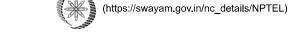
Χ



rg0102@srmist.edu.in ~



NPTEL (https://swayam.gov.in/explorer?ncCode=NPTEL) » Fundamentals of Object Oriented Programming (course)

Announcements (announcements) About the Course (preview) Q&A (forum) Progress (student/home) Mentor (student/mentor)

Review Assignment (assignment review) Course Recommendations New! (/course recommendations)

Click to register for **Certification exam**

(https://examform.nptel.ac.]n/2025_01/Weeksh2ard/Assignment 2

The due date for submitting this assignment has passed.

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

If already registered, click to check your payment status

Course outline

About NPTEL ()

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

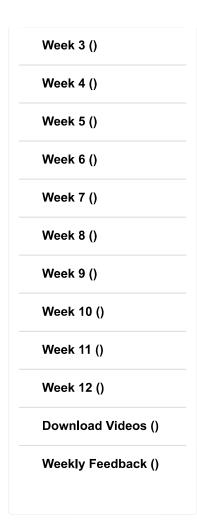
1) Which of the following best describes a class in object-oriented programming?

- 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.

How does an NPTEL online course work? () Week 0 () Week 1 () Week 2 () Classes and Objects in C++ (unit? unit=32&lesson=33) Classes and Objects in Java and Solved problems (unit? unit=32&lesson=34) Constructors in C++: Default and Parameterized (unit? unit=32&lesson=35) Constructors in C++: Copy Constructor (unit? unit=32&lesson=36) Constructors in Java: Default and Parameterized (unit? unit=32&lesson=37) Quiz: Week 2: **Assignment 2** (assessment?name=38)

Solution for Week 2 (unit? unit=32&lesson=93)

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++. 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"; }</pre>
      "Test() { std::cout << "Destructor called\n"; }
     void display() { std::cout << "Display function\n"; }</pre>
};
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++? 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:
#include <iostream>
class Sample {
public:
     Sample() { std::cout << "Constructor called\n"; }</pre>
     ~Sample() { std::cout << "Destructor called\n"; }
};
void createObject() {
     Sample obj;
     std::cout << "Inside createObject function\n";</pre>
int main() {
     std::cout << "Before calling createObject\n";</pre>
     createObject();
     std::cout << "After calling createObject\n";</pre>
     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	
	Defense colling and de Object	
	Before calling createObject	
	Constructor called	
	Inside createObject function	
	After calling createObject	
	Destructor called	
	No, the answer is incorrect. Score: 0	
A	Accepted Answers:	
Ε	Before calling createObject	
C	Constructor called	
I.	Inside createObject function	
L	Destructor called	
P	After calling createObject	
٥,		
) A class in C++ has multiple constructors. How does the compiler decide which constructor	1 point
) A class in C++ has multiple constructors. How does the compiler decide which constructor use?	1 point
		1 point
	use?	1 point
	Based on the return type.	1 point
	Based on the return type. Based on the arguments passed during object creation.	1 point
to u	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	1 point
to u	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:	1 point
to u	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	1 point
to u	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.	1 point
to u	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.	
to u	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.	
to u	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. In a C++ program that: Defines a class FileHandler with a constructor that opens a file and a destructor that closes the file.	

```
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:
#include <iostream>
class Rectangle {
    int length, width;
public:
    Rectangle(int 1, int w) : length(1), 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
```