Lab9_classes

OBJECT ORIENTED PROGRAMMING

BSCS-SPRING-2022

Class

A class is a blueprint for the object. It is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

Class Declaration

In C+, a class is defined using the class keyword. This should be followed by the class name. The class body is then added between curly braces { }.

Syntax:

```
class class-name
{
        // data
        // functions
};
```

- The class-name is the name to assign to the class.
- The data is the data for the class, normally declared as variables.
- The functions are the class functions.

Private and Public Keywords

You must have come across these two keywords. They are access modifiers.

- **Private:** When the private keyword is used to define a function or class, it becomes private. Such are only accessible from within the class.

- **Public:** The public keyword, on the other hand, makes data/functions public. These are accessible from outside the class.

Object Definition

Objects are created from classes. Class objects are declared in a similar way as variables are declared. The class name must start, followed by the object name. The object of the class type.

Syntax:

class-name object-name;

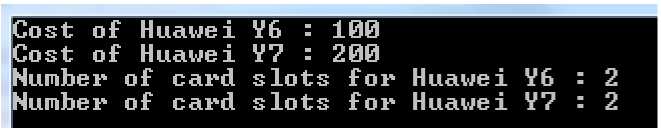
- The class-name is the name of the class from which an object is to be created.
- The object-name is the name to be assigned to the new object.

This process of creating an object from a class is known as instantiation.

Example:

```
#include <iostream>
using namespace std;
class Phone {
public:
        double cost;
        int slots;
};
int main() {
        Phone Y6;
        Phone Y7;

        Y6.cost = 100.0;
        Y6.slots = 2;

        Y7.cost = 200.0;
        Y7.slots = 2;
        cout << "Cost of Huawei Y6 : " << Y6.cost << endl;
        cout << "Cost of Huawei Y7 : " << Y7.cost << endl;

        cout << "Number of card slots for Huawei Y6 : " << Y6.slots << endl;
        cout << "Number of card slots for Huawei Y7 : " << Y7.slots << endl;

        return 0;
}
```

Output:



```
Cost of Huawei Y6 : 100
Cost of Huawei Y7 : 200
Number of card slots for Huawei Y6 : 2
Number of card slots for Huawei Y7 : 2
```

Constructors and Destructors

What is Constructors?

Constructs are special functions that initialize objects. The C++ compilers calls a constructor when creating an object. The constructors help to assign values to class members. Of course, this is after they have been allocated some memory space.

What is Destructors?

Destructors on the other hand help to destroy class objects.

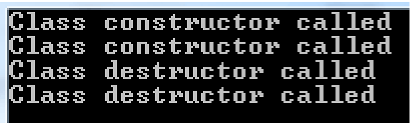
The constructor name must be similar to the class name. Constructors do not have a return type.

The constructor can be defined inside or outside the class body. If defined outside the class body, it should be defined with the class name and the scope resolution operator (::).

Example:

```
#include <iostream>
using namespace std;
class ClassA {
public:
        ClassA() {
                cout << "Class constructor called"<<endl;
        }
        ~ClassA() {
                cout << "Class destructor called"<<endl;
        }
};
int main() {
        ClassA a;
        int p = 1;
                if (p) {
                        ClassA b;
                }
}
```

Output:



```
Class constructor called
Class constructor called
Class destructor called
Class destructor called
```

Tasks

Problem 1:

Write a class **Point** with data members

x : a integer x coordinate

y : a integer x coordinate

The class has the following member functions.

1. Default constructor initializing the coordinates to zero
   Point()
2. A constructor that takes the values and initializes coordinates with x1 and y1
   Point (int x1, int y1)
3. A destructor that prints the following statement on screen :"Destructor Called"
4. Getter and Setter functions for x and y.

Problem 2:

Write a class **Cube** (to calculate volume of a cube) having data members

side: a integer value for side of cube

parametrized constructor: to initialize value of side

destructor: print statement in that ‘class destructor called

calculate_volume: a function, return volume of cube

Problem 3:

You have to create a class, named **Student**, representing the student's details. Create setter and getter functions for each element; that is, the class should have default constructor along with the following functions:

getAge, setAge

getFirstName, setFirstName

getLastName, setFastName

getStandard, setStandard

Also, you have to create another method toString() which returns the string consisting of the above elements, separated by a comma(,). You can refer to string stream for this.

Sample Input:

15

john carmack

10

Sample Output:

15,john,carmack,10

Problem 4:

Kristen is a high school teacher. She wants to know how many students (if any) have scored higher than 200 in the exams given during this semester.

Create a class named **Student** with the following specifications:

Number of student n in Kristen class

An instance variable named scores to hold a student's exam scores.

A void input() function that reads integers and saves them to scores.

A calculateTotalScore() function that returns the sum of the student's scores.

Input Format:

In the void Student::input() function, you must read 5 scores from user and save them to your scores instance variable.

Constraints:

1 <= students_strength <= 20

0 <= examScore <= 50

Output Format:

In the int Student::calculateTotalScore() function, you must print the student's total marks (the sum of the values in scores) of students having >=200 marks.

Sample Input:

Total students: 4

Student1: 30 40 45 10 10

Student2: 40 40 40 50 50

Student3: 50 20 30 10 10

Student4: 26 40 35 49 50

Sample Output:

Student2 have marks 220.

Student4 have marks 200.