X

Week 3: Assignment 3

The due date for submitting this assignment has passed.

As per our records you have not submitted this assignment.

Due on 2025-02-12, 23:59 IST.

1) Consider the following code: ***1 point***

```
class A {
public:
    void display() { std::cout << "Base class A\n"; }
};

class B : public A {
public:
    void show() { std::cout << "Derived class B\n"; }
};

int main() {
    B obj;
    obj.display();
    obj.show();
    return 0;
}
```

What is the output of the above program?

- Base class A
  Derived class B
- Derived class B
  Base class A
- Base class A
- Derived class B

No, the answer is incorrect.
Score: 0

Accepted Answers:
Base class A
Derived class B

2) In C++, ambiguity occurs in multiple inheritance when: ***1 point***

- A derived class has a method with the same name as a method in the base class.
- Two base classes have methods with the same name, and a derived class inherits from both.
- A derived class has no constructor defined.
- A base class has a private member function.

No, the answer is incorrect.
Score: 0

Accepted Answers:
Two base classes have methods with the same name, and a derived class inherits from both.

3) In a program to demonstrate multilevel inheritance with the following requirements: ***1 point***

- Class A contains a method display() that prints ”Class A”.
- Class B inherits from A and adds a method show() that prints ”Class B”.
- Class C inherits from B and adds a method output() that prints ”Class C”.

Which of the following correctly calls all three methods from an object of C?

- C obj; obj.display(); obj.show(); obj.output();
- B obj; obj.show(); obj.output();
- A obj; obj.display(); obj.show(); obj.output();
- C obj; obj.output();

No, the answer is incorrect.
Score: 0

Accepted Answers:
C obj; obj.display(); obj.show(); obj.output();

4) Consider the following script: ***1 point***

```
class A {
    void display() { System.out.println("Class A"); }
}

class B extends A {
    void show() { System.out.println("Class B"); }
}

class C extends A {
    void output() { System.out.println("Class C"); }
}

public class Main {
    public static void main(String[] args) {
        B objB = new B();
        objB.display();
        objB.show();

        C objC = new C();
```

```
        objC.display();
        objC.output();
    }
}
```

What is the output of the above program?

- Class A
  Class B
  Class A
  Class C
- Class B
  Class A
  Class C
- Class A
  Class A
- Class B
  Class C

No, the answer is incorrect.
Score: 0

Accepted Answers:
Class A
Class B
Class A
Class C

5) Which of the following is true about method overriding in Java? ***1 point***

- The overridden method must have a different return type.
- The overridden method must have the same name and parameters as the base class method.
- The base class method must be private.
- Overriding is not possible in Java.

No, the answer is incorrect.
Score: 0

Accepted Answers:
The overridden method must have the same name and parameters as the base class method.

6) Consider the following C++ code: ***1 point***

```
class Base {
public:
    virtual void display() { std::cout << "Base class\n"; }
};

class Derived : public Base {
public:
    void display() override { std::cout << "Derived class\n"; }
};

int main() {
    Base* ptr;
    Derived obj;
    ptr = &obj;
    ptr->display();
    return 0;
}
```

What is the output of the program?

- Base class
- Derived class
- Compilation error
- Undefined behavior

No, the answer is incorrect.
Score: 0

Accepted Answers:
Derived class

7) Which of the following statements about virtual functions is false? ***1 point***

- Virtual functions allow dynamic (runtime) polymorphism.
- A virtual function must be declared as virtual in the base class.
- A virtual function can have default arguments.
- A virtual function cannot be overridden in a derived class.

No, the answer is incorrect.
Score: 0

Accepted Answers:
A virtual function cannot be overridden in a derived class.

8) Write a C++ program to demonstrate virtual functions with the following requirements: ***1 point***

- Base class Shape has a virtual function area().
- Derived class Rectangle overrides area() to compute the area of a rectangle.
- Derived class Circle overrides area() to compute the area of a circle.

Which of the following correctly uses a base class pointer to call area() for both shapes?

- Shape* ptr; Rectangle rect; ptr = ▭ ptr->area();
- Shape* ptr = new Circle(); ptr->area();
- Both A and B
- None of the above

No, the answer is incorrect.
Score: 0

Accepted Answers:
Both A and B

9) When a class is derived from a base class using protected inheritance, how are the public and protected members of the base class treated in the derived class? ***1 point***

- They both become private members in the derived class.
- They both remain public members in the derived class.
- They both become protected members in the derived class.
- They are inaccessible in the derived class.

No, the answer is incorrect.
Score: 0

Accepted Answers:
They both become protected members in the derived class.

10) Which of the following is a potential problem associated with multiple inheritance in C++? ***1 point***

- Ambiguity in accessing members when two base classes have members with the same name.
- Lack of runtime polymorphism.
- Inability to overload operators in derived classes.
- Restriction on the number of base classes a derived class can inherit from.

No, the answer is incorrect.
Score: 0

Accepted Answers:
Ambiguity in accessing members when two base classes have members with the same name.