



SYLLABUS FOR BCA (MAJOR)

Under Single Major Single Minor (FYUGP)
(To be implemented from Session 2024-25)

SEM. I & II

Proposed Syllabus for four years B.C.A. (Major) Programme

Year	Semester	Paper Code	Paper	Credits	Periods/Week	Exam. Marks	Practical /Tutorial	Total
1st Year	I	BCAPMAJ101	Computer Fundamentals	3	3	60	-	80
		BCAPMAJ101T	Computer Fundamentals (Tutorial)	1	1	-	20	
		BCAPMAJ102	Programming Fundamentals Using C	3	3	60	-	80
		BCAPMAJ102L	Programming Fundamentals Using C (Lab)	1	1	-	20	
		POOASEC105 POOASEC106	Basic Programming in Python MS PowerPoint	2	2	40	-	60
		POOASEC105L POOASEC106L	Basic Programming in Python (Lab) MS PowerPoint (Lab)	1	1	-	20	
		MIN1	Student has to choose only ONE discipline from the subjects given below: 1. Physics 2. Mathematics 3. Statistics 4. Economics 5. Business Management	4	4	60	20	80
		VAC	Environmental Education	3	3	60	-	80
		VACT	Environmental Education (Tutorial)	1	1	-	20	
	II	BCAPMAJ203	Digital Electronics	3	3	60	-	80
		BCAPMAJ203L	Digital Electronics (Lab)	1	1	-	20	
		BCAPMAJ204	Java Programming	3	3	60	-	80
		BCAPMAJ204L	Java Programming (Lab)	1	2	-	20	
		POOBSEC218 POOBSEC219	Cyber Security HTML Programming	2	2	40	-	60
		POOBSEC218T POOBSEC219L	Cyber Security (Tutorial) HTML Programming (Lab)	1	1	-	20	
		MIN2	Student will be provided the SAME discipline from the subjects selected previously as Minor.	4	4	60	20	80
		AEC1	Compulsory English	3	3	60	-	80
		AECT	Compulsory English (Tutorial)	1	2	-	20	
		IDC1	Student has to choose only ONE discipline from a pool of subjects that will be provided.	2	2	40	-	60
		IDCT	Inter Disciplinary (Tutorial)	1	1	-	20	
		IN1	Summer Internship The Colleges are expected to network with skill development centres, vocational training institutes for facilitating student internships. Online based internships programs are also permitted in case of BCA (Major) students. The students must submit a certificate of completion of the internship at the end of the semester.	2	-	-	-	-

NOTE: Tutorials should involve problem solving session/activity related to the subject taught

1st Year: Semester I
--

Course-MAJOR	Paper Code - BCAPMAJ101	Credits-3	Lectures-45 /Week-3
---------------------	--------------------------------	------------------	----------------------------

COMPUTER FUNDAMENTALS

Prerequisite(s) and/or Note(s):

- (1) Basic knowledge of Computers

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances.
However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge acquired:

- (1) Introduce students to the fundamental concepts of system tools, their functionalities, and their role in computer systems.
- (2) Familiarize students with various peripheral devices commonly used in computer systems and their functionalities.
- (3) Develop students' proficiency in utilizing system tools to optimize computer performance, troubleshoot issues, and maintain system integrity.
- (4) Enable students to effectively manage peripheral devices, including printers, scanners, external storage devices, and input/output devices.

Skills gained:

- (1) Basic idea programming.
- (2) Troubleshooting hardware and software issues.
- (3) Efficient internet and file management.

Competency Developed:

- (1) Problem-solving in technical contexts.
- (2) Adaptability to various software environments.
- (3) Effective communication of technical concepts.

Syllabus Overview

Unit 1 : Introduction of Computer

10 Lectures

Introduction, Definition, Characteristic of computer, Block Diagram of a computer, Generation of Computers, Classification of a Computers, Software: Systems and Application, Hardware.

Unit 2 : Peripheral Devices and Storage Organization

15 Lectures

Input and Output Devices – Punched Card, Keyboard, Mouse, Joystick, Trackball, Light Pen, Touch Screen, Magnetic Ink Character Recognition (MICR), Optical Character Recognition (OCR), Optical Mark Recognition (OMR), Display units, Printers- Impact and Non-Impact. Primary storage – RAM-SRAM, DRAM, ROM-PROM, EPROM, EEPROM, Secondary storage – Hard drive, Magnetic drive, Compact Disk, Cache memory, components of motherboard.

Unit 3 : Basics of Operating Systems (OS)

10 Lectures

Definition, types of OS programs, types of Operating System (OS) - Batch processing, Multiprogramming, Multiprocessing, Multitasking, Time sharing, Virtual storage, Functions of OS, Advantages of OS, Popular OS Disk Operating System (DOS) and Windows.

Unit 4 : Problem solving and programming

10 Lectures

Programming language: Machine language, Assembly language, High level language and their merits and demerits, Algorithms: Definition, Algorithm for simple problems, Flowchart: Definition, Need for flowchart, basic symbols of flowchart. Simple flowcharts.

Suggested Readings

1. M.M.Jana & N.K.Samanta, An overview of Information Technology and its Application in Business,NCBA(P) LTD.,2007.
2. Pradeep K.Sinha & Priti Sinha, Computer Fundamentals, 3rd Edt, BPB Publication.
2. Sinha P.K., “Computer Fundamentals”, 6th Edition, BPB Publication, 2012.
2. Rajaraman,V,“Computer Fundamentals”, 6th Edition, PHI,2012.
3. Thareja R., “Fundamentals of Computers”, Oxford University Press, 2014.
4. Stallings W., “Operating systems”, 8th Edition, Pearson, 2014.

Course – MAJOR

Paper Code- BCAPMAJ101T

Credits-1,Tut/Week - 1

Paper- Computer Fundamentals (Tutorial)

Computer Fundamentals Tutorial as assigned and advised by teacher(s).

PROGRAMMING FUNDAMENTALS USING C

Prerequisite(s) and/or Note(s):

- (1) High school mathematics.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives

Knowledge acquired:

- (1) Knowledge about program development and implementation
- (2) Syntax of C programming language
- (3) Knowledge about how humans interact with computers through a language.

Skills gained:

- (1) Problem solving skills
- (2) Logical thinking to approach a problem
- (3) Building programs for different problems at hand.

Competency Developed:

- (1) Applying the skills learnt to model real world problems
- (2) Facility in solving real life problems by thinking logically and outside of box.
- (3) Ease of switching to any other programming language

Syllabus Overview

Unit 1: Introduction to C, Data Types, Variables and Operators

6 Lectures

History of C, Overview of Procedural Programming, Introduction to Algorithm & Flowcharts. Using main() function Compiling and Executing Simple Programs in C. Declaring, Defining and Initializing Variables, Scope of Variables, Using Named Constants, Keywords, Data Types, Casting of Data Types, Operators (Arithmetic, Logical and Bitwise), Using Comments in programs, Character I/O (getc, getchar, putc, putchar etc), Formatted and Console I/O (printf(), scanf()) , Using Basic Header Files (stdio.h, conio.h, stdlib.h, etc).

Unit 2: Expressions, Conditional Statements and Iterative Statements

10 Lectures

Simple Expressions in C (including Unary Operator Expressions, Binary Operator Expressions), Understanding Operators Precedence in Expressions, Conditional Statements (if construct, switch- case construct), Understanding syntax and utility of Iterative Statements (while, do-while, and for loops), Use of break and continue in Loops, Using Nested Statements (Conditional as well as Iterative)

Unit 3: Understanding Functions

5 Lectures

Utility of functions, Call by Value, Call by Reference, Functions returning value, Void functions, Return type of functions, Functions parameters, Differentiating between Declaration and Definition of Functions, Functions with variable number of Arguments.

Unit 4: Implementation of Arrays and Strings

10 Lectures

Creating and Using One Dimensional Arrays (Declaring and Defining an Array, Initializing an Array, Accessing individual elements in an Array, Manipulating array elements using loops), Use Various types of arrays (integer, float and character arrays / Strings) Two-dimensional Arrays (Declaring, Defining and Initializing Two Dimensional Array, Working with Rows and Columns). String handling functions.

Unit 5: User-defined Data Types (Structures and Unions)

5 Lectures

Understanding utility of structures and unions, Declaring, initializing and using simple structures and unions, Manipulating individual members of structures and unions, Array of Structures.

Unit 6: Pointers and References in C

4 Lectures

Understanding a Pointer Variable, Simple use of Pointers (Declaring and Dereferencing Pointers to simple variables)

Unit 7: File and I/O

5 Lectures

Opening and closing a file, Reading and writing Text Files, Using put(), get(), read() and write() functions, Random access in files.

Suggested Readings

1. "The C Programming Language ANSI C Version", Kernighan & Ritchie, Prentice Hall Software Series
2. "ANSI C - Made Easy", Herbert Schildt, Osborne McGraw-Hill
3. "Learning to Program in C", N. Kantaris, Babani
4. "C - The Complete Reference", Herbert Schildt, Osborne McGraw-Hill
5. "Programming in C", Reema Thareja, Oxford University Press
6. "A First Course in Programming With C", T. Jeyapoovan, Vikas Publishing House
7. "Let Us C", Yashavant P. Kanetkar, BPB Publications

Paper: Programming Fundamentals using C (Lab)

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. WAP to perform input/output of all basic data types.
2. WAP to enter two numbers and find their sum.
3. WAP to find the simple interest.
4. WAP to Swap Two Numbers (using and without using a third variable).
5. WAP to check whether a number is even or odd
6. WAP to compute the factors of a given number.
7. WAP to print the sum and product of digits of an integer.
8. WAP to check whether a character is vowel or consonant
9. WAP to find the largest among three numbers
10. WAP to compute the sum of the first 'n' terms of the following series $S = 1 - 2 + 3 - 4 + 5 + \dots n$
11. WAP to compute the sum of the first 'n' terms of the following series $S = 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots n$
12. WAP to find the factors of a number.
13. WAP to print all the prime numbers within a given range
14. WAP to display the Fibonacci series.
15. WAP to find the factorial of a number.
16. WAP to check if a number is prime or not.
17. WAP to check if a number is Armstrong or not.
18. WAP to check if a number is Perfect or not.
19. WAP to print a triangle of stars as follows (take number of lines from user):

* ** *** **** *****	* ** *** **** *****	***** **** *** ** *	***** **** *** ** *	* ** *** **** *****	* *
1 1 2 1 2 3 1 2 3 4 1 2 3 4 5	5 4 5 3 4 5 2 3 4 5 1 2 3 4 5	1 2 2 3 3 3 4 4 4 4 5 5 5 5 5	5 4 4 3 3 3 2 2 2 2 1 1 1 1 1	A A B A B C A B C D A B C D E	1 1 2 1 1 2 3 2 1 1 2 3 4 3 2 1 1 2 3 4 5 4 3 2 1

BASIC PROGRAMMING IN PYTHON***Prerequisite(s) and/or Note(s):***

(1) High school mathematics.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course. ***Course***

Objectives***Knowledge Acquired:***

- (1) Fundamental Concepts: Students acquire knowledge of fundamental programming concepts such as variables, data types, loops, conditionals, and functions in Python.
- (2) Data Structures: They learn about essential data structures like lists, tuples, dictionaries, and sets, understanding their usage and implementation.

Skills Gained:

- (1) Coding Proficiency: Through hands-on practice and assignments, students develop coding proficiency in Python, enabling them to write clear, concise, and functional code.
- (2) Problem-Solving: They enhance their problem-solving skills by applying Python programming concepts to solve various computational problems and algorithms.
- (3) Debugging and Troubleshooting: Students acquire skills in debugging code and troubleshooting errors, learning how to identify and fix common programming mistakes effectively.

Competency Developed:

- (1) Logical Thinking: Python programming exercises require logical thinking and algorithmic problem-solving skills, helping students develop a logical mindset.
- (2) Attention to Detail: Writing code necessitates attention to detail to ensure accuracy and functionality. Students develop this competency through debugging and code review processes.
- (3) Collaboration and Documentation: Students learn to collaborate on coding projects using version control systems like Git and to document their code effectively, enhancing their ability to work in teams and communicate technical concepts clearly.

Syllabus Overview

Unit 1: Introduction to Python

10 Lectures

Structure of a Python Program, Elements of Python, Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables

Unit 2: Flow control and Functions

10 Lectures

Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution, Flow Control Statements, Importing Modules, Ending a Program Early with `sys.exit()`, `def` Statements with Parameters, Return Values and `return` Statements, The `None` Value, Keyword Arguments and `print()`, Local and Global Scope, The global Statement, Exception Handling.

Unit 3: List, Dictionary, String and Tuples

10 Lectures

String, String functions, Manipulating Strings, Lists: Creating Lists; Operations on Lists; Built-in Functions on Lists; Implementation of Stacks and Queues using Lists; Nested Lists. Dictionaries: Creating Dictionaries; Operations on Dictionaries; Built-in Functions on Dictionaries; Dictionary Methods; Populating and Traversing Dictionaries. Tuples and Sets: Creating Tuples; Operations on Tuples; Built-in Functions on Tuples; Tuple Methods; Creating Sets; Operations on Sets; Built-in Functions on Sets; Set Methods.

Suggested Readings

1. T. Budd, Exploring Python, TMH, 1st Ed, 2011
2. PythonTutorial/Documentation www.python.org 2015
3. Allen Downey, Jeffrey Elkner, Chris Meyers , How to think like a computer scientist : learning with Python, Freely available online. 2012
4. <http://docs.python.org/3/tutorial/index.html>
5. <http://interactivepython.org/courselib/static/pythonds>
6. <http://www.ibiblio.org/g2swap/byteofpython/read/>

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. WAP to print Hello World!
2. WAP to calculate total marks, percentage and grade of a student. Marks obtained in each of the three subjects are to be input by the user. Assign grades according to the following criteria:
 - a. Grade A: Percentage ≥ 80
 - b. Grade B: Percentage ≥ 70 and < 80
 - c. Grade C: Percentage ≥ 60 and < 70
 - d. Grade D: Percentage ≥ 40 and < 60
 - e. Grade E: Percentage < 40
3. Write a menu-driven program, using user-defined functions to find the area of rectangle, square, circle and triangle by accepting suitable input parameters from user.
4. WAP to display the first n terms of Fibonacci series.
5. WAP to find factorial of the given number.
6. WAP to implement the use of arrays in Python.
7. WAP to implement String Manipulation in python in Python.
8. WAP to find sum of the following series for n terms: $1 - 2/2! + 3/3! - \dots - n/n!$
9. WAP to generate a random number
10. Write a menu driven program to convert the given temperature from Fahrenheit to Celsius and vice versa depending upon users' choice.
11. WAP to find the area of a circle.
12. WAP to the largest among the three numbers.

MS POWERPOINT

Prerequisite(s) and/or Note(s):

- (1) Basic knowledge of Computer.
- (2) Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives Knowledge***Acquired:***

- (1) Presentation design principles understanding.
- (2) MS PowerPoint interface familiarity.
- (3) Slide layout and formatting comprehension.

Skills Gained:

- (1) Slide creation and editing proficiency.
- (2) Visual content insertion capability.
- (3) Animation and transition application skill.

Competency Developed:

- (1) Effective presentation delivery competency.
- (2) Audience engagement techniques mastery.
- (3) Time management during presentations efficiency.

Syllabus Overview

Unit 1: Creating and Managing Presentations

15 Lectures

Create a Presentation: Insert and Format Slides, Modify Slides, Handouts, and Notes, Change Presentation Options and Views Configure a Presentation for Print, Configure and Present a Slide Show, Insert and Format Text: Insert and Format Shapes and Text Boxes, Insert and Format Images, Order and Group Objects.

Unit 2: Tables, Charts, SmartArt, and Media

7 Lectures

Insert and Format Tables: Insert and Format Charts, Insert and Format SmartArt graphics, Insert and Manage Media.

Unit 3: Transitions and Animations

8 Lectures

Apply Slide Transitions, Animate Slide Content, Set Timing for Transitions and Animations, Working with bullets and numbering, Working with different views, working with slide Master, Slide show option.

Suggested Readings

1. Microsoft power point 2019 ,learning the basics by Eric Stockson
2. Microsoft power point 2019 for beginners by J.Davidson.
3. Marquee series Microsoft power point 2019 by Audrey Roggenkamp & Lan Rutkowski ,Nita Rutkosky

**Course-SEC
Paper:**

**Paper Code- POOASEC106L
MS Power Point (Lab)**

Credits-1

Lab hours-30/Week-2

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

- (1) Creating a Title Slide
- (2) Creating Slides Using Layouts
- (3) Create a presentation that consists of 5 slides and save your Presentation in desktop.
- (4) Demonstrate slide transitions and animation
- (5) Insert slide number, slide date, slide header and footer
- (6) Demonstrate rehearse time.
- (7) Demonstrate master slide.
- (8) Demonstrate different views-Slide sorter, normal, note page.
- (9) Create a minimum of 5 slides with a slide theme appropriate for the topic and utilize WordArt on your title slide for the tile.
- (10) Apply a different picture style or picture effect to each picture in your presentation and select a transition and apply it to all slides.

Course-MAJOR

Paper Code- BCAPMAJ203

Credits-3

Lectures-45 /Week-3

DIGITAL ELECTRONICS

Prerequisite(s) and/or Note(s):

(1) High school Physics.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances.
However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:

Knowledge acquired:

- (1) Basic knowledge of digital logic and digital circuits,
- (2) Overall idea about how computers function and the internal building blocks of a computer.
- (3) Knowledge about how operations are performed in a computer
- (4) A thorough understanding of the fundamental concepts and techniques used in digital electronics.

Skills gained:

- (1) Application of the knowledge of digital logic to understand digital electronics circuits.
- (2) The ability to understand, analyze and design various combinational and sequential circuits.
- (3) To understand and examine the structure of various number systems and its application in digital design.

Competency Developed:

- (1) Ability to identify basic requirements for a design application and propose a cost effective solution.
- (2) The ability to identify and prevent various hazards and timing problems in a digital design.
- (3) Ability and skill to develop/build, and troubleshoot digital circuits.

Syllabus Overview

Unit 1 : Number Systems

10 Lectures

Binary, octal, hexadecimal and decimal number systems and their inter conversion, BCD numbers (8421-2421), Gray code, excess-3 code, code conversion, ASCII, EBCDIC codes, their advantages and disadvantage, Binary addition and subtraction, Negative number representation: Sign magnitude, 1's, 2's Complement. signed and unsigned binary numbers, Fixed and floating-point representation.

Unit 2 : Gates in Digital Design

10 Lectures

AND, OR, NOT Gates and their Truth Tables, NOR, NAND & XOR gates, Boolean algebra, Basic Boolean Laws, De-morgan's theorem, Boolean function and their truth tables, Minimization techniques, K-Map for 2, 3 and 4 variables, Sum of Product & Product of Sum, Don't care conditions.

Unit 3 : Logic Families

7 Lectures

Introduction to digital logic family such as RTL, DTL, TTL, ECL, CMOS, IIR, HTL etc., their comparative study, Basic circuit, performance characteristics.

Unit 4 : Combinational Logic

8 Lectures

Half adder, Full adder, parallel adder, half subtractor, full subtractor, 4-bit binary adder cum subtractor, Multiplexer, Demultiplexer, Decoder, BCD to seven segment Decoder, Encoders.

Unit 5 : Sequential Logic

10 Lectures

Set-reset latches, D-flip-flop, R-S flip-flop, J-K flip-flop, Master slave flip-flop, edge triggered flip-flop, T flipflop, Synchronous/Asynchronous counter, Up/down synchronous counter, Ripple Counter, Applications of counter, Serial in/Serial out shift register, Parallel in/Serial out shift register, Serial in/parallel out shift register, parallel in/ parallel out shift register, Bi-directional register, Applications of register.

Suggested Readings

1. Rajaraman V. & Radhakrishnan, An Introduction To Digital Computer Design, PHI.
2. Malvino & Leach, Digital Principles & Applications, TMH
3. S. Salivahanan, S. Arivazhagan, Digital Circuits and Design, Oxford University Press

Course-MAJOR

Paper Code- BCAPMAJ203L

Credits-1

Lab hours-30 /Week-2

Paper:

Digital Electronics (Lab)

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Design of a XOR gate using basic gates.
2. Design of a 8 x 1 MUX using basic gates.
3. Design of a Half Adder using basic gates.
4. Design of a Full Subtractor using basic gates.
5. Design of a 2-to-4 Decoder using basic gates.

JAVA PROGRAMMING***Prerequisite(s) and/or Note(s):***

- (1) High school mathematics.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives:***Knowledge acquired:***

- (1) Understanding of Object-Oriented Concepts: Students will acquire knowledge of fundamental ObjectOriented Programming (OOP) concepts such as classes, objects, inheritance, polymorphism, and encapsulation. They'll grasp the theoretical underpinnings of these concepts and their practical applications in software development.
- (2) Java Syntax and Language Features: Through hands-on coding exercises and projects, students will become proficient in Java syntax, learning about data types, control flow structures, and exception handling. They'll understand how to write Java programs that follow best practices and adhere to industry standards.
- (3) Software Design Principles: The course will equip students with knowledge of software design principles like SOLID principles, design patterns, and anti-patterns. They'll learn how to architect well-structured, maintainable, and extensible software systems using object-oriented design principles.

Skills gained:

- (1) Programming Proficiency: Students will develop practical programming skills in Java, including the ability to write, compile, and execute Java programs independently. They'll gain confidence in coding by solving progressively challenging programming problems and implementing real-world applications.
- (2) Debugging and Troubleshooting: Through debugging exercises and code reviews, students will learn how to identify and fix errors in Java code effectively. They'll develop skills in using debugging tools and techniques to diagnose and resolve software issues efficiently.

Competency Developed:

- (1) Problem-Solving Skills: Students will enhance their problem-solving abilities by applying object-oriented principles to solve complex programming problems. They'll learn how to break down problems into smaller, manageable components and devise elegant solutions using OOP concepts.
- (2) Critical Thinking and Analysis: The course will foster students' ability to critically evaluate software designs and code implementations. They'll learn to analyze trade-offs, identify design flaws, and propose alternative solutions, honing their critical thinking skills essential for software development.
- (3) Software Development Practices: By working on practical projects, students will develop competency in software development practices such as unit testing, code documentation, and code refactoring. They'll understand the importance of writing clean, readable, and maintainable code, preparing them for careers in software engineering or further academic pursuits in computer science.

Syllabus Overview

Unit 1 : Introduction to Java

7 Lectures

Java Architecture and Features, Understanding the semantic and syntax differences between C++ and Java, Compiling and Executing a Java Program, Variables, Constants, Keywords Data Types, Operators (Arithmetic, Logical and Bitwise) and Expressions, Comments, Doing Basic Program Output, Decision Making Constructs (conditional statements and loops) and Nesting, Java Methods (Defining, Scope, Passing and Returning Arguments, Type Conversion and Type and Checking, Built-in Java Class Methods).

Unit 2 : Arrays, Strings and I/O

7 Lectures

Creating & Using Arrays (One Dimension and Multi-dimensional), Referencing Arrays Dynamically, Java Strings: The Java String class, Creating & Using String Objects, Manipulating Strings, String Immutability & Equality, Passing Strings To & From Methods, String Buffer Classes. Simple I/O using System.out and the Scanner class, Byte and Character streams, Reading/Writing from console and files.

Unit 3 : Object -Oriented Programming Overview

5 Lectures

Principles of Object-Oriented Programming, Defining & Using Classes, Controlling Access to Class Members, Class Constructors, Method Overloading, Class Variables & Methods, Objects as parameters, final classes, Object class, Garbage Collection.

Unit 4 : Inheritance, Interfaces, Packages, Enumerations, Autoboxing and Metadata

10 Lectures

Inheritance: (Single Level and Multilevel, Method Overriding, Dynamic Method Dispatch, Abstract Classes), Interfaces and Packages, Extending interfaces and packages, Package and Class Visibility, Using Standard Java Packages (util, lang, io, net), Wrapper Classes, Autoboxing/Unboxing, Enumerations and Metadata.

Unit 5 : Exception Handling, Threading, Networking and Database Connectivity

6 Lectures

Exception types, uncaught exceptions, throw, built-in exceptions, Creating your own exceptions; Multithreading: The Thread class and Runnable interface, creating single and multiple threads, Thread prioritization, synchronization and communication, suspending/resuming threads. Using java.net package, Overview of TCP/IP and Datagram programming. Accessing and manipulating databases using JDBC.

Unit 6 : Applets and Event Handling

10 Lectures

Java Applets: Introduction to Applets, Writing Java Applets, Working with Graphics, Incorporating Images & Sounds. Event Handling Mechanisms, Listener Interfaces, Adapter and Inner Classes. The design and Implementation of GUIs using the AWT controls, Swing components of Java Foundation Classes such as labels, buttons, textfields, layout managers, menus, events and listeners; Graphic objects for drawing figures such as lines, rectangles, ovals, using different fonts. Overview of servlets.

Suggested Readings

- 1.Ken Arnold, James Gosling, David Homes, "The Java Programming Language", 4th Edition, 2005.
- 2.James Gosling, Bill Joy, Guy L Steele Jr, GiladBracha, Alex Buckley"The Java Language Specification, Java SE 8 Edition (Java Series)", Published by Addison Wesley, 2014.
- 3.Joshua Bloch, "Effective Java" 2nd Edition,Publisher: Addison-Wesley, 2008.
- 4.Bruce Eckel, "Thinking in Java", 3rd Edition, PHI, 2002.
- 5.E. Balaguruswamy, "Programming with Java", 4th Edition, McGraw Hill.2009.
- 6.Paul Deitel, Harvey Deitel, "Java: How to Program", 10th Edition, Prentice Hall, 2011

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. To find the sum of any number of integers entered as command line arguments
2. To find the factorial of a given number
3. To learn use of single dimensional array by defining the array dynamically.
4. To learn use of “.length” in case of a two dimensional array
5. To convert a decimal to binary number
6. To check if a number is prime or not, by taking the number as input from the keyboard
7. To find the sum of any number of integers interactively, i.e., entering every number from the keyboard, whereas the total number of integers is given as a command line argument

Course-SEC2	Paper Code- POOBSEC218	Credits-2	Lectures-30 /Week-3
-------------	------------------------	-----------	---------------------

CYBER SECURITY

Prerequisite(s) and/or Note(s):

(1) Anyone interested in learning about the subject.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives Knowledge

Acquired

- (1) Cyber threats landscape understanding.
- (2) Principles of cryptography comprehension.
- (3) Network security protocols familiarity.

Skills Gained

- (1) Ethical hacking techniques application.
- (2) Security assessment tools utilization.
- (3) Incident response plan development.

Competency Developed

- (1) Risk assessment proficiency.
- (2) Security policy formulation expertise.
- (3) Communication of security concepts clarity

Syllabus Overview

Unit 1: Introduction of Cyber Security

8 Lectures

A Brief History of the Internet, Computer Crime, Defining Cyber Security and Cyberspace, Communication and web technology, Internet, World wide web, regulation of cyberspace, concept of cyber security, Issues and challenges of cyber security. Cyber security terminologies: Security, Attacks, risk, vulnerability, exploit, hacker, Computer Criminals, Cyber warfare, Security Services, Security Mechanisms, Case Studies.

Unit 2: Cyber crimes

9 Lectures

Cyber crimes targeting Computer systems and Mobiles - spyware, logic bombs, DoS, virus, Trojans, ransomware, data breach, Online scams and frauds - email scams, Phishing, Online job fraud, Online sextortion, Debit/ credit card fraud, Online payment fraud, website defacement, Cyber espionage, Darknet - illegal trades, drug trafficking, human trafficking, Social Media Scams & Frauds- impersonation, identity theft, misinformation, fake news, cyber crime against persons - cyber grooming, child pornography, cyber stalking, Cyber bullying, Social Engineering attacks, Crime reporting procedure, Case studies.

Unit 3: Unit 3: Digital Devices Security, Tools & Technologies for Cyber Security

7 Lectures

End Point device and Mobile phone security, Password policy, Security management, Data backup, Downloading and management of third party software, Device security policy, Cyber Security best practices, Significance of host firewall and Ant-virus, Wi-Fi security, Configuration of basic security policy and permissions.

Unit 4: Cyber law and Investigation

6 Lectures

Cyber crime and legal landscape around the world, IT Act, 2000 and its amendments. Limitations of IT Act, 2000. Cyber crime and punishments, Cyber Laws and Legal and ethical aspects related to new technologies- AI/ML, IoT, Blockchain, Darknet and Social media, Cyber Laws of other countries, Case Studies.

Suggested Readings

1. "Cybersecurity for Dummies" by Chey Cobb.
2. "Computer Hacking Beginners Guide" by Alan T. Norman
3. "Hacking: Computer Hacking, Security Testing, Penetration Testing, and Basic Security" by John Slawo.
4. Bharat Bhaskar, *Electronic Commerce: Framework, Technology and Application, 4th Ed.*, McGraw Hill Education
5. Security in Computing, 3rd Edition, Charles P. Pfleeger & Shari Lawrence Pfleeger, PHI
6. Cryptography and Network Security, A. Kahate, TMH

Paper:**Cyber Security (Lab)**

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

1. Write the steps for creating the User account, setting permissions and protecting your files with password.
2. Write the steps for installation of software from Open source Mode and Paid subscription mode.
3. Write the steps to establish peer to peer network connection using two systems in a LAN.
4. Write the steps in providing network security and to set Firewall Security in windows.
5. Prepare a Case study on Ransomware attacks.
6. Prepare a case study on Social Media Crime.
7. Write the steps to prevent the denial of Service attacks.
8. What is Malware? Write the steps to remove the malware from your PC.
9. List out the various Mobile security apps. Write the steps to install and use, one of the mobile security app.
10. Write the steps to analyze the E-Mail Application's security vulnerabilities.
11. Write the steps to read Email Headers and identify them as SPAM.
12. Write the steps to check whether the website is legitimate or not.
13. Write the steps to prevent the denial of Service attacks
14. Demonstrate the use of Network tools: ping, ipconfig, ifconfig, tracert, arp, netstat, whois
15. Demonstrate sending of a protected word document.

HTML PROGRAMMING

Prerequisite(s) and/or Note(s):

(1) Anyone interested in learning about the subject.

1. Experience with any text editor like notepad,
2. How to create directories and files on your computer.
3. How to navigate through different directories.
4. How to type content in a file and save them on a computer.
5. Understanding about images in different formats like JPEG, PNG format.

Note(s): Syllabus changes yearly and may be modified during the term itself, depending on the circumstances. However, students will be evaluated only on the basis of topics covered in the course.

Course Objectives Knowledge

Acquired:

- (1) Understanding of web page structure.
- (2) Notepad, browser familiarity.
- (3) Knowledge about building web pages.

Skills Gained:

- (1) Proficiency in web page development.
- (2) Visual content insertion capability.
- (3) Creation of forms, tables.

Competency Developed:

- (1) Effective web page development competency.
- (2) Understanding the core concepts of web development and how web pages are constructed.
- (3) Ability to structure and organize content on a web page effectively.
- (4) Skills in creating forms to collect and manage user input.

Syllabus Overview

Unit 1: HTML Overview

5 Lectures

Introduction to web page designing using HTML: create and save an HTML document, access a web page using a web browser, Basic HTML document, HTML document structure.

Unit 2: HTML basic tags

10 Lectures

Heading tag, Paragraph tag, Line break tag, Centering content, Horizontal lines, Preserve formatting,

HTML tags: html, head, title, body, (attributes: text, background, bgcolor, link, vlink, alink), br (break), hr (horizontal rule), inserting comments, h1..h6 (heading), p (paragraph), b (bold), i (italics), u (underline), ul (unordered list), ol (ordered list), and li (list item). Description lists: dl, dt and dd. Attributes of ol (start, type), ul (type).

Unit 3: HTML Formatting

5 Lectures

Bold Text, Italic Text, Underlined Text, Strike Text, Monospaced Font, Superscript Text, Subscript Text, Inserted Text, Deleted Text, Larger Text, Smaller Text, Grouping Content

Unit 4: HTML – Images, Tables, Forms

10 Lectures

Insert images: img (attributes: src, width, height, alt), sup (super script), sub(subscript). HTML Forms: Textbox, radio buttons, checkbox, password, list, combo box. Embed audio and video in a HTML page. Create a table using the tags: table, tr, th, td, rowspan, colspan

Suggested Readings

1. HTML 5.0 for Beginners, Vinod KumarMurugesan
2. Learn HTML in Easy way, Ritesh Kumar.
3. A Complete Reference, HTML and CSS, Thomas A. Powell.
4. Web Enabled Commercial Application Development Using HTML, Javascript, DHMTL And PhP (4Th Revised Edition), Ivan Bayross.

Course-SEC2	Paper Code- POOBSEC219L	Credits-1	Lab hours-30/Week-2
Paper:	HTML Programming (Lab)		

Students are advised to do laboratory/practical practice not limited to, but including the following types of problems:

- (1) Creating an HTML document for displaying a web page with the following tags:
 - a. Bold
 - b. Italic
 - c. Alignment
 - d. Paragraph
 - e. Underline
 - f. Text colour
 - g. Background colour
 - h. Heading
 - i. Line break
 - j. pre
- (2) Design a web page of your CV with headings as objective, educational qualification, achievement, strength, hobbies and personal details.
 Apply the following specifications:
 - a. Set any light colour as page background
 - b. Bold and underline all headings
 - c. Insert your image on the left side of the web page
 - d. Use heading tag to specify the heading
 - e. After every heading is over put a horizontal line
 - f. Use pre tag for educational qualification.
- (3) HTML program to create nested lists
- (4) HTML program to create a form to take input from user and display it
- (5) HTML program demonstrating use of tables.