hottest in that week

Part-II: Advanced Applications

1) Scientific Calculator: Develop an application that performs the following operations.

- Addition, subtraction, multiplication, division and reminder
- Sin(x), Cos(x), Tan(x)
- Log(x), e^x, 2^x

Note: The application should repeatedly perform operations until a user quits the program, based on some condition (eg. operator == '#')

2) Top Employee: Develop an application that reads salariesof N employees in **M** departments and perform the following operations

- Display the average salaryof each employee
- Displaythe top employee, the one whose average salary is the highest

3) Students Mark List: Create a class *Student* with *rollno, name* and 3 marks (say, *m1, m2, m3*). Create a *constructor* that assignsthe values of rollno, name and 3 marks. Then, create a method ***display()*** that displays the rollno, name and 3 marks. Create another method ***calculate_result()*** that prints the result as either *"Pass"* or *"Fail"*. The student has passed all exams if he secures at least 40 marks in each subject. Create a public class *MarkList* that instantiates ***N*** student objects by reading rollno, name and marks and displays the details and result of those students.

4) Brackets Matcher: Develop an application that reads a sequence of characters, and determines whether its parentheses, braces, and curly braces are balanced. (Hint: for left delimiters, push onto Stack; for right delimiters, pop from Stack and check whether popped element matches right delimiter).

5)Word Counter: Develop an application to read the contents of a large text file and count the occurrences of each word (Hint: use StringTokenizer to parse the file and Hashtable/HashMap to store the words and their count, also use FileReader for reading the file contents).

6)Smiley face Applet: Create an applet that will display 3 smiley faces in three different colors.

Basics of Computer Programming

NMEC – I

Course Objectives:

To impart knowledge on basics of computers and programming techniques.

Learning Outcomes:

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

- Understand Computer Architecture.
- Know Algorithm, flowchart.
- Appreciate C programming

UNIT - I

Introduction to computer: Introduction – Characteristics of computer – Generation of computers – Classification of computers – The computer system – Application of Computer. **Computer Architecture**: Introduction – Central Processing Unit – Memory.

UNIT – II

Computer Program: Introduction – Developing a program – Algorithm – Flow chart. **Computer Languages**: Introduction – evolution of programming languages Classification of programming languages.

UNIT – III

Computer Software: Introduction – Software definition – Relationship between software and hardware – software categories – System Software – Application Software.

UNIT – IV

Introduction to C – overview of computers and interpreters – structure of a C program – C Character set – C keyword – Constants – Variables – Data types – Types Conversion – Operators and Expressions.

UNIT – V

Input and Output in C – Decision statements: IF, ELSE – IF, BREAK, CONTINUE, GOTO and SWITCH. Loop Control statements: FOR, WHILE, DO-WHILE.

Text Books:

1. ***"Introduction to Computer Science"***, ITL Education Solution Limited – First Indian print 2004
2. ***"Programming with ANSI and TURBO C"***, Ashok, N.Kamthane. Third Indian print, 2003

Reference Book:

1. ***"Programming in C"***, Balagurusamy.E, -Tata McGraw Hill, 4th Edition, 2008.

SEMESTER: IV **HOURS/WEEK : 6**
Code: U18CS404 **CREDITS : 5**

Database Design

Course Objectives

This course introduces the design and implementation aspects of a database management system. Students will learn how to model data for the chosen problem using ER diagram, identify a set of relations from ER diagram and implement the relations in the target relational database such as Oracle. Topics include DB architectures, SQL, DBS development life cycle, ER Modeling, Normalization, DB design: conceptual, logical and physical design.

Learning Outcomes

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

- Appreciate the need for DB approach and understand the components and roles of DBMS
- Write SQL queries for the given problem statement
- Apply DB system development life cycle to business problems
- Develop ER diagram for representing conceptual data model
- Convert ER diagram into a set of relations representing logical data model
- Implement a set of relations in the chosen DBMS product, such as ORACLE

Unit I: Introduction and Structured Query Language Part-I

Physical file based system versus database system approaches – DBMS: terminologies, components, roles, advantages and disadvantages – Database architectures: teleprocessing, file server, 2-tier, 3-tier, N-tier, middleware and Transaction processing monitor – Software components of DBMS and Database Manager – Objectives of SQL and BNF notation to define SQL statements – SELECT statement to retrieve all rows – Selecting rows using WHERE clause – Sorting result using ORDERBY clause – SQL aggregate functions – Grouping results using GROUPBY clause and HAVING clause.

Unit II: Structured Query Language Part-II

Designing Sub Queries – Using keywords ANY and ALL – JOIN – inner and outer joins for joining relations – EXISTS and NON EXISTS keywords – Combining results of queries using UNION, INTERSECT and EXCEPT – Updating databases using INSERT, UPDATE and DELETE – SQL data types and Integrity control – Creating, altering and removing tables – Indexes: CREATE and REMOVE. Views: CREATE and REMOVE

Unit III: DBS development life cycle and ER modeling

DBS development lifecycle stages and their activities – DBS development: Requirements analysis – DBS development: Database design – DBS development: DBMS selection and Application design – ER diagram: Entity types and relationship types – Attributes types and keys of ER diagram – Strong and weak entity types; attributes on relationships – Relationship types: One-to-one, One-to-many, Many-to-many – N-ary relationships, cardinality and participation concepts – Connection traps.

Unit IV: Normalization and Conceptual DB design

Purpose of normalization and its support for DB design – Update anomalies: problems related to redundant data – Functional dependencies: characteristics and identification – Normalizations: 1NF, 2NF and 3NF – Normalization Example – Overview of DB Design Methodology – Conceptual DB Design Methodology: building conceptual data model (ie. ER diagram) – Identifying entity types, relationship types and attribute types – Determining domain and key attributes (candidate, primary and alternate) – Checking for redundancy, validating ER diagram and reviewing ER diagram with users.

Unit V: Logical Database design

Logical DB Design Methodology: building logical data model (ie. Relational schema) – Creating relations: for strong entity types and weak entity types – Creating relations: for 1-M binary, 1-1 Binary and 1-1 Recursive relationships – Creating relations: Super class/sub class relationship types – Creating relations: for M-M Binary relationships – Creating relations: for complex relationship types and multi valued attributes – Validating relations using normalization – Checking integrity constraints – Reviewing logical data model with user – Merging logical data models into global model.

Text Book(s):

1. Thomas M. Connolly and Carolyn E. Begg. Database Systems: "***A Practical Approach to Design, Implementation, and Management",*** 6th Edition, Pearson, 2015. **(Chapters 1, 3.1, 3.6, 6, 7, 10, 12, 14, 16, 17 only)**

References Book(s):

1. Hoffer, J. A., Venkataraman, R., and Topi, H. "***Modern database management"*** (12th ed.), Pearson, 2016.
2. Coronel, Morris and Rob. "***Database Systems: Design, Implementation and Management"***, 12th ed, Cengage Learning, 2017.

Database Design Lab

Course Objectives

This lab course teaches students how to design and implement a database system for the given business application. Students will have an opportunity to practice the entire database development life cycle using the target database system.

Learning Outcomes

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

- Develop ER diagram for representing conceptual data model
- Convert ER diagram into a set of relations representing logical data model
- Create tables using the target database system
- Write SQL queries for the given problem statement
- Design and develop database system for the given problem scenario

1. DDL commands in RDBMS.
2. DML and DCL commands in RDBMS Part-I.
3. DML and DCL commands in RDBMS Part-II.
4. High level language extensions with cursers.
5. High level language extensions with triggers.
6. Procedures and triggers.
7. Embedded SQL.
8. Design and implementation of Student information system.
9. Design and implementation of Employee pay roll processing system.
10. Design and implementation of Banking system.

Fundamentals of Information Technology

NMEC - II

Course Objectives:

To impart knowledge on basic of computer, Internet and E-Commerce.

Learning Outcomes:

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

- Computer Classification
- Understand Computer Peripherals
- Appreciate DBMS.
- Know Internet and E-Commerce

UNIT - I

Introduction to computers – Classification of computers – Central Processing Unit (CPU).

UNIT - II

Computer Memory – Secondary storage devices: - Introduction – Classification of secondary storage devices – Advantages – Magnetic Disks – Optical Disk – Magnetic Tape. Input Devices and Technologies: - Introduction – Keyboard – Mouse – TrackBall-Game controllers – Scanners – Barcode Reader – OCR – Digitizer – Voice Recognition – Web Cams – Digital Cameras – Video Cameras (Camcorders).

UNIT - III

Output devices and technologies: - Introduction – Monitor – Printer – Plotter. Computer Software :- Introduction – What is computer software?-Hardware/Software interaction – Software categories – Classification of software – Operating systems – Utilities Compilers and Interpreters – Word processors – Spread sheets – Presentation – Database Management Systems (DBMS)-Image processors.

UNIT - IV

Internet and www:- Introduction – Evolution of Internet – What can I do on the Internet? – Internet protocols – Internet addressing – The World Wide Web (www) – Web pages and HTML – Web browsers – Searching the web-Internet code of conduct.

UNIT -V

E-Commerce – I: Introduction – Definition – Evolution – Types – Business-to-Business (B2B) E-Commerce Business-to-Consumer (B2C) E-Commerce Benefits of E Commerce for companies – Benefits of E-Commerce for customers. E-Commerce – II: E-Money Types of E-Money and E-Money Systems – Digital cash and E-Cash Credit Cards-Debit cards – Smart Cards – Benefits of E-Money – Credit Cards and Credit Card processing – Legal issues of E-Commerce – Ethical issues of E-Commerce.

Text Book:

1. ***"Introduction to Information Technology"*** – By P.Rizwan Ahmed, Margham Publications, 2014.

Reference Books:

1. ***"Introduction to Computer Science"*** – ITL Education Solutions Ltd., Pearson – 2nd Edition-2013(Units I,II,III,IV)
2. ***"Doing Business on the Internet e-commerce"*** By S.Jaiswal, Galgotia Publications Pvt., Ltd., 2010(Unit V)

SEMESTER: V **HOURS/WEEK : 5**
Code: U18CS505 **CREDITS : 5**

Database – Driven Web Design

Course Objectives

This course introduces PHP as a language for Web design with MYSQL. Topics include introduction, PHP functions, OOP, MYSQL Basics, Accessing MYSQL using PHP, Cookies, Sessions, Authentication and Validation.

Learning Outcomes

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

- Appreciate the need for DB approach and understand the components of PHP in Web design
- Write MYSQL queries for the given problem statement
- Connect to MYSQL through PHPMYADMIN.
- Apply language features of PHP and MYSQL to business problems.

Unit I: Fundamentals of PHP

The structure of PHP, Basic syntax of PHP, Incorporating PHP within HTML - Using Comments, Variables, Variable Assignment - Operators, Multiple line commands - Constants, Predefined constants, The echo and print statements, variable scope – Functions - Expressions, Literals and Variables - Operator Precedence with Examples - Associativity, Relational Operators - Conditionals – ((if,else ..)– check minimum bank balance example)(switch – Menu selection example) - Looping (example for all loops – multiplication tables).

Unit II: PHP Fuctions and OOP

Define Function, Return types(Values, array) - Return global variables, variable scope - Include Statement, Using - (include once, require – once) - OOP – Declaring class, objects - Accessing objects, cloning objects, constructors, Destructor - Methods, Declaring constants - Inheritance - Arrays, For each loop, Accessing array elements - Using Array function - File Handling.

Unit III: MYSQL Basics

Database terms, Accessing MYSQL via command line - MYSQL commands – Datatypes – Indexes - Querying MYSQL database - Joining Tables - Using logical operators - MYSQL Functions - Simple example: Student details - Accessing MYSQL via PHPMYADMIN.

Unit IV: Accessing MYSQL using PHP

Querying MYSQL database through PHP - The $-POST array - Table Processing (Create, describe, drop,..) - Data Manipulation through PHP (Insert, update, select, delete), Using AUTO¬_INCREMENT - Simple Example: - Employee details - Performing additional queries - Using Mysqli procedurally - Form Handling: Building forms, Retrieving submitted data - Register globals, Default values, input types, sanitizing input - Example program: program to convert values between Farenheit to Celsius.

Unit V: Cookies, Sessions, Authentication and Validation

Using cookies in PHP - HTTP Authentication - Using Sessions - Validating user Input - Regular Expression, Matching through metacharacters, fuzzy character matching, grouping - Character classes, Indicating range, Negation - General Modifiers - Simple example: Tracking user session in online store, login validation - Using regular expression in PHP - Redisplaying a form after PHP validation.

Text Book(s):

1. ***"Learning PHP, MySQL & JavaScript With jQuery, CSS & HTML5"***, by Robin Nixon, December 2014: Fourth Edition, Copyright © 2015 Robin Nixon. All rights reserved. Published by O'Reilly Media, Inc.,

References Book(s):

1. ***"PHP: The Complete Reference",*** Steven Holzner, Publisher: Mc Graw-Hill, 1st Edition, 2007.
2. ***"PHP and MySQL Web Development"*** (4th Edition), Luke Welling, Laura Thomson Copyright © 2009, Pearson Education, Inc

Principles of Operating Systems

Course Objectives

This course introduces students Operating System basic concepts, main memory management concepts, virtual memory management fundamentals, Process management concepts, device and file management concepts and Android mobile operating system basics

Learning Outcomes

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

- Describe fundamental concepts of operating system
- Explain different types of operating system
- Illustrate basic functionality of the three memory allocation algorithms
- Explain importance of paged and segmented memory management system
- Describe basic functionality of virtual memory management system
- Explain importance of Page replacement algorithms
- Describe basic functionality of process management system, process states, Process control block
- Demonstrate various types of process scheduling algorithms
- Demonstrate basic functionality of device and file management systems
- Describe basic concepts of Android mobile operating system

Unit I: Memory management System

Operating Systems : Overview – Types of operating system – Single Contiguous Memory management system – Fixed Partition Memory management – Dynamic Partition Memory management – Best Fit, Worst Fit and First fit memory allocation algorithms – Internal Fragmentation and External Fragmentation – Re-locatable dynamic partition memory management – Paged memory management – Segmented memory management system.

Unit II: Virtual Memory Management

Virtual Memory management: overview – Demand paging and Page Fault of virtual memory management system – Paged Memory Allocation memory management – Demand Paging memory management – Page Replacement Policies – First come First Serve page replacement algorithm and Belady Anomaly – Least Recently Used page replacement algorithm – Segmented memory allocation management – Segmented/Demand paged memory allocation management – Virtual memory and Cache memory.

Unit III: Process Management

Process Management : overview – Process scheduler, Process status – Process control block and process scheduling algorithms – First In First Serve and Shortest Job Next process scheduling algorithms – Priority scheduling and Shortest Remaining Time process scheduling algorithms – Round Robin process scheduling algorithm – Deadlock, conditions for deadlock – Seven cases of deadlock – Semaphores : concurrency mechanism – Threads and concurrent programming.

Unit IV: Device and File management

Device management and type of devices – Sequential access storage media – Direct access storage devices – Components of I/o sub system of device management – Management of I/O requests of device management – Device Handler seek strategies – FCFS, SSTS disk scheduling algorithms – SCAN, C-SCAN, LOOK , C-LOOK disk scheduling algorithms – File management, file manager – File organization of file management system.

Unit V: Case Study : Android OS Fundamentals

Android mobile operating system: overview – Android Environment setup – Android architecture – Android Application components – Create android application – Anatomy of android application – The main activity file creation – The manifest file creation – The string file, the R file creation – The layout file, Running the android application.

Text Book(s):

1. Ida M. Flynn, Ann MClver MCHocs *,"**Understanding Operating Systems, Course Technology"*** Cengage Learning,2011, 6th Edition, .

References Book(s):

1. Abraham Silberschatz, Peter Baer Galvin, Greg Gagne, *"**Operating System Concepts"**,* Wiley Publication Inc, 2013,9th Edition.

SEMESTER: V **HOURS/WEEK : 5**
Code: U18CS507 **CREDITS : 5**

Computer Graphics

Course Objectives

This course imparts knowledge on fundamental principles of computer graphics.

Learning Outcomes

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

- Classify Image representation.
- Understand 2D & 3D transformation
- Appreciate Mathematics of Projections

UNIT - I

Introduction to computer graphics – Image representation – Display Monitor – Printer – Image files – Setting the color attributes – Scan conversion – Line Drawing Algorithms – Circle Generation Algorithms.

UNIT - II

Region filling – Area filling algorithm – Scan converting a character – Anti aliasing – Recursively defined drawings – Two Dimensional Transformations – Geometric Transformations – Coordinate Transformations – Composite Transformation – Instance Transformations.

UNIT - III

2D viewing and clipping: Window to viewport mapping – Point Clipping – Line Clipping – Cohen Sutherland Algorithm – Midpoint subdivision – Liang Barsky Algorithm – Polygon Clipping – Sutherland Hodgman Algorithm – The Weiler Artherton Algorithm – 2D Graphics pipeline with example.

UNIT - IV

3D Transformations: Geometric Transformations – Coordinate Transformations – Composite Transformation – Instance Transformations – Hidden surfaces – Depth comparisons Z-Buffer Algorithms – Back – Face Removal – The painter's Algorithm – Scan Line Algorithm – Subdivision Algorithm.

UNIT - V

Mathematics of Projections: Taxonomy of projections – Perspective and Parallel Projections – 3D viewing – Viewing Transformations – 3D Graphics Pipeline.

Text Book:

1. ***"Computer Graphics"*** – ZHIGANG XIANG & ROY PLASTOCK, Schaum's outline series McGraw-Hill International Edition, 2001.

Reference Books:

1. ***"Computer Graphics"*** – Donald Hearn & M. Pauline Baker, Pearson Education, 3rd Edition, 2003.

2. ***"Principles of Interactive Computer Graphics"*** – William M. Newman and Robert F. Sproull Mc Graw Hill International Edition, 1998.

SEMESTER: V **HOURS/WEEK : 6**
Code: U18CS5P5 **CREDITS : 4**

Database – Driven Web Design Lab

Course Objectives

This Course introduces Web design using PHP and MYSQL. It enables to create dynamic web pages that tailor themselves to the user.

Learning Outcomes

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

- Apply language features of PHP including strings, regular expressions, files and forms.
- Apply OO features to given problem scenarios.
- Connect to MYSQL through PHPMYADMIN.
- Create web pages with Validation.

1. (a) Construct a PHP program that converts given temperature in celcius to Farenheit and vice versa.
   (b) Develop an application to find the biggest of given three numbers using PHP.
2. (a) Develop an application using String functions of PHP.
   (b) Develop an application to maintain student records using files.
3. (a) Develop a simple calculator using PHP.
   (b) Develop a Login form using PHP.
4. (a) Develop an application to display employee details submitted in forms.
   (b) Develop an application to display customer purchase details stored in MYSQL in table format.
5. Develop an application to manipulate student marksheet stored in MYSQL via PHP.
6. Develop an application to upload a file that contains salary details of employees and check if the file already exists.
7. Develop a page counter application in PHP using Cookies.
8. Develop an application to track user sessions of an online store.
9. Develop a job application with image upload(photo) with required field validation.
10. Develop an application for digital library access with registration and login validation.

SEMESTER: V **HOURS/WEEK : 5**
Code: U18CS5:1 **CREDITS : 5**

Elective - (1a) Software Engineering

Course Objectives

This course introduces the basics of software engineering. Topics include building a software system, principles of software engineering, software process models, requirements specification, design details, implementation issues, testing and maintenance.

Learning Outcomes

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

- Define terms software, software engineering and various software process models.
- Develop software requirements specification.
- Apply design techniques and handle implementation issues.
- Apply testing strategies and software product maintenance

Unit I: INTRODUCTION TO SOFTWARE ENGINEERING

Building a software system: Characteristics of building a system. – Size, complexity, technical considerations – Building hypothetical system – Co-ordination efforts: Process, Product and people. – Engineering of software: Characteristics of software failures. – Software engineering definitions, Relevancy of software engineering and software. – Software engineering Profession and ethics: Code of ethics, Professional behavior – Principle of Software engineering: Davis Early Principle. – Principle of Software engineering: Royces More Principle – Principle of Software engineering: Wassermann's Fundamental Software engineering.

Unit II: SOFTWARE PROCESS MODELS

Software Process : Goal and the simplest process model. – Traditional Process Model: Water Fall Model. – Traditional Process Model: Chief Programmers Team Approach. – Traditional Process Model: Incremental Model. – Traditional Process Model: Spiral Model. – Entry and Exit Criteria: Entry Criteria. – Entry and Exit Criteria: Exit Criteria. – Process Assessment Models: SEI'S Capability Maturity Model. – Process Assessment Models: SEI'S Capability Maturity Model Integrated. – Process Definition and Communication.

Unit III: REQUIREMENTS ENGINEERING

Requirements Processing. – Requirements Elicitation and Gathering: Elicitation of high-level Requirements. – Elicitation of Detailed Requirements. – Requirements Analysis: By Business Flow. – View-Point Oriented Requirements Definitions. – Requirements Analysis and Prioritization. – Requirements Traceability. – Requirements Definitions, Prototyping and Reviews. – Requirements Specifications. – Requirements Agreements.

Unit IV: DESIGN AND IMPLEMENTATION

Introduction To Design: Architectural Design. – Introduction To Design: Detailed Design. – Design Characteristics. – Design Attributes. – Introduction to Implementation: Characteristics of good implementation. – Programming Style. – Coding Guidelines. – Comments. – Debugging. – Performance Optimization

Unit V: TESTING AND MAINTENANCE

Introduction To Testing: Testing and Quality Assurance – Purpose of Testing, Testing Techniques. – Inspections and Reviews. – Formal Methods. – Static Analysis. – Software Support and Maintenance. – Customer Support. – Product Maintenance Updates – Release Cycle. – Change Control.

Text Book(s):

1. ***"Essentials of Software Engineering"***, Frank Tsui, Orlando Karam, Barbara Bernal.3rd Edition

References Book(s):

1. "***Software Engineering"***, Ian Somerville, 10th Edition, Pearson Edition.

SEMESTER: V **HOURS/WEEK : 5**
Code: U18CS5:2 **CREDITS : 5**

Elective – (1b) Internet of Things

Course Objectives:

Learn the interconnection and integration of the physical world and the cyber space. It represents the trend of future networking, and leads the third wave of the IT industry revolution.

Learning Outcomes

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

- Vision and Introduction of IoT.
- Data and Knowledge Management and use of Devices in IoT Technology.
- Real World IoT Design Constraints, Industrial Automation and Commercial Building Automation in IoT.

UNIT I

M2M to IoT: The Vision-Introduction, From M2M to IoT, M2M towards IoT-the global context, A use case example, Differing Characteristics.

UNIT II

M2M to IoT – A Market Perspective: Introduction, Some Definitions, M2M Value Chains, IoT Value Chains, An emerging industrial structure for IoT, The international driven global value chain and global information monopolies. M2M to IoT-An Architectural Overview: Building an architecture, Main design principles and needed capabilities, An IoT architecture outline, standards considerations.

UNIT III

M2M and IoT Technology Fundamentals: Devices and gateways, Local and wide area networking, Data management, Business processes in IoT, Everything as a Service(XaaS), M2M and IoT Analytics, Knowledge Management . IoT Architecture-State of the Art: Introduction, State of the art. Architecture Reference Model: Introduction, Reference Model and architecture, IoT reference Model.

UNIT IV

IoT Reference Architecture: Introduction, Functional View, Information View, Deployment and Operational View, Other Relevant architectural views. Real-World Design Constraints: Introduction, Technical Design constraints-hardware is popular again, Data representation and visualization, Interaction and remote control.

UNIT V

Industrial Automation: Service-oriented architecture-based device integration, SOCRADES: realizing the enterprise integrated Web of Things, IMC-AESOP: from the Web of Things to the Cloud of Things. Commercial Building Automation: Introduction, Case study: phase one-commercial building automation today, Case study: phase two- commercial building automation in the future.

Textbook:

1. Jan Holler, VlasiosTsiatsis, Catherine Mulligan, Stefan Avesand, Stamatis Karnouskos, David Boyle, ***"From Machine-to-Machine to the Internet of Things: Introduction to a New Age of Intelligence"***, 1st Edition, Academic Press, 2014.

Reference Book:

1. Vijay Madisetti and ArshdeepBahga, ***"Internet of Things (A Hands-on-Approach)",*** 1st Edition, VPT, 2014

SEMESTER: V **HOURS/WEEK : 5**

Elective – (1c) Cloud Computing

Course Objectives:

To impart knowledge on Cloud Computing basics and the various Cloud Computing Architectures, infrastructure models and the Fundamental Concepts cloud services.

Learning Outcomes

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

- Pronounce Task Management and Event Management.
- Evaluating Web-Based Presentation Applications

Unit – 1:

Beyond the Desktop: Introduction to the Cloud Computing- Are you ready for computing the Cloud? -Developing Cloud Services.

Unit – 2:

Cloud Computing for Everyone- Cloud Computing for the Family- Cloud Computing for the Community- Cloud Computing for the Corporation.

Unit – 3:

Using cloud Services: Collaborating on Calendars, Schedules, and Task Management -Collaborating on Event Management -Collaborating on Contact Management -Collaborating on Project Management

Unit – 4:

Using cloud Services: Collaborating on Word -Collaborating on Spreadsheets-Collaborating on Presentations: Preparing Presentations Online-Evaluating Web-Based Presentation Applications

Unit – 5:

Using cloud Services: Collaborating on Databases- Storing and Sharing files and other online content: Understanding Cloud storage- Evaluating Online File Storage and Sharing Services.

Text Books:

1. Michael Miller, ***"Cloud Computing",*** Pearson Education Inc, 7th Edition, 2012

References Book:

1. RajkumarBuyya& Co., ***"Cloud Computing Principles and Paradigms"***, John Wiley & Sons Publications, 2011.

SEMESTER: V **HOURS/WEEK : 2**

SBEC – II Project Proposal

Course Objectives

The Project Proposal emphasizes on application of the theoretical concepts of software analysis and design concepts. The analysis component consists of preparing formal Software Requirements Specifications (SRS) document. The SRS will contain a problem statement, scope, justification, requirements, cost estimation, assumptions, limitations, methodology and tools to be used in the development of the project. The scope of the proposed problem should be clear and well defined in the problem statement. All the functional and non-functional requirements of the system should be identified and analyzed in the proposal. The students will be encouraged to develop logical model of the proposed system based on the identified requirements. The design component of the course includes prototype including input and output of the proposed system.

Learning Outcomes

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

- Identify and define the problem statement
- Define and justify scope of the proposed problem
- Gather and analyze system requirements
- Propose an optimized solution among the existing solutions
- Practice software analysis and design techniques
- Develop technical report writing and oral presentation skills

Text Book(s):

1. Lynn E. Miner & Jeremy T. Miner, ***"Proposal Planning and Writing",*** Third Edition, Greenwood Publishing Group, 2003,

References Book(s):

1. William Navidi, ***"Statistics for Engineers and Scientists",*** 2nd Edition, McGraw-Hill, 2007.

.

SEMESTER: V	**HOURS/WEEK : 2**
Code: U18CSPS3	**CREDITS : 2**

Technical Communication for Computer Scientists

SBEC - III

Course Objective

This course trains students soft skills required to take up jobs.

Learning Outcomes

Upon completion of this course students should be able to

- Demonstrate their activity to manage conversations, and take part in discussion
- Take notes, and effectively prepare documents

1. **Oral communication:** Starting and ending a conversation, telling and asking people to do things- expressing opinions and ideas- decisions and intentions- offers and invitations-feelings- right and wrong- numbers and money.

2. Purpose and audience- dealing with customers and clients- face to face discussions- meetings and attending meetings- checking understanding- raising questions- giving and receiving feedback- using body language- leading and directing discussions- concluding discussions, using graphics in oral presentation.

3. Reading comprehension and reference skills: Skimming and scanning, factual and inferential comprehension- prediction- guessing meaning of words from context- word reference-comprehending graphics in technical writing.

4. Reading strategies- reading speed, reading between lines for hidden meaning- interpreting graphics- using a dictionary- using an index- using a contents list to find information-choosing the right reference source.

5. **Written Communication:** Note making and note taking- summarizing- notes and memos-developing notes into text- organization of ideas- cohesion and coherence- paragraph writing- ordering information in space and time- short essays- description and argument-comparison and contrast- illustration- using graphics in writing- tables and charts- diagram and flow charts- maps, plans and graphs.

6. Spelling rules and tips- writing a rough draft- editing and proof reading- writing the final draft- styling text- filing in complex forms- standard letters- CV- writing a report- writing leaflets and brochures- writing references- essay writing- expository writing- description of processes and products- classification- the instructional process- arguments and presentation of arguments- narrating events chronologically.

SEMESTER: VI **HOURS/WEEK : 6**
Code: U18CS608 **CREDITS : 5**

Computer Networking

Course Objectives

This Course Provides information of Computer Network and its Layered Services

Learning Outcomes

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

- Apply Layered Architecture into LAN.
- Learn Socket Programming in Network.
- Understand Reliable Data transfer in Network.
- Appreciate the need of Error Detection and Correction in the network.

Unit I: Introduction of Internet and Networking Protocols

Internet (Network Description, Network Services & Protocol) - Components of a Computer Network : Access Network, Physical media - Network Switching : Packet Switching, Circuit Switching - Quantitative QoS(Delay, Loss, Throughput, and Bandwidth) : Overview of Delay and Packet Loss, Queuing Delay and Packet Loss, End-to-End Delay, Throughput in Computer Network - Protocols Layers and their Service Models : Layered Architecture, Encapsulation.

Unit II: Application Layer

Network Application : Application Architectures - Process Communication - The web and HTTP : Overview of HTTP, HTTP Message format,Non persistent and Persistent Connection, Cookies, web caching-File Transfer Protocol (FTP) - Electronic Mail: SMTP, Mail Access Protocol - DNS : Services, Functions, Records and Messages - Peer-to-Peer Applications: File Distributions – Architectures, BitTorrent- Peer-to-Peer Applications : Distributed Hass Table- Socket Programming.

Unit III: Transport Layer

Transport Layer Services - Multiplexing and De multiplexing (Connectionless & Connection-Oriented) - Connection Less Transport : UDP - Reliable Data Transfer - Connection-Oriented Transport: TCP Connection, TCP Segment Structure, Round-Trip Time Estimation, Reliable Data Transfer (Doubling the Time interval), Reliable data Transfer (fast Retransmit, Go-Back-N), TCP Connection Management - Congestion Control - TCP Congestion Control.

Unit IV: The Network Layer

Responsibilities of Network Layer (Forwarding & Routing, Network Service Model) - Virtual Circuit and Datagram Network - Router: Input Processing, Switching, & Output Processing, Routing Control plane, Internet Protocol(IP): Datagram format - Internet Protocol(IP): IPv4 Addressing, ICMP, IPv6, & IP Security - Routing Algorithm: Distance Vector, Link State, Hierarchical - Routing in the Internet: RIP, OSPS, BGP - Broadcast and Multicast Routing.

Unit V: The Link Layer

Link Layer: Services, Implementation - Error-Detection & Correction: Parity, Checksum, & CRC - Multi Access Link and Protocols: Channel Partitioning protocol, Talking-Turns Protocols DOCSIS, Random Access Protocol - Switched Local Area Network: Link-Layer Addressing and ARP, Ethernet, and Link Layer Switches - Link Virtualization - Data Center Networking.

Text Book(s):

1. James F Kurose and Keith W. Ross, ***“Computer Networking”*** – A Top-Down Approach, Sixth Edition, 2013, Pearson Publication.

References Book(s):

1. Mani Subramanian; Tlmothy A. Gonsalves; N. Usha Rani, “***Network Management: Principles and Practice”,*** Pearson Education India, 2010

SEMESTER: VI **HOURS/WEEK : 6**
Code: U18CS6:1 **CREDITS : 5**

Elective – (2a) Multimedia Technologies

Course Objectives

This course introduces the various technologies underlying digital images, videos and audio contents, including various compression techniques and standards.

Learning Outcomes

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

- Appreciate the need for multimedia and understand its components
- Understand the basics of data and color representations
- Understand the various concepts involved in video and audio
- Appreciate the need for data compression
- Relish the experience with MPEG

Unit I: Introduction to multimedia

Getting started with multimedia and its components – World wide web:History of WWW – World wide web:HyperText Transfer Protocol – World wide web:Extensible Markup Language – World wide web:Synchronised Multimedia Interaction Protocol – Overview of multimedia software tools – Multimedia authoring:production – Multimedia authoring:presentation – Overview of Virtual Reality Modeling Language – Animation and interactions.

Unit II: Data representations and color representations

Graphics data types:1-bit images,8-bit Gray level images – Image Data Types:24Bit Color Images,8 bit Color Images,Color Lookup Tables – File Formats – Color Science: Human Vision,Camera Systems,Color Matching Functions – Color Models in Images: RGB Color Model for CRT Displays – Color Models in Images: Subtractive Color Model – Color'Models in Images:Transformation from RGB to CMY – Color'Models in Images: CMYK System – Color models in video – Types of video signals – Analog video – Digital video.

Unit III: Basics of digital audio

Digitization of sound:definition,Nyquist Theorem – Digitization of sound:Signal to Noise Ratio – Digitization of sound:Signal to quantization noise,Audio Filtering – Musical Instrument Digital Interface Overview – Hardware Aspects of MIDI – Structure of MIDI Messages:Channel Messages – Structure of MIDI Messages:System Messages – Quantization and transmission of audio:Coding of audio,Pulse Code Modulation:General – Quantization and transmission of audio:Pulse Code Modulation in speech compression – Quantization and transmission of audio:Differential Coding Audio.

Unit IV: Data Compression

A general data compression scheme and forms of data compression – Shanon-Fano Algorithm – Huffman Coding-algorithm – Huffman Coding –properties – Dictionary Based-The Lempel-Ziv-Welch (LZW) algorithm – LZW decompression algorithm – Steps in JPEG Image Compression – Steps in JPEG Image Compression – Video Compression – Video Compression.

Unit V: MPEG Video and Audio Coding

Introduction to MPEG -1 – Motion Compensation in MPEG-1 – MPEG-1 Video Bitstream – Introduction to MPEG-2 and its support to interlacing – Differences between MPEG-1 and MPEG -2 – MPEG Layers and Audio Strategy – MPEG Audio Compression Algorithm – MPEG Audio Compression Algorithm – MPEG-2 Advanced Audio Coding– MPEG-4 Audio.

Text Book(s):

1. Ze-Nian Li & Mark S Drew, ***"Fundamentals of Multimedia"***, Pearson Education,2004.

References Book(s):

1. Dr. Ashok Banerji & Dr. Ananda Mohan Ghosh ***,"Fundamentals of Multimedia Technology"***,McGraw Hill Education,2010.

Elective – (2b) Web Applications Development

Course Objectives

This course imparts knowledge on the development of Front end application using VB.NET and the web controls of ASP.NET.

Learning Outcomes

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

- Understand DLL, COM.
- Looping Statement and Condition Construct
- Develop dot net application

Unit I

Evolution of the .Net framework – Overview of the .NET framework – DLL,COM and Assemblies – VB.Net language – IDE main window – Value data type – Reference data type – Arithmetic operators.

Unit II

IF–statement – Select-case statement – While statement – do statement – For statement – Methods and arrays – Class – Constructor overloading – Copy constructor – Instance class.

Unit III

Inheritance – Abstract class and abstract method – Interface: Multiple Interface, Inheritance – Namespace – Access modifiers – Exception handling – Multithreading.

Unit IV

Text box control – Label control – Button Control – Radio button – Check box control – Group box control – List box control – Combo box – Timer control – Link Label – Dialog box– Tool bar control – Status bar control.

Unit V

Basic Web Control: Introduction – ASP.Net Object Model – Server-side Control – Calendar Control – AdRotator Control – Validation Control – User Control.

Text book

1. “***Visual Basic.Net”***, by C. Muthu, Vijay Nicole Imprint Private Limited, 2007.

Reference Books

1. ***“VB.Net”,*** by Radhaganesan, Scitech Publications Pvt., Ltd., 2004.
2. ***“Visual Basic.Net: Tips & Techniques”***, Kris Jamsa, Tata-McGrawHill book company, 2002.

Elective – (2c) XML Basics

Course Objectives

This course imparts knowledge on Meta mark-up languages, XML basics, Style sheets, and XML's significance in web documents.

Learning Outcomes

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

- XML documents
- XML Schemas
- Handling XLinks,XPointers and XForms

Unit 1:

Creating XML Documents: Welcome to XML-All about XML-Working with XML-data yourself-Structuring your data-Creating well-formed XML documents-Creating valid XML documents. **Creating XML Documents**: Choosing XML editor-Using XML browsers-Using XML validators-creating XML documents piece by piece-Creating XML documents, XML comments, processing instructions, tags and elements, CDATA sections and Handling entities. **Creating well-formed XML Documents:** Understanding well formed ness Constraints-Using XML namespaces.

Unit II:

Creating Valid XML Documents DTD's: Validating a document by using a DTD-Creating element content models-Commenting a DTD-Supporting XML dtd's-Handling namespace in DTD's. **Handling attributes and Entities in DTDs:** Declaring attributes in DTD's-Specifying default values-Specifying attribute types-Handling Entities. **Creating valid XML documents: XML Schemas**-Creating XML schemas-Creating elements and types-Specifying number of elements-Element default values-Creating attributes.

Unit III:

Creating types in XML schemas:-Creating XML schema Choices-Using anonymous type definition-Declaring empty elements-Mixed content element-Grouping element together-attributes together-Declaring all groups-Handling namespace in schemas. **Formatting XML by using Cascading Style Sheet:** Introducing CSS-Connecting CSS style sheet and XML documents-Creating Style sheet selectors-Using inline styles-Creating Rule specification in style sheets.

UnitIV:

Formatting XML by using XSLT: Introducing XSLT-Transforming XML by using XSLT-Writing XSLT stylesheet-Matching nods by using match attributes- Working with the select attributes and XPath-using <xsl:copy>-<xsl:if>-<xsl:choose>. **Handling XLinks,XPointers and XForms:** Introducing XLinks-Beyond simple XLinks-Introducing XPointers-Introducing XBase,XForms-Creating Input Controls

Unit V:

Using Javascript and XML: Introducing the W3C DOM-Introducing DOM Objects-Working with XMLDOM in Javascript-Searching for elements by name-Reading attribute values. **Using java and .Net DOM:** Using java to read XML data-Finding element by name- Writing XML using DOM.**Using Java and .NET SAX:** Using SAX-Using SAX to find elements by name-Writing XML by using Java and SAX.

Text Book:

1. ***"XML in 21 days"* -** Steven Holzner, Pearson Education ,Third Edition, First Indian Reprint, 2004

Reference Book:

1. ***"XML Bible"*** - Elliotte Rusty Harold,IDG Books India, First Edition 2000.

Elective – (3a) Business Analytics

Course Objectives

This course provides the fundamental concepts and tools needed to understand the emerging role of business analytics in organizations to apply business analytics tools in spreadsheet environment, and to communicate with Analytics Professionals to effectively use and interpret analytic models and results for making better business decisions.

Learning Outcomes

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

- Learn to use and apply various Excel tools and add-ins.
- Learn to compare mathematical formulas with Spreadsheet formulas
- Learn how to use tools and techniques effectively and work through basic numeric calculations
- Gain better understanding of concepts of Business Analytics and
- Increase proficiency in MS Excel.

Unit I: INTRODUCTION TO BUSINESS ANALYTICS

What is Business Analytics?- Evolution of Business Analytics-Scope of Business Analytics- Data for Business Analytics-Models in Business Analytics-Problem solving with Analytics.

Unit II: ANALYTICS ON SPREADSHEETS

Basic Excel skills - Basic Excel Functions - Using Excel Lookup functions for Database Queries - Spreadsheet Add-Ins for Business Analytics.

Unit III: DESCRIPTIVE ANALYTICS – Visualizing and Exploring Data

Data Visualization - Creating charts in Microsoft Excel - Other Excel visualization tools - Data queries: Tables, Sorting and Filtering – Statistical Methods for Summarizing Data.

Unit IV: DESCRIPTIVE STATISTICAL MEASURES

Population and Samples - Measures of Location - Measures of Dispersion - Measures of Association - Measures of Shape - Excel Descriptive Statistical Tool - Statistical thinking in Business Decisions.

Unit V: PREDICTIVE ANALYTICS –Trend lines and Regression Analysis

Trend Lines and Regression Analysis: Modeling Relationships and trends in data- Simple Linear Regression- Forecasting Techniques: Qualitative and Judgmental forecasting-Historical Analogy – The Delphi Method – Statistical Forecasting models – Forecasting models for stationary time series.

Text Book(s):

1. ***"Business Analytics"***, James R. Evans, Second Edition, Pearson Education, 2016. Indian Edition 2017, Pearson India Services.

References Book(s):

1. ***"Essentials of Business Analytics"***, CAMM, COCHRAN, FRY, OHLMANN, ANDERSON, SWEENEY, WILLIAMS, 2015, CENGAGE LEARNING

SEMESTER: VI **HOURS/WEEK : 6**
Code: U18CS6:5 **CREDITS : 5**

Elective – (3b) Foundations of Enterprise Computing

Course Objectives:

This course imparts programming knowledge on J2EE.

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

- Client Tier Implementation.
- Threads, Interfaces
- . Java Server Pages and JDBC

Unit I
J2EE and J2SE-J2EE Multi-Tier Architecture: Distributive Systems-The Tier-J2EE Multi-Tier Architecture-Client Tier Implementation-Web Tier Implementation-Enterprise JavaBeans Tier Implementation-Enterprise Information Systems Tier Implementation-Challenges. J2EE Best Practices: Enterprise Application Strategy-The Enterprise Application-Clients-Sessions Management-Web Tier and JavaServer Pages.

Unit II
Enterprise JavaBeans Tier-The Myth of using Inheritance-Maintainable Classes-Performance Enhancements-The Power of Interfaces-The Power of Threads-The Power of Notification. J2EE Design Patterns and Frameworks: The Pattern Concept-Pattern Catalog.

Unit III
J2EE Database Concepts: Data-Database-Database Schema-The Art of Indexing. JDBC Objects: The Concept of JDBC-JDBC Driver Types-JDBC Packages-A Brief Overview of the JDBC process-Database Connection-Associating the JDBC/ODBC Bridge with the Database-Statement Objects-ResultSet-Transaction Processing-Metadata.

Unit IV
Java Servlets: Java Servlets and Common Gateway Interface programming-A Simple Java Servlet-Anatomy of a Java Servlet. Java Server Pages: JSP-JSP Tags-Tomcat-Request String-User Sessions-Cookies-Session Objects.

Unit V
Enterprise JavaBeans: Enterprise JavaBeans-Deployment Descriptors-Session Java Bean-Entity Java Bean-Message Driven Bean-The JAR File.

Text Book:

1. ***"J2EE: The Complete Reference"***, Jim Keogh, Tata McGraw Hill Publishing Company, New Delhi 2002

Reference Books:

1. ***"J2EE Bible"***, Justin Couch and Daniel H. Steinberg, Willey India Pvt. Ltd, New Delhi, 2002
2. ***"Instant Enterprise Java Beans",*** Paul Tremblett, Tata McGraw Hill Publishing Company, New Delhi 2001

SEMESTER: VI **HOURS/WEEK : 6**
Code: U18CS6:6 **CREDITS : 5**

Elective – (3c) Mobile Application Development

Course Objectives

This course enables students to develop a working knowledge of Android app development tool. and to identify need and opportunity in app markets

Learning Outcomes

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

- Understand mobile devices and mobile platforms
- Demonstrate earn wireless communication standards and data transmission standards
- Setup programming tools for a mobile application developer

UNIT I

Developing Spectacular Android Applications: Why Develop for Android? - Android Programming Basics – Hardware Tools – Software Tools.

Prepping Your Development Headquarters: Assembling Your Toolkit – Installing and Configuring Your Support Tools – Getting Acquainted with the Android Development Tools.

UNIT II

Your First Android Project: Starting a New Project in Eclipse – Deconstructing Your Project – Setting up an Emulator – Creating Launch Configurations – Running the Hello Android App – Understanding the Project Structure.

Designing the User Interface: Creating the Silent Mode Toggle Application – Laying Out the Application – Developing the User Interface – Adding an Image to Your Application – Creating a Launcher Icon for the Application – Adding a Toggle Button Widget – Previewing the Application in the Visual Designer.

UNIT III

Coding Your Application: Understanding Activities – Creating Your First Activity – Working with the Android Framework Classes – Installing Your Application – Reinstalling Your Application – Responding to Errors. Understanding Android Resources: Understanding Resources – Working with Resources.

UNIT IV

Handling User Input: Creating the User Input Interface – Getting Choosy with Dates and Times – Creating Your First Alert Dialog Box – Validating Input.

UNIT V

Getting Persistent with Data Storage: Finding Places to Put Data – Asking the User for Permission – Creating Your Application's SQLite Database. Creating and Editing Tasks with SQLite: Inserting, Deleting and Updating a task.

Text Book:

1. Donn Felker and Joshua Dobbs, ***"Android Application Development – for Dummies"***, Wiley Publishing Inc., 2011.

Reference Book:

1. Jerome (J.F.) DiMarzio, ***"Android – A Programmer's Guide",*** Tata McGraw-Hill Publication, 2008.

SEMESTER: VI **HOURS/WEEK : 6**
Code: U18CS6PJ **CREDITS : 5**

Project Implementation

Course Objectives

In this phase of the course, students are required to implement the proposed design of the project. The students will review the design specification and make any necessary enhancements to synchronize the implementation details. The students will identify and learn the use of tools required for the project implementation. The students will be expected to: prepare application architecture, code, debug, document, and test the application software within suggested timeframe. Efforts will be made to ensure the quality of software project through various evaluation aspects such as professional coding style, documentation of code, intuitive user interface design, input validation, verification and user guide. The students will be further required to evaluate the developed system by generating test cases of the critical components of the designed model.

Learning Outcomes

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

- Develop a functional application based on the software design
- Apply coding, debugging and testing tools to enhance the quality of the software
- Construct new software system based on the theory and practice gained through this exercise
- Prepare the proper documentation of software projects following the standard guidelines
- Learn technical report and oral presentation skills

Text Book(s):

1. Lynn E. Miner & Jeremy T. Miner, ***"Proposal Planning and Writing",*** Third Edition, Greenwood Publishing Group, 2003.

References Book(s):

1. William Navidi, ***"Statistics for Engineers and Scientists",*** 2nd Edition, McGraw-Hill, 2007.