

## **Assignment 1:**

### **Unit 2:**

1. How does encapsulation contribute to the concept of data hiding in C++? Provide an example to illustrate its benefits.
2. Explain the process of message passing in C++. How does it facilitate communication between objects?
3. Discuss the significance of constructors in C++. How do default constructors, parameterized constructors, and copy constructors differ in their usage?
4. How can constructor overloading be implemented in C++? Provide an example showcasing different constructors with varying parameters.
5. What is the role of destructors in C++? How are they invoked and what actions are typically performed in a destructor?
6. Explain the concept of dynamic memory allocation using the new and delete operators in C++. When and why is it necessary to allocate memory dynamically?
7. How can the dynamic constructor be utilized in C++? Provide an example demonstrating its usage and advantages.
8. Discuss the implementation of data abstraction in C++. How does it contribute to the encapsulation and hiding of implementation details?
9. Explain the concept of inline functions in C++. How are they different from regular functions? What are the advantages and potential drawbacks of using inline functions?
10. How can pointer and reference variables be used in C++ functions? Provide examples to illustrate their usage in passing and returning values.

### **Unit 3:**

1. What is inheritance in C++ and how does it promote software reusability? Provide an example to demonstrate the concept.
2. How can the relationship between base classes and derived classes be represented using an inheritance relationship diagram? Explain with an illustration.
3. Discuss the different modes of inheritance in C++: public, private, and protected. How does each mode affect the accessibility of base class members in the derived class?
4. Explain the various types of inheritance in C++: single, multilevel, hierarchical, multiple, and hybrid. Provide examples to illustrate each type.
5. How is ambiguity resolved in multiple inheritance scenarios in C++? Discuss the methods used to handle conflicts when multiple base classes have members with the same name.
6. What is multipath inheritance and how does it relate to the concept of a virtual base class? Explain the purpose and usage of a virtual base class in C++.
7. How are constructors and destructors implemented in a derived class? Discuss any special considerations or modifications required compared to the base class.
8. Define the terms subclass, subtype, and the principle of substitutability in the context of inheritance in C++. Explain the relationship between these concepts.

9. Describe the concept of composition in C++ and its role in achieving software reusability. Provide an example demonstrating the implementation of composition in a program.
10. How can the relationship between classes be represented using a composition relationship diagram? Explain the notation and connections used to depict composition relationships.
11. Discuss the importance of software reusability in programming. How does inheritance in C++ contribute to achieving this goal?