

Teaching Plan (Nov 2021- March 2022)

Name of the Faculty : Mr. Pradeep Kumar

Name of the Course : B.Sc (H) Computer Science

Semester : 1st Sec (if any) : B

Title of the Paper : Programming Fundamentals using C++ (BHCS01)

Course objective:

This course is designed to develop structured as well as object-oriented programming skills using C++ programming language. The course not only focuses on basic C++ constructs but also covers object-oriented programming features in-depth, namely Encapsulation, Abstraction, Inheritance and Polymorphism for writing efficient codes.

Course Learning Outcomes:

On successful completion of the course, students will be able to:

1. Explain significance of object-oriented paradigm
2. Solve programming problems using object-oriented features.
3. Handle external files as well as exceptions.
4. Reuse classes to create new classes.
5. Handle exceptions in programs.

Month	Topics Covered	References
Nov	<p>Introduction to C++, Decision and Control structure: (16 Lectures) Overview of Procedural Programming and Object-Oriented Programming, Using main () function, Header Files, Compiling and Executing Simple Programs in C++. Data types, Variables, Operators, Expressions, Arrays, Keywords, Naming Convention, Type Casting, Input-output statements. Decision making constructs (if, switch), Looping (for, while, do...while). Functions, Command Line Arguments/Parameters, strings.</p> <p>Practicals: Q1 Basic Programs. Q2 Write a program to compute the sum of the first n terms of the following series: $S = 1 - 1 / (2 \wedge 2) + 1 / (3 \wedge 3) - \dots 1 / (n \wedge n)$ where \wedge is exponentiation. The number of terms n is to be taken from user through command line. If command line 12 argument is not found then prompt the user to enter the value of n. Q3. Write a program to remove the duplicates from an array.</p>	[1,2,3]
Dec	<p>OOPs Features: (16 Lectures) Overview of Abstraction, Encapsulation, Inheritance, and Polymorphism. Creating Classes and objects, Modifiers and Access Control, Constructors, Inheritance (Single and multilevel), Polymorphism (Function Overloading, Operator Overloading, Function Overriding)</p> <p>Practicals: Q4. Write a program that prints a table indicating the number of occurrences of each alphabet in the text entered as command line arguments. Q5. Write a menu driven program to perform following operations on strings (without using inbuilt string functions): a) Show address of each character in string b) Concatenate two strings. c) Compare two strings d) Calculate length of the string (use pointers)</p>	[2,4]

	<p>e) Convert all lowercase characters to uppercase f) Reverse the string</p> <p>Q6. Write a program to merge two ordered arrays to get a single ordered array.</p> <p>Q7. Write a program to search a given element in a set of N numbers using Binary search (i) with recursion (ii) without recursion.</p> <p>Q8. Write a program to calculate GCD of two numbers (i) with recursion (ii) without recursion.</p>	
Jan	<p>Memory allocations and Pointers :(14 Lectures) Static and dynamic memory allocation, Pointer variables, Reference Variables, Pointers vs. References, Runtime polymorphism using pointers and references</p> <p>Practicals: Q9. Create Matrix class. Write a menu-driven program to perform following Matrix operations: a) Sum b) Product c) Transpose Q10. Define a class Person having name as a data member. Inherit two classes Student and Employee from Person. Student has additional attributes as course, marks and year and Employee has department and salary. Write display() method in all the three classes to display the corresponding attributes. Provide the necessary methods to show runtime polymorphism. Q11. Create a class Triangle. Include overloaded functions for calculating area. Overload assignment operator and equality operator.</p>	[2,4]
Feb	<p>Exception and File Handling:(10 Lectures) Using try, catch, throw, throws and finally; Nested try, creating user defined exceptions, File I/O Basics, File Operations.</p> <p>Practicals: Q12. Write a program to read two numbers p and q. If q is 0 then throw an exception else display the result of p/q. Q13. Rewrite Matrix class of Q8 with exception handling. Exceptions should be thrown by the functions if matrices passed to them are incompatible and handled by main() function. Q14. Create a class Student containing fields for Roll No., Name, Class, Year and Total Marks. Write a program to store 5 objects of Student class in a file. Retrieve these records from file and display them. Q15. Copy the contents of one text file to another file, after removing all whitespaces.</p> <p>Assignment :2nd week of February. Quiz :1st week of March Discussion of previous year question Papers: 2nd week of March</p>	[5,6]

References

1. Forouzan & Gilbert (2012). Computer Science: A Structured Approach Using C++. Cengage Learning.
2. Schildt, H. (2003). C++: The Complete Reference. 4th edition. Tata McGraw-Hill.
3. <https://fdocuments.in/document/c-string-class-outline-string-initialization-basic-operations-.html>
4. https://www.slideshare.net/AdilAslam4/object-oriented-programming-using-c-slides-13-69148519?next_slideshow=69148519
5. <https://www.slideshare.net/AdilAslam4/file-handling-in-c-69352960>.
6. https://www.slideshare.net/AdilAslam4/exception-handling-in-c-69353237?qid=88582d0d-3809-4a98-89f5-aaaf2da7dbe6&v=&b=&from_search=10

Additional Resources

1. Balaguruswamy, E. (2017). Object Oriented Programming with C++ (7th ed.). McGraw Hill Education.
2. Kanetkar, Y. P. (2015). Let us C++ .2nd edition. BPB Publishers.
3. Prata, S. (2015). C++ Primer Plus 6th edition. Pearson Education India.
4. Stroustrup, B. (2013). The C++ Programming Language .4th Edition. Pearson Education.

Teaching Plan (Jan 2022- April 2022)

Name of the Faculty : Mr. Pradeep Kumar

Name of the Course : B.Sc (Prog.) Mathematical Science

Semester : IV Sec (if any) :

Title of the Paper : Computer System Architecture (BSCS04)

Course objective:

The course will introduce students to the fundamental concepts of digital computer organization, design and architecture. It aims to develop a basic understanding of the design of a computer system.

Course Learning Outcomes:

On successful completion of the course, students will be able to:

1. design combinational circuits using basic building blocks. Simplify these circuits using Boolean Algebra and Karnaugh maps.
2. differentiate between combinational circuits and sequential circuits
3. represent data in binary form, convert numeric data between different number systems and perform arithmetic operations in binary.
4. determine various stages of instruction cycle, various instruction formats and instruction set. describe interrupts and their handling.
5. explain how CPU communicates with memory and I/O devices.

Month	Topics Covered	References
January	<p>Introduction: (12 Lectures) Logic gates, boolean algebra, combinational circuits, circuit simplification, flip-flops and sequential circuits, decoders, multiplexors, registers, counters and memory units.</p> <p>Data Representation and basic Computer Arithmetic: (4 Lectures) Number systems, complements, fixed and floating-point representation, character representation, addition, subtraction, magnitude comparison.</p> <p>Practicals:</p> <ol style="list-style-type: none">1). Write a program to convert a number in Radix 'R' to radix 10 and vice versa. Test the same by<ol style="list-style-type: none">a. Converting an unsigned number from binary, octal, hex to decimal.b. Converting an unsigned number from decimal to binary, octal, hexadecimal.2) Write a program that will prompt for the input of two integer values. Then using the bitwise shift operators show the result of<ol style="list-style-type: none">a. Left shifting the first number by the secondb. Right shifting the first number by the second	[1,2,3]

	<p>c. OR of the first number by the second bitwise d. AND of the first number by the second bitwise</p>	[1,2,3]
February	<p>Basic Computer Organization and Design: (12 Lectures) Computer registers, bus system, instruction set, timing and control, instruction cycle, memory reference, input-output and interrupt. Central Processing Unit: (8 Lectures) Register organization, arithmetic and logical Micro-operations, stack organization, micro programmed control Practicals: 3). Write a program that will prompt for the input of a binary value. Find out following complements. a) One's complement. b) Two's complement 4). Write a program to print the values of a 5 bits binary up-down counter. User should be able to specify the up or down nature of the counter.</p>	[1,2]
March	<p>Programming the Basic Computer: (12 Lectures) Instruction formats, addressing modes, instruction codes, input output programming Practicals: 5) Write a program to implement the following binary operations: a) Addition b) Subtraction using 2's complement 6) Write a program to implement JK and SR flip flop tables</p>	[1,2]
April	<p>Input-output Organization: (5 Lectures) Peripheral devices, I/O interface, Modes of data transfer, direct memory access Test: First week of April Assignment: Second week of April Remedial Classes: Last week of April</p>	[1,4]

References:

- [1]. M. Mano, Computer System Architecture, Pearson Education 1992.
- [2]. W. Stallings, Computer Organization and Architecture Designing for Performance, 8th Edition 2009, Prentice Hall of India
- [3]. Digital Design, M.M. Mano, Pearson Education Asia
- [4]. <https://www.geeksforgeeks.org/modes-of-dma-transfer/?ref=gcse>