**Time allowed: 90 Minutes Max. Marks: 40**

**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1. How is the "this" pointer represented in C++?
   1. this.ptr
   2. self
   3. thisptr
   4. **this**
2. What is the purpose of the "new" operator in C++?
   1. **Allocate memory for a new object**
   2. Create a new class definition
   3. Allocate memory for a variable
   4. Create a new object with initial values
3. What is the keyword used to destroy an object explicitly in C++?
   1. break
   2. **delete**
   3. free
   4. destroy
4. Which operator is used to access class members through a pointer to an object?
   1. \*
   2. **->**
   3. ::
   4. .
5. What is the function of a static member variable in a class?
   1. It is a constant value for all objects of the class
   2. **It is shared among all objects of the class**
   3. It is accessible only within the class
   4. It is used to control access to private members
6. Which of the following is true about the default constructor?
   1. **It is automatically provided by the compiler if no constructors are defined**
   2. It is automatically called when an object is destroyed
   3. It is automatically called when an object is created
   4. It is automatically provided by the compiler if the class has a destructor
7. When does the copy constructor get called in C++?
   1. When an object is passed to a function by value
   2. When an object is returned from a function by value
   3. When an object is explicitly destroyed using the delete keyword
   4. **When an object is created using another object of the same class**
8. What is the purpose of using the "explicit" keyword with a constructor?
   1. **It prevents the constructor from being called implicitly**
   2. It allows the constructor to be called implicitly
   3. It makes the constructor a friend of the class
   4. It converts the constructor into a virtual constructor
9. Can a destructor be overloaded in C++?
   1. Yes
   2. **No**
10. Which constructor is called first in a derived class?
    1. Derived class's constructor
    2. **Base class's constructor**
    3. Destructor
    4. Copy constructor

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

1. What is the output of the following code?

#include <iostream>

using namespace std;

class Base {

public:

Base() { cout << "Base constructor" << endl; }

virtual ~Base() { cout << "Base destructor" << endl; }

};

class Derived : public Base {

public:

Derived() { cout << "Derived constructor" << endl; }

~Derived() { cout << "Derived destructor" << endl; }

};

int main() {

Base\* ptr = new Derived();

delete ptr;

return 0;

}

1. **Base constructor, Derived constructor, Derived destructor, Base destructor**
2. Derived constructor, Base constructor, Base destructor, Derived destructor
3. Base constructor, Derived constructor, Base destructor
4. Derived constructor, Base destructor
5. What will be the output of the following C++ code?

#include <iostream>

using namespace std;

class CDummy

{

public:

int isitme (CDummy& param);

};

int CDummy::isitme (CDummy& param)

{

if (&param == this)

return true;

else

return false;

}

int main ()

{

CDummy a;

CDummy \*b = &a;

if (b->isitme(a))

{

cout << "execute";

}

else

{

cout<<"not execute";

}

return 0;

}

1. **execute \*\*\*\*\*\*\***
2. not execute
3. error
4. both execute & not execute
5. What is the correct syntax of accessing a static member of a Class?

Example class:

class A

{

public:

static int value;

}

* 1. A.value
  2. **A::value**
  3. A->value
  4. A^value

1. What will be the output of the following C++ code?

#include <iostream>

using namespace std;

class S

{

int m;

public:

#define MAC(S::m)

};

int main(int argc, char const \*argv[])

{

cout<<"Hello World";

return 0;

}

* 1. Hello World
  2. **Error**
  3. Segmentation Fault
  4. Blank Space

1. What will be the output of the following C++ code?

#include <iostream>

#include <string>

using namespace std;

class A

{

static int a;

public:

void show()

{

a++;

cout<<"a: "<<a<<endl;

}

};

int A::a = 5;

int main(int argc, char const \*argv[])

{

A a;

return 0;

}

* 1. Error as a private member a is referenced outside the class
  2. Segmentation fault
  3. **No output**
  4. Program compiles successfully but gives run-time error

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16) You have been assigned the task of implementing a C++ class that represents a special kind of number.

Your task is to design and implement the class, along with overloading the unary '-' operator to handle the negation of this unique number.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 10 | 75 | 80 |
| **Output** | -10 | -75 | -80 |

Solution :

**#include <iostream>**

**using namespace std;**

**class Number {**

**private:**

**int value;**

**public:**

**Number(int v) : value(v) {}**

**// Overloading the unary '-' operator**

**Number operator-() {**

**return Number(-value);**

**}**

**void display() {**

**cout << value << endl;**

**}**

**};**

**int main() {**

**Number num(10);**

**Number negNum = -num;**

**negNum.display();**

**return 0;**

**}**

Q17) In a virtual reality game, the players are exploring a fantasy world filled with magical objects.

As part of their adventure, they come across enchanted circles and cylinders with unique properties.

To help the players understand these objects, you need to create a program that calculate area of circle and volume of the cylinder.

Cylinder should inherit properties of Circle.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | radius: 5  height: 7 | radius: 4  height: 7 | radius: 3  height: 6 |
| **Output** | Cylinder's Volume: 549.5 | Cylinder's Volume: 351.68 | Cylinder's Volume: 169.56 |

Solution :

**#include <iostream> // Include the standard input/output stream library**

**#include <cmath> // Include the math library for mathematical functions**

**#define M\_PI 3.14 // Define the constant value of pi**

**// Base class: Circle**

**class Circle {**

**protected:**

**double radius; // Protected data member to store the radius of the circle**

**public:**

**// Constructor to initialize the radius of the circle**

**Circle(double r) : radius(r) {}**

**// Function to calculate the area of the circle**

**double calculateArea() {**

**return M\_PI \* radius \* radius; // Area formula for a circle**

**}**

**};**

**// Derived class: Cylinder (inherits from Circle)**

**class Cylinder : public Circle {**

**private:**

**double height; // Private data member to store the height of the cylinder**

**public:**

**// Constructor to initialize the radius and height of the cylinder using the Circle's constructor**

**Cylinder(double r, double h) : Circle(r), height(h) {}**

**// Function to calculate the volume of the cylinder**

**double calculateVolume() {**

**return calculateArea() \* height; // Volume formula for a cylinder: area of base \* height**

**}**

**};**

**int main() {**

**double radius, height;**

**std::cout << "Enter radius: ";**

**std::cin >> radius;**

**std::cout << "Enter height: ";**

**std::cin >> height;**

**Cylinder cylinder(radius, height); // Create a Cylinder object with the given radius and height**

**std::cout << "Cylinder's Volume: " << cylinder.calculateVolume() << std::endl;**

**return 0;**

**}**

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18) Imagine you are working on a weather monitoring system for a meteorological station, and you need to create a C++ program to handle temperature conversions. You want to design a class called Temperature to represent temperature in Celsius and provide a method to convert Celsius to Fahrenheit.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 25.5 | 39.4 | 35.48 |
| **Output** | Temperature in Celsius: 25.5  Temperature in Fahrenheit: 77.9 | Temperature in Celsius: 39.4  Temperature in Fahrenheit: 102.92 | Temperature in Celsius: 35.48  Temperature in Fahrenheit: 95.864 |

Solution :

**#include <iostream>**

**class Temperature {**

**private:**

**double celsius;**

**public:**

**// Explicit Constructor to convert Celsius to Fahrenheit**

**explicit Temperature(double tempCelsius) {**

**celsius = tempCelsius;**

**}**

**double getCelsius() const {**

**return celsius;**

**}**

**double getFahrenheit() const {**

**return (celsius \* 9 / 5) + 32;**

**}**

**};**

**int main() {**

**Temperature tempCelsius(35.48);**

**std::cout << "Temperature in Celsius: " << tempCelsius.getCelsius() << std::endl;**

**std::cout << "Temperature in Fahrenheit: " << tempCelsius.getFahrenheit() << std::endl;**

**return 0;**

**}**