|  |  |
| --- | --- |
| **SESSION** | **APRIL 2025** |
| **PROGRAM** | **BACHELOR OF COMPUTER APPLICATIONS (BCA)** |
| **SEMESTER** | **II** |
| **COURSE CODE & NAME** | **DCA1210 OBJECT-ORIENTED PROGRAMMING USING C++** |
|  |  |
|  |  |

**Set-I**

**Q1. Describe the main differences between procedural programming in C and object-oriented programming in C++. 10**

**Ans 1.**

**Main Differences Between Procedural Programming in C and Object-Oriented Programming in C++**

**Programming Paradigms**

Programming languages adopt different paradigms to structure and organize code. C is a procedural programming language that focuses on functions and procedures, while C++ is an object-oriented language designed to handle complexity using the concept of objects and classes. Understanding their differences helps in choosing the right approach for various programming scenarios.

**Conceptual Differences**

The most fundamental difference between C and C++ lies in their design philosophy. C follows a top-down approach where the primary focus is on functions and the sequence of actions, often

Its Half solved only

Buy Complete assignment from us

**Price – 190/ assignment**

**MUJ Manipal University Complete SolvedAssignments MARCH 2025**

buy cheap assignment help online from us easily

we are here to help you with the best and cheap help

**Contact No – 8791514139 (WhatsApp)**

**OR**

**Mail us-** [bestassignment247@gmail.com](mailto:bestassignment247@gmail.com)

**Our website -** [www.assignmentsupport.in](http://www.assignmentsupport.in)

**Q2. Define an inline function and explain its advantages. 10**

**Ans 2.**

**Understanding Inline Functions**

An inline function in C++ is a function where the compiler attempts to expand the function body at the point of each function call instead of performing a traditional call. This is done using the inline keyword. It is particularly useful for small, frequently used functions where the overhead of calling a function can impact performance.

**How Inline Functions Work**

When a function is marked as inline, the compiler replaces the function call with the actual code

**Q3. Explain the concept of exception handling in C++ and its necessity. Discuss the roles of try, throw, and catch in the exception handling mechanism.**

**Ans 3.**

**Concept of Exception Handling in C++ and the Roles of try, throw, and catch**

**Exception Handling in C++**

Exception handling is a crucial feature in C++ that allows a program to deal with unexpected errors or unusual situations that occur during runtime. Instead of crashing the program or producing incorrect results, exception handling enables the program to detect, handle, and recover from these errors gracefully. This is especially important for creating robust, secure, and

**Set-II**

**Q4. Describe basic programming using streams in C++. Include the process of creating, connecting, and disconnecting streams, and provide a simple example program. 10**

**Ans 4.**

**Basic Programming Using Streams in C++ with Process and Example**

**Streams in C++**

In C++, streams are used to perform input and output operations. A **stream** is a flow of data from a source to a destination. The input stream reads data into a program, while the output stream sends data out of the program. C++ provides the **iostream** library that includes classes like istream for input and ostream for output, enabling developers to manage input and output operations efficiently.

**Process of Creating and Connecting Streams**

To perform file operations, C++ uses fstream, which includes ifstream (input file stream),

**Q5. What are access specifiers in C++? Provide examples to demonstrate the use of each access specifier in a class. 10**

**Ans 5.**

**Access Specifiers in C++ with Examples**

**Access Specifiers**

Access specifiers in C++ are keywords used to define the visibility and accessibility of class members (variables and functions). They control how the members of a class can be accessed in a program. The three main access specifiers in C++ are public, private, and protected. Using access specifiers allows developers to implement the principle of data encapsulation, which is a key concept in object-oriented programming.

**Public Access Specifier**

The public access specifier allows class members to be accessible from anywhere in the

**Q6. Explain the concept of operator overloading in C++.**

**Ans 6.**

**Operator Overloading**

Operator overloading is a feature in C++ that allows existing operators to be redefined so that they can work with user-defined data types, such as objects of a class. This improves code readability and makes operations involving class objects more intuitive. It is a type of polymorphism where the same operator behaves differently depending on its operands.

**Purpose and Need for Operator Overloading**

In C++, built-in data types like int or float can be manipulated using arithmetic, relational, and logical operators. However, these operators don’t work with user-defined types by default. Operator overloading solves this limitation by allowing developers to redefine how operators