

Course Code: BCSA103C	Course Title : Principles of Programming using C	Credits:	3
Hours/Week (L:T:P) : 2:0:2		CIE Marks :	50
Total Hours of Pedagogy (Theory+Lab): 40 hours		SEE Marks	50
Course Type: Theory/Practical/Integrated			

Course Objectives:

1. Elucidate the basic architecture and functionalities of a Computer.
2. Apply programming constructs of C language to solve the real-world problems.
3. Explore user-defined data structures like arrays, structures and pointers in implementing solutions to problems.
4. Design and Develop Solutions to problems using structure dprogramming constructs such as functions and procedures

Module-1

6 Hrs.

Introduction to C: Introduction to computers, input and output devices, designing efficient programs. Introduction to C, Structure of C program, Files use dina C program, Compilers, Compiling and executing C programs, variables, constants, Input/output statements in C,

Text book:Chapter1.1-1.9,2.1-2.2,8.1- 8.6,9.1-9.14

Module-2

6 Hrs.

Operators in C, Type conversion and type casting.

Decision control and Looping statements: Introduction to decision control, Conditional branching statements, iterative statements, nested loops, break and continue statements, go to statement.

Textbook:Chapter9.15-9.16, 10.1-10.6

Module-3

6 Hrs.

Functions: Introduction using functions, Function definition, function declaration, function call, return statement, passing parameters to functions, scope of variables, storage classes, recursive functions.

Arrays: Declaration of arrays, accessing the elements of an array, storing values in arrays, Operations onarrays, Passing arrays to functions, two dimensional arrays, operations on two-dimensional arrays, two-dimensional arraysto functions.

Textbook: Chapter11.1-11.10,12.1-12.10,12.12

Module-4

6 Hrs.

Strings and Pointers: Introduction ,string taxonomy, operations on strings, Miscellaneous string and character functions, arrays of strings. Pointers: Introduction to pointers, declaring pointer variables, Types of pointers,Passing arguments to functions using pointers.

Textbook:Chapter13.1-13.6,14-14.7

Module-5

6 Hrs.

Structure,Union,and Enumerated DataType: Introduction, structures and functions, Unions, unions inside structures, Enumerated data type.

Files: Introduction to files,using files in C,reading and writing data files, Detecting end of file.

Practical Module

1. Simulation of a Simple Calculator.
2. Compute the roots of a quadratic equation by accepting the coefficients. Print appropriate messages.
3. An electricity board charges the following rates for the use of electricity: for the first 200 units 80 paise per unit: for the next 100 units 90 paise per unit: beyond 300units Rs 1perunit. All users are charged aminimum of Rs.100 as meter charge. If the total amount is more than Rs 400, then an additional surcharge of 15% of total amount is charged.Write a program to read the nameof the user,number of units consumed and print out the charges.
4. Write a C Program to display the following by reading the number of rows as input,

```
      1
    1  2  1
  12  3 21
123  4 32 1
```

nth row

5. Implement Binary Search on Integers.
6. Implement Matrix multiplication and validate the rules of multiplication.
7. Compute $\sin(x)/\cos(x)$ using Taylor series approximation. Compare your result with the built-in library function.Print both the results with appropriateinferences.
8. Sort the given set of Nnumbers using Bubblesort.
9. Write functions to implement string operations such as compare, concatenate, and find string length. Use the parameter passing techniques.
10. Implement structures to read, write and compute average- marks of the students, list the students scoring above and below the average marks for a classof N students.
11. Develop a program using pointers to compute the sum, mean and standard deviation of all elements stored in an arrayof N real numbers.
12. Write a C program to copy a text file to another, read both the input file name and target file name.

Suggested Learning resources

Textbooks:

1. Computer fundamentals and programming in C, “Reema Thareja”, Oxford University, Second edition, 2017.

Reference Books:

1. E.Balaguruswamy, Programming in ANSIC, 7th Edition, Tata Mc Graw-Hill.
2. BrianW.Kernighan and Dennis M. Ritchie, The‘C’ Programming Language, Prentice Hall of India.
3. Kernighan and Ritchie, C Programming Language, 2nd Edition, 1988, 49th Reprint, 2017.
4. Wesley J. Chun, A Structured Programming approach using C, Pearson Education India, 3rd Edition, 2015.

5. Stephen Kochan, Programming in C, 4th Edition, 2014.

6. B. S. Anami, S. A. Angadi & S. S. Manvi, Computer Concepts and C programming-A Holistic approach to learning C, 2nd Edition, PHI, 2010.

Weblinks and Video Lectures (e-Resources):

1. elearning.vtu.ac.in/econtent/courses/video/BS/15PCD23.html
2. <https://nptel.ac.in/courses/106/105/106105171/MOOCcoursescanbeadoptedformoreclarityinunderstandingthetopicsandveritiesofproblemsolvingmethods>.
3. <https://tinyurl.com/4xmrexre>

Course Outcomes:

CO1. Elucidate the basic architecture and functionalities of a computer and also recognize the hardware parts.

CO 2. Apply programming constructs of C language to solve the real world problem.

CO 3. Explore the usage of arrays and functions in implementing solutions to problems

CO 4. Illustrate the modular programming approach using string and pointers.

CO5. Design and Develop Solutions to problems using modular programming approach using structures and files.

Course Outcomes	Programme Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	2			2							
CO2		1			2							2
CO3		1										2
CO4		2										2
CO5		3	3	2								2

BCSA104C	Course Title: Introduction to C Programming	Credits:	03
Hours/Week (L:T:P) : 2:0:2		CIE Marks :	50
Total Hours of Pedagogy (Theory+Lab): 40		SEE Marks :	50
Course Type: Theory/Practical/Integrated: Integrated			

Course Objectives:

CLO1.Elucidate the basic architecture and functionalities of a Computer
CLO2.Apply programming constructs of C language to solve the real-world problems
CLO3.Explore user-defined data structures like arrays, and structures in implementing solutions to problems
CLO 4. Design and Develop Solutions to problems using modular programming constructs such as functions and procedures

Module-1	6 Hrs.
Introduction to C: Introduction to computers, input and output devices, designing efficient programs. Introduction to C, Structure of C program, Files used in a C program, Compilers, Compiling and executing C programs, variables, constants, Input/output statements in C Textbook:Chapter1.1-1.9,2.1-2.2,8.1–8.6,9.1-9.14	
Module-2	6 Hrs.
Operators in C, Type conversion and typecasting. Decision control and Looping statements: Introduction to decisioncontrol, Conditional branchingstatements, iterative statements, nested loops, break and continue statements, goto statement. Textbook:Chapter9.15-9.16,10.1-10.6	
Module-3	6 Hrs.
Functions Introduction using functions, Function definition, function declaration, function call, return statement, passing parameters to functions, scope of variables, storage classes, recursive functions. Arrays: Declaration of arrays, accessing the elements of an array, storing value sin arrays, Operations on arrays, Passing arrays to functions, Textbook:Chapter11.1-11.13,12.1-12.6	
Module-4	6 Hrs.
Two dimensional arrays, operations on two-dimensional arrays, two-dimensional arrays to functions, multi dimensional arrays. Applications of arrays: Applications of arrays, case study with sorting techniques. Introduction to strings: Reading strings, writing strings, summary of functions used to read and write characters. Textbook:Chapter12.7-12.12	
Module-5	6 Hrs.
Strings: String-handling functions, operations on strings, Miscellaneous string and character functions, arrays of strings. Structures: Introduction to structures, Defining a structure, declaring structure variables, Initialization, Accessing structure members. Arrays of structure, Textbook:Chapter13.1-13.6,14.1-14.3,15.1	
Practical Module	
1. C program to find mechanical energy of a particle using $e = mgh + \frac{1}{2} mv^2$. 2. C program to convert kilometers into meters and centimeters. 3. C program to check the given character is lowercase or uppercase or special character. 4. To find reverse of a number and check for palindrome 5. To print prime numbers between two numbers	

6. To compute factorial of an input number using recursive function
7. Sort the given set of n numbers using bubble sort.
8. Implement matrix multiplication and validate the rules of multiplication.
9. Write functions to implement string operations such as compare, concatenate, string length. Convince the parameter passing techniques.
10. Use structures to read, write and compute average- marks of N students. Also, list the students scoring above and below the average marks.

Suggested Learning resources

Textbooks

1. Computer fundamentals and programming in C, “Reema Thareja”, Oxford University, Second edition, 2017.

ReferenceBooks:

1. E. Balaguruswamy, Programming in ANSIC, 7th Edition, TataMcGraw-Hill.
2. Brian W.Kernighan and Dennis M.Ritchie, The ‘C’ Programming Language, Prentice Hall of India.

Course Outcomes:

CO1:	Elucidate the basic architecture and functionalities of a computer and also recognize the hardware parts
CO2:	Apply programming constructs of C language to solve the real world problem
CO3:	Explore user-defined data structures like arrays in implementing solutions to problems like searching and sorting
CO4:	Explore user-defined data structures like structures in implementing solutions
CO5:	Design and Develop Solutions to problems using modular programming constructs using functions

CO and PO Mapping

Course Outcomes	Programme Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3											
CO2	2	3	2	1	2	2						1
CO3	2	3	2	1	2	2						1
CO4	1	2	3	1	2	1						1
CO5	1	2	3	2	2	1						1

Course Code:BCSA105N	Introduction to Web Programming	Credits :	03
Hours/Week (L:T:P) : 2:0:2		CIE Marks :	50
Total Hours of Pedagogy (Theory + Lab): 40hrs		SEE Marks :	50
Course Type: Theory/Practical/Integrated			

Course Objectives:

- To use the syntax and semantics of HTML and XHTML
- To develop different parts of a webpage
- To understand how CSS can enhance the design of a webpage.
- To create and apply CSS styling to a webpage
- To get familiarity with the JavaScript language and understand Document Object Model handling of Java Script

Module-1	06 Hrs.
Fundamentals of Web: Introduction to Internet, World Wide Web, Web Browser, Web Server, Uniform Resource Locators, MIME, HTTP. Hypertext Markup Language: Basic HTML, The Document Body, Text, Hyperlinks, Adding More Formatting, Lists: Ordered Lists, Unordered Lists, and Definition Lists.	
Module-2	06 Hrs.
HTML: Tables, Using Color and Images, Multimedia Objects, Frames, Forms, The HTML Document Head in Detail: Document type declaration. Hello HTML5, Loose Syntax Returns, HTML5: Embracing the Reality of Web Markup, Presentational Markup Removed and Redefined, HTML5 Document Structure Changes, Adding Semantics.	
Module-3	06 Hrs.
Cascading Style Sheets(CSS) Introduction, CSS Overview, CSS Rules, Example with Type Selectors and the Universal Selector, CSS Syntax and Style, Class Selectors, ID Selectors, span and div Elements, Cascading, style Attribute, style Container, External CSS Files, CSS Properties, Color Properties, RGB Values for Color, Font Properties.	
Module-4	06 Hrs.
Tables and CSS, Links and Images Table Elements, Formatting a Data Table: Borders, Alignment, and Padding, CSS Structural Pseudo- Class Selectors, thead and tbody Elements, Cell Spanning, Web Accessibility, CSS display Property with Table Values, a Element, Relative URLs, Navigation Within a Web Page, CSS for Links, Bitmap Image Formats: GIF, JPEG, PNG.	
Module-5	06 Hrs.
Introduction to JavaScript: Functions, DOM, Forms, and Event Handlers JavaScript: An Introduction to JavaScript, What is Dynamic HTML?, JavaScript, JavaScript The Basics, Variables, String Manipulation, Mathematical Functions, Statements, Operators, Arrays, Functions. Objects in JavaScript: Data and Objects in JavaScript, Regular Expressions, Exception Handling, Built-in Objects, Cookies, Events.	

Practical Module

1. To create a simple html file to demonstrate the use of different tags.
Create an html page named as “Basic_Html_Tags.html” and add the following tags
 - a) Different heading tags (h1 to h6)
 - b) Paragraph
 - c) Horizontal line
 - d) Line Break
 - e) Pre tag
 - f) Different Logical Style (,<u>,<Sup>,<sub>)
2. Create a HTML code to display the following web page using list.

Learning Web Development

- I. Background Skills
 - A. Unix Commands
 - B. Vim Text Editor
- II. HTML
 - A. Minimal Page
 - B. Headings
 - C. Tags
 - D. Lists
 - i. Unordered
 - ii. Ordered
 - iii. Definition
 - iv. Nested
 - E. Links
 - i. Absolute
 - ii. Relative
 - F. Images

3. Create an html page named as “Table.html” to display the following table.

NAME	SUBJECT	MARKS
Hillary	Advanced Web	75
	Opearating System	60
Lary	Advanced Web	80
	Opearating System	75
Total Average:72.5		

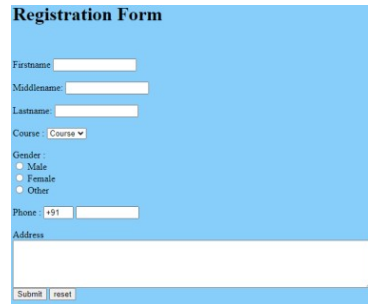
4. Create an internal style sheet to define the following CSS properties for element(s) on your page:
 - a. text-decoration
 - b. text-align
 - c. font-size
 - d. font-family
 - e. font-weight
5. Create an external style sheet to define the following CSS properties for element(s) on your page:
 - a. background-color,color,margin-xxx(left,right,top,orbottom),padding
 - b. border-style,border-color,border-width
6. To create HTML code to insert image and to add a link to a web page.
7. Write a HTML code to perform the following CSS properties
 - a. Text Properties:text-color,text-align,text-decoration,text-transformation
 - b. Font Properties:font-style,font-size,font-family

8. Create the following table using CSS properties.

Roll No	Name	Team
1001	John	Red
1002	Peter	Blue
1003	Henry	Green

9. Create a HTML code for simple registration form which include

- i. First name
- ii. Middle name
- iii. Last name
- iv. Course with option button
- v. Gender with radio button
- vi. Phone No
- vii. Address
- viii. Submit and reset button.



10. Using HTML and Javascript, design a webpage to calculate payroll of an employee. Read Employee details such as Emp_Id, Name, Designation, Department and Basic Salary. Read Deduction in percentage such as PF, LIC. Read Allowance in percentage such as DA and HRA. Calculate Gross and Net Salary of an Employee.

Suggested Learning resources

Textbooks:

1. Thomas A.Powell, THE COMPLETE REFERENCE HTML&CSS, Fifth Edition, Tata Mc Graw Hill, 2017.

Reference Books:

1. Chris Bates, WEB PROGRAMMING(Second Edition),Wiley,2006.
2. John Dean, WEB PROGRAMMING with HTML5 CSS and JavaScript, First Edition, Jones & BartlettLearning, 2019.
3. Robert Cebesta, Programming the World WideWeb,4th Edition, Pearson,2008.
4. Eric Lad, JimO. Donnel, Using HTML4,XML and Java,PHI,1999

Course Outcomes:

CO 1: Analyze historical context and justification for HTML

CO 2: Develop HTML5 documents and adding various semantic markup tags.

CO 3: Learn various attributes, values and types of CSS.

CO 4: Build a web page using links and images.

CO 5: Develop web applications using JavaScript.

Course Outcomes	Programme Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	-	3	3	-	-	-	-	-	-	-	-	3
CO2	-	2		2	-	-	-	-	-	-	-	3
CO3	-	-	2	-	-	-	-	-	-	-	-	3
CO4	3	-	-	2	-	-	-	-	-	-	-	3
CO5	-	3	2	-	1	-	-	-	-	-	-	-

Course Code:BCSB105N	Course Title: Introduction to Python Programming	Credits:	03
Hours/Week (L:T:P) : 2:0:2:0		CIE Marks :	50
Total Hours of Pedagogy (Theory+Lab): 40		SEE Marks :	50
Course Type: Theory/Practical/Integrated: Integrated			

Course Objectives:

- Learn the syntax and semantics of the Python programming language.
- Illustrate the process of structuring the data using lists, tuples and dictionaries
- Appraise the need for working with various file types
- Demonstrate the use of built-in functions to navigate the file system.

Module-1	8 Hrs.
-----------------	---------------

Introduction to Programming: Problem solving using programming, algorithm and flowcharts, examples, Introduction to programming languages, Introduction to Python, keywords, identifiers. variables, basic input and output: printing using the print function, Taking input using the input function

Python Basics: Entering Expressions into the Interactive Shell, The Integer, Floating-Point, and String Data Types, String Concatenation and Replication, Storing Values in Variables, Your First Program, Dissecting Your Program, Flow control: Boolean Values, Comparison Operators, Boolean Operators, Mixing Boolean and Comparison Operators, Elements of Flow Control, Program Execution

Textbook 1: Chapters 1 – 2

Module-2	8 Hrs.
-----------------	---------------

Flow Control Statements, decisions, loops, terminating control, Importing Modules, Ending a Program Early with sys.exit()

Functions: 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, A Short Program: Guess the Number

Textbook 1: Chapters 2 – 3

Module-3	8 Hrs.
-----------------	---------------

Lists: The List Data Type, Working with Lists, Augmented Assignment Operators, Methods, Example Program: Magic 8 Ball with a List

Tuples: Creating Tuples, accessing tuple elements, counting tuple elements, Iterating through tuple elements, searching elements within tuples, tuple slices, adding multiplying and copying tuples, comparison of tuples and lists, simple programs

Introduction to sets: Creating sets, Accessing set elements, Iterating through set elements

Textbook 1: Chapters 4 – 5

Module-4	8 Hrs.
-----------------	---------------

Dictionaries and Structuring Data: The Dictionary Data Type, Pretty Printing, Using Data Structures to Model Real-World Things,

Manipulating Strings: Working with Strings, Useful String Methods, Project: Password Locker
Project: Adding Bullets to Wiki Markup

Textbook 1: Chapters 5 – 6

Module-5

8 Hrs.

Reading and Writing Files: Introduction to file handling, Types of files, opening and closing files, reading from text files, writing to text files, seeking within files, reading to binary files, writing to binary files, Files and File Paths, The os.path Module, Saving Variables with the shelve Module, Saving Variables with the print.format() Function, compressing files with zip file module

Project: Generating Random Quiz Files, Project: Multiclip board,

Textbook 1: Chapters 8 – 9

Practical Module

1. Develop a program to read the student details like Name, USN, and Marks in three subjects. Display the student details, total marks and percentage with suitable messages.
2. Write a Python program to calculate total expenditure for the following: Given rectangular field of size L x B. It is required to setup circular lawn of radius R at the center of field which costs Rs. 100/ square unit, remaining area is to be tiled which costs Rs 150/ square unit and also it is required to fence the field which costs Rs. 55/ unit length.
3. Develop a program to generate Fibonacci sequence of length (N). Read N from the console.
4. Write a function to calculate factorial of a number. Develop a program to compute binomial coefficient (Given N and R).
5. Read N numbers from the console and create a list. Develop a program to print mean, variance and standard deviation with suitable messages.
6. Read a multi-digit number (as chars) from the console. Develop a program to print the frequency of each digit with suitable message.
7. Write a python program to display all the common characters between two strings. Return -1 if there are no matching characters. (Note: Ignore blank spaces if there are any. Perform case sensitive string comparison wherever necessary.)
8. Given price list: { item1: price1, item2: price2, . . . } and list of items purchased with quantity, write python program to find amount to be paid for purchased items.
Ex: pricelist ; { "pen":10, "Notebook":50, "book":100}
Items_purchased = { "pen":4, "book":3, "Notebook":7, ...}
o/p: amt_to_be_paid = Rs. 690
9. Develop a program to sort the contents of a text file and write the sorted contents into a separate text file. [Hint: Use string methods strip(), len(), list methods sort(), append(), and file methods open(), readlines(), and write()].
10. Develop a program to backing Up a given Folder (Folder in a current working directory) into a ZIP File by using relevant modules and suitable methods

Suggested Learning resources

Text Books

1. Al Sweigart, "Automate the Boring Stuff with Python", 1st Edition, No Starch Press, 2015. (Available under CC-BY-NC-SA license at <https://automatetheboringstuff.com/>) (Chapters 1 to 18, except 12) for lambda functions use this link:

<https://www.learnbyexample.org/python-lambda-function/>

2. Allen B. Downey, "Think Python: How to Think Like a Computer Scientist", 2nd Edition, Green Tea Press, 2015. (Available under CC-BY-NC license at <http://greenteapress.com/thinkpython2/thinkpython2.pdf> (Chapters 13, 15, 16, 17, 18) (Download pdf/html files from the above link)
3. Learning Python, B. Nagesh Rao, Cyberplus publication, 2nd edition
4. Introduction to python programming, Gaowrishankar S., Veena A, CRC press

Course Outcomes:

At the end of the course the student will be able to:

CO1: Demonstrate proficiency in handling loops and creation of functions.

CO2: Identify the methods to create and manipulate lists, tuples and dictionaries.

CO3: Develop programs for string processing and file organization

CO4: Develop simple Python programs for real life applications

CO and PO Mapping

Course Outcomes	Programme Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	2	1	3	1	1							
CO2	1	3	3	1	1							
CO3	1	2	3	1	1							
CO4	2	3	3	1	1	2						1

Course Title:	Introduction to Java Programming		
Course Code:	BCSC105N	CIE Marks	50
Course Type (Theory/Practical /Integrated)	Integrated	SEE Marks	50
		Total Marks	100
Teaching Hours/Week (L:T:P: S)	2:0:2	Exam Hours	03
Total Hours of Pedagogy	40 hours	Credits	03
Course objectives <ul style="list-style-type: none"> ● Learn fundamental features of object oriented language and JAVA ● Set up Java JDK environment to create, debug and run simple Java programs. ● Learn object oriented concepts using programming examples. ● Study the concepts of importing of packages and exception handling mechanism. 			
Teaching-Learning Process These are sample Strategies, which teacher can use to accelerate the attainment of the various course outcomes and make Teaching –Learning more effective <ol style="list-style-type: none"> 1. Use https://pythontutor.com/visualize.html#mode=edit in order to visualize the Java programs 2. Chalk and talk 3. Online demonstration 4. Hands on problem solving 			
Module-1 (8 hours)			
An Overview of Java: Object-Oriented Programming, A First Simple Program, A Second Short Program, Two Control Statements, Using Blocks of Code, Lexical Issues, The Java Class Libraries, Data Types, Variables, and Arrays: Java Is a Strongly Typed Language, The Primitive Types, Integers, Floating-Point Types, Characters, Booleans, A Closer Look at Literals, Variables, Type Conversion and Casting, Automatic Type Promotion in Expressions, Arrays, A Few Words About Strings Text book 1: Ch 2, Ch 3			
Module-2 (8 hours)			
Operators: Arithmetic Operators, The Bitwise Operators, Relational Operators, Boolean Logical Operators, The Assignment Operator, The ? Operator, Operator Precedence, Using Parentheses, Control Statements: Java's Selection Statements, Iteration Statements, Jump Statements. Text book 1: Ch 4, Ch 5			
Module-3 (8 hours)			
Introducing Classes: Class Fundamentals, Declaring Objects, Assigning Object Reference Variables, Introducing Methods, Constructors, The this Keyword, Garbage Collection, The finalize() Method, A Stack Class, A Closer Look at Methods and Classes: Overloading Methods, Using Objects as Parameters, A Closer Look at Argument Passing, Returning Objects, Recursion, Introducing Access Control, Understanding static, Introducing final, Arrays Revisited Text book 1: Ch 6, Ch 7 (7.1-7.9)			
Module-4 (8 hours)			
Inheritance: Inheritance, Using super, Creating a Multilevel Hierarchy, When Constructors Are Called, Method Overriding, Dynamic Method Dispatch, Using Abstract Classes, Using final with Inheritance, The Object Class. Text book 1: Ch 8			

Module-5 (8 hours)

Packages and Interfaces: Packages, Access Protection, Importing Packages, Interfaces, Exception Handling: Exception-Handling Fundamentals, Exception Types, Uncaught Exceptions, Using try and catch, Multiple catch Clauses, Nested try Statements, throw, throws, finally, Java"s Built-in Exceptions, Creating Your Own Exception Subclasses, Chained Exceptions, Using Exceptions.

Text book 1: Ch 9, Ch 10

Course outcome (Course Skill Set)

At the end of the course the student will be able to:

CO1	To explain the features and object oriented concepts in JAVA programming
CO2	To analyse working of bitwise operators in JAVA
CO3	To develop simple programs based on polymorphism and inheritance
CO4	To describe the concepts of importing packages and exception handling mechanism

Programming Assignments

1. Write a JAVA program that prints all real solutions to the quadratic equation $ax^2+bx+c=0$. Read in a, b, c and use the quadratic formula.
2. Write a JAVA program for multiplication of two arrays.
3. Demonstrate the following operations and sign extension with Java programs
(i) << (ii) >> (iii) >>>
4. Write a JAVA program to sort list of elements in ascending and descending order
5. Create a JAVA class called Student with the following details as variables within it.

USN

NAME

BRANCH

PHONE

PERCENTAGE

Write a JAVA program to create n Student objects and print the USN, Name, Branch, Phone, and percentage of these objects with suitable headings.

6. Write a JAVA program demonstrating Method overloading and Constructor overloading.
7. Design a super class called Staff with details as StaffId, Name, Phone, Salary. Extend this class by writing three subclasses namely Teaching (domain, publications), Technical (skills), and Contract (period). Write a JAVA program to read and display at least 3 staff objects of all three categories.
8. Demonstrate dynamic dispatch using abstract class in JAVA.
9. Create two packages P1 and P2. In package P1, create class A, class B inherited from A, class C . In package P2, create class D inherited from class A in package P1 and class E. Demonstrate working of access modifiers (private, public, protected, default) in all these classes using JAVA.
10. Write a JAVA program to read two integers a and b. Compute a/b and print, when b is not zero. Raise an exception when b is equal to zero. Also demonstrate working of Array Index Out Of Bound Exception.

Assessment Details (both CIE and SEE)

Assessment Details (both CIE and SEE)

The weightage of Continuous Internal Evaluation (CIE) is 50% and for Semester End Exam (SEE) is 50%. The minimum passing mark for the CIE is 40% of the maximum marks (20 marks out of 50). The minimum passing mark for the SEE is 35% of the maximum marks (18 marks out of 50). A student shall be deemed to have satisfied the academic requirements and earned the credits allotted to each subject/ course if the student secures not less than 35% (18 Marks out of 50) in the semester-end examination (SEE), and a minimum of 40% (40 marks out of 100) in the sum total of the CIE (Continuous Internal Evaluation) and SEE (Semester End Examination) taken together.

Continuous Internal Evaluation (CIE):

The CIE marks for the theory component of the IC shall be **30 marks** and for the laboratory component **20 Marks**.

CIE for the theory component of the IC

- Three Tests each of 20 Marks; after the completion of the syllabus of 35-40%, 65-70%, and 90-100% respectively.
- Two Assignments/two quizzes/ seminars/one field survey and report presentation/one-course project totaling 20 marks.

Total Marks scored (test + assignments) out of 80 shall be scaled down to **30 marks**

CIE for the practical component of the IC

- On completion of every experiment/program in the laboratory, the students shall be evaluated and marks shall be awarded on the same day. The **15 marks** are for conducting the experiment and preparation of the laboratory record, the other **05 marks shall be for the test** conducted at the end of the semester.
- The CIE marks awarded in the case of the Practical component shall be based on the continuous evaluation of the laboratory report. Each experiment report can be evaluated for 10 marks. Marks of all experiments' write-ups are added and scaled down to 15 marks.
- The laboratory test (**duration 03 hours**) at the end of the 15th week of the semester /after completion of all the experiments (whichever is early) shall be conducted for 50 marks and scaled down to **05 marks**.

Scaled-down marks of write-up evaluations and tests added will be CIE marks for the laboratory component of IC for **20 marks**.

- The minimum marks to be secured in CIE to appear for SEE shall be 12 (40% of maximum marks) in the theory component and 08 (40% of maximum marks) in the practical component. The laboratory component of the IC/IPCC shall be for CIE only.

However, in SEE, the questions from the laboratory component shall be included. The maximum of 05 questions is to be set from the practical component of IC/IPCC, the total marks of all questions should not be more than 25 marks.

The theory component of the IC shall be for both CIE and SEE.

Semester End Examination (SEE):

Theory SEE will be conducted by University as per the scheduled timetable, with common question papers for the subject (**duration 03 hours**)

- The question paper shall be set for 100 marks. The medium of the question paper shall be English).
The duration of SEE is 03 hours.

The question paper will have 10 questions. Two questions per module. Each question is set for 20 marks. The students have to answer 5 full questions, selecting one full question from each module. The student has to answer for 100 marks and marks scored out of 100 shall be proportionally reduced to 30 marks. There will be 2 questions from each module. Each of the two questions under a module (with a maximum of 3 sub-questions), should have a mix of topics under that module.

Suggested Learning Resources:

Books (Title of the Book/Name of the author/Name of the publisher/Edition and Year)

- Herbert Schildt, Java The Complete Reference, 7th Edition, Tata McGraw Hill, 2007.

Web links and Video Lectures (e-Resources):

- https://onlinecourses.nptel.ac.in/noc22_cs47/preview

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Conduct on spot problem solving based on JAVA
- Develop simple GUI interfaces for a computer program to interact with users

COs	POs						
	1	2	3	4	5	6	7
C01	3	3	-	-	2		
C02	-	1	-	-	3		
C03	-	2	-	-			
C04	-	3	-	-			

Level 3- Highly Mapped, Level 2-Moderately Mapped, Level 1-Low Mapped, Level 0- Not Mapped

Course Code: BCSD105N	Introduction to C++ Programming	Credits :	03
Hours/Week (L:T:P) : 2:0:2		CIE Marks :	50
Total Hours of Pedagogy (Theory + Lab): 40 Hours (28T+12 P)		SEE Marks :	50
Course Type: Integrated			

Course Objectives:

1. Understanding about object oriented programming and Gain knowledge about the capability to store information together in an object.
2. Understand the capability of a class to rely upon another class and functions.
3. Understand about constructors which are special type of functions.
4. Create and process data in files using file I/O functions
5. Use the generic programming features of C++ including Exception handling.

Module-1	8 Hrs.
Introduction to Object Oriented Programming: Computer programming background- C++ overview. First C++ Program -Basic C++ syntax, Object Oriented Programming: What is an object, Classes, methods and messages, abstraction and encapsulation, inheritance, abstract classes, polymorphism.	
Textbook 1: Chapter 1(1.1 to 1.8)	
Module-2	8 Hrs.
Functions in C++: Tokens – Keywords – Identifiers and constants – Operators in C++ – Scope resolution operator – Expressions and their types – Special assignment expressions – Function prototyping – Call by reference – Return by reference – Inline functions -Default arguments – Function overloading.	
Textbook 2: Chapter 3(3.2,3.3,3.4,3.13,3.14,3.19, 3.20) , chapter 4(4.3,4.4,4.5,4.6,4.7,4.9)	
Module-3	8 Hrs.
Inheritance & Polymorphism: Derived class Constructors, destructors-Types of Inheritance Defining Derived classes, Single Inheritance, Multiple, Hierarchical Inheritance, Hybrid Inheritance.	
Textbook 2: Chapter 6 (6.2,6.11) chapter 8 (8.1 to,8.8)	
Module-4	8 Hrs.
I/O Streams: C++ Class Hierarchy- File Stream-Text File Handling- Binary File Handling during file operations.	
Textbook 1: Chapter 12(12.5) , Chapter 13 (13.6,13.7)	
Module-5	8 Hrs.
Exception Handling: Introduction to Exception - Benefits of Exception handling- Try and catch blockThrow statement- Pre-defined exceptions in C++.	
Textbook 2: Chapter 13 (13.2 to13.6)	
Practical Module	
<ol style="list-style-type: none"> 1. Write a C++ program to sort the elements in ascending and descending order. 2. Write a C++ program to find the sum of all the natural numbers from 1 to n. 3. Write a C++ program to swap 2 values by writing a function that uses call by reference 	

technique.

4. Write a C++ program to demonstrate function overloading for the following prototypes.

add(int a, int b)

add(double a, double b)

5. Create a class named Shape with a function that prints "This is a shape". Create another class named Polygon inheriting the Shape class with the same function that prints "Polygon is a shape". Create two other classes named Rectangle and Triangle having the same function which prints "Rectangle is a polygon" and "Triangle is a polygon" respectively. Again, make another class named Square having the same function which prints "Square is a rectangle". Now, try calling the function by the object of each of these classes.

6. Suppose we have three classes Vehicle, FourWheeler, and Car. The class Vehicle is the base class, the class FourWheeler is derived from it and the class Car is derived from the class FourWheeler. Class Vehicle has a method 'vehicle' that prints 'I am a vehicle', class FourWheeler has a method 'fourWheeler' that prints 'I have four wheels', and class Car has a method 'car' that prints 'I am a car'. So, as this is a multi-level inheritance; we can have access to all the other classes' methods from the object of the class Car. We invoke all the methods from a Car object and print the corresponding outputs of the methods. So, if we invoke the methods in this order, car(), fourWheeler(), and vehicle(), then the output will be

I am a car

I have four wheels

I am a vehicle

Write a C++ program to demonstrate multilevel inheritance using this.

7. Write a C++ program to create a text file, check file created or not, if created it will write some text into the file and then read the text from the file.

8. Write a C++ program to write and read time in/from binary file using fstream.

9. Write a function which throws a division by zero exception and catch it in catch block. Write a C++ program to demonstrate usage of try, catch and throw to handle exception.

10. Write a C++ program function which handles array of bounds exception using C++.

Suggested Learning resources

Books (Title of the Book/Name of the author/Name of the publisher/Edition and Year)

Textbooks

1. Bhushan Trivedi, "Programming with ANSI C++", Oxford Press, Second Edition, 2012.
2. Balagurusamy E, Object Oriented Programming with C++, Tata McGraw Hill Education Pvt.Ltd , Fourth Edition 2010.

Weblinks and Video Lectures (e-Resources):

1. Basics of C++ - <https://www.youtube.com/watch?v=BCIS40yzssA>
2. Functions of C++ - <https://www.youtube.com/watch?v=p8ehAjZWjPw>

Tutorial Link:

1. https://www.w3schools.com/cpp/cpp_intro.asp
2. <https://www.edx.org/course/introduction-to-c-3>

Activity Based Learning (Suggested Activities in Class)/ Practical Based learning

- Assign small tasks to Develop and demonstrate using C++

Course Outcomes:

- CO1: Able to understand and design the solution to a problem using object-oriented programming concepts.
- CO2: Able to reuse the code with extensible Class types, User-defined operators and function Overloading.
- CO3: Achieve code reusability and extensibility by means of Inheritance and Polymorphism
- CO4: Implement the features of C++ including templates, exceptions and file handling for providing programmed solutions to complex problems.

CO and PO Mapping

Course Outcomes	Programme Outcomes											
	1	2	3	4	5	6	7	8	9	10	11	12
CO1	3	2			2							
CO2		1			2							2
CO3		1			3							3
CO4		2			3							2