X

Week 2: Assignment 2

The due date for submitting this assignment has passed.

Due on 2025-02-05, 23:59 IST.

Assignment submitted on 2025-02-01, 16:36 IST

1) Which of the following best describes a class in object-oriented programming? ***1 point***

- A block of code that defines functions only.
- A blueprint for creating objects, encapsulating data and methods.
- A template for functions without data.
- A data structure for storing primitive data types.

Yes, the answer is correct.
Score: 1

Accepted Answers:
A blueprint for creating objects, encapsulating data and methods.

2) What happens when an object is created from a class in C++? ***1 point***

- Memory is allocated for the object's member variables only.
- Memory is allocated for member functions and variables.
- Member functions are inherited and memory is allocated for them.
- Member functions are shared across objects, and memory is allocated only for variables.

Yes, the answer is correct.
Score: 1

Accepted Answers:
Member functions are shared across objects, and memory is allocated only for variables.

3) Which of the following statements about constructors is false? ***1 point***

- Constructors must have the same name as the class.
- Constructors can be overloaded in C++.
- Constructors cannot be private in C++.
- Constructors do not have a return type.

Yes, the answer is correct.
Score: 1

Accepted Answers:
Constructors cannot be private in C++.

4) In a Java program to define a class Circle with: ***1 point***

- A constructor that initializes its radius.
- A method getArea() to return the area of the circle.

Identify the correct syntax to create an object of this class.

- Circle c1 = new Circle();
- Circle c1 = new Circle(radius);
- Circle c1 = Circle(radius);
- Circle c1 = new Circle[radius];

Yes, the answer is correct.
Score: 1

Accepted Answers:
Circle c1 = new Circle(radius);

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

```
#include <iostream>
class Test {
public:
    Test() { std::cout << "Constructor called\n"; }
    ~Test() { std::cout << "Destructor called\n"; }
    void display() { std::cout << "Display function\n"; }
};

int main() {
    Test t1;
    t1.display();
    return 0;
}
```

What is the output of this program?

- Constructor called
  Display function
  Destructor called
- Constructor called
  Destructor called
  Display function

Display function
Constructor called
Destructor called

Constructor called
Display function

Yes, the answer is correct.
Score: 1

Accepted Answers:
Constructor called
Display function
Destructor called

6) What is the primary purpose of a destructor in C++? ***1 point***

- To initialize an object when it is created.
- To release memory and resources when an object is destroyed.
- To overload operators for memory allocation.
- To define default behavior for inheritance.

Yes, the answer is correct.
Score: 1

Accepted Answers:
To release memory and resources when an object is destroyed.

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

```
#include <iostream>
class Sample {
public:
    Sample() { std::cout << "Constructor called\n"; }
    ~Sample() { std::cout << "Destructor called\n"; }
};

void createObject() {
    Sample obj;
    std::cout << "Inside createObject function\n";
}

int main() {
    std::cout << "Before calling createObject\n";
    createObject();
    std::cout << "After calling createObject\n";
    return 0;
}
```

What is the output of this program?

- Constructor called
  Inside createObject function
  Destructor called
  Before calling createObject
  After calling createObject
- Before calling createObject
  Constructor called
  Inside createObject function
  Destructor called
  After calling createObject
- Before calling createObject
  Inside createObject function

Constructor called
Destructor called
After calling createObject

Before calling createObject
Constructor called
Inside createObject function
After calling createObject
Destructor called

No, the answer is incorrect.
Score: 0
Accepted Answers:
Before calling createObject
Constructor called
Inside createObject function
Destructor called
After calling createObject

8) A class in C++ has multiple constructors. How does the compiler decide which constructor to use? ***1 point***

- Based on the return type.
- Based on the arguments passed during object creation.
- The first constructor is always used.
- The last constructor is always used.

Yes, the answer is correct.
Score: 1
Accepted Answers:
Based on the arguments passed during object creation.

9) In a C++ program that: ***1 point***

- Defines a class FileHandler with a constructor that opens a file and a destructor that closes the file.
- Demonstrates file handling using objects of this class.

Choose the correct constructor signature:

- FileHandler(std::string filename);
- FileHandler(char* filename);
- FileHandler(const char filename[]);
- All of the above

Yes, the answer is correct.
Score: 1

Accepted Answers:
All of the above

10) Analyze the following C++ code and identify the correct output: ***1 point***

```
#include <iostream>
class Rectangle {
    int length, width;
public:
    Rectangle(int l, int w) : length(l), width(w) {}
    int area() { return length * width; }
};

int main() {
    Rectangle rect(5, 3);
    std::cout << rect.area();
    return 0;
}
```

- 15
- 8
- Compilation error
- Undefined behavior

Yes, the answer is correct.
Score: 1

Accepted Answers:
15