## Section A: Multiple Choice Questions

1. **What is the default access specifier for class members in C++?**  
   **B. private**
2. **Which of the following is true about cin in C++?**  
   **B. It is an object of class istream**
3. **What does the explicit keyword prevent in constructors?**  
   **B. Implicit type conversion**
4. **Which operator is used for formatted output in C++?**  
   **B. <<**
5. **What is the purpose of a friend function?**  
   **B. To access private members of a class**
6. **Which of the following is a correct way to prevent multiple inclusions of a header file?**  
   **D. All of the above**
7. **What is the result of passing a variable by reference?**  
   **C. The address of the variable is passed**
8. **Which of the following is true about virtual destructors?**  
   **D. They ensure proper cleanup of derived class objects**
9. **What is the purpose of the override keyword in C++11?**  
   **B. To indicate that a function is overriding a base class function**
10. **Which of the following best describes the role of a constructor initializer list?**  
    **A. To initialize member variables before the constructor body executes**

**Section B: Short Answer Questions**

1. Value-based vs Reference-based Object Models:  
   Value-based means objects are copied when assigned or passed.  
   Reference-based means objects are accessed using references or pointers, avoiding copies.
2. Operator Overloading (+):  
   Operator overloading allows custom behavior for operators with user-defined types by defining special functions like operator+.
3. Role of const:  
   const prevents modification. It is used to declare constant variables, to make references or pointers read-only, and to ensure member functions do not modify the object.
4. Inheritance and Access Specifiers:  
   Inheritance allows a class to reuse another class's features.  
   Access specifiers control visibility: public keeps original access, protected makes base members protected, and private makes them private
5. Virtual Function and Polymorphism:  
   A virtual function allows a derived class to override a base class method.  
   It supports runtime polymorphism by calling the correct method based on the actual object type.