Object Oriented ProgrammingLab

SPRING - 2023

LAB 09

FAST National University of Computer and Emerging Sciences

Learning Outcomes

In this lab you are expected to learn the following:

- Classes and member functions

Class:

A class 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. In other words, a class is a blueprint for the object.

A class is defined in C++ using the keyword **class** followed by the **name** of the class.

```
class className {
   // some data
   // some functions
};
```

For example,

```
class Room {
    public:
        double length;
        double breadth;
        double height;

        double calculateArea(){
            return length * breadth;
        }

        double calculateVolume(){
            return length * breadth * height;
        }

};
```

Here, we defined a class named **Room**.

The variables **length**, **breadth**, and **height** declared inside the class are known as **data members**. And, the functions **calculateArea()** and **calculateVolume()** are known as **member functions** of a class

Constructors:

A constructor is a special type of member function that is called automatically when an object is created.

In C++, a constructor has the same name as that of the class and it does not have a return type. For example,

```
class  Wall {
  public:
    // create a constructor
    Wall() {
      // code
    }
};
```

Default Constructors:

A constructor with no parameters is known as a default constructor. In the example above, Wall() is a default constructor

```cpp
// C++ program to demonstrate the use of default constructor

#include <iostream>
using namespace std;

// declare a class
class  Wall {
  private:
    double length;

  public:
    // default constructor to initialize variable
    Wall() {
      length = 5.5;
      cout << "Creating a wall." << endl;
      cout << "Length = " << length << endl;
    }
};

int main() {
  Wall wall1;
  return 0;
}
```

Parameterized Constructors:

In C++, a constructor with parameters is known as a parameterized constructor. This is the preferred method to initialize member data.

```cpp
// declare a class
class Wall {
  private:
    double length;
    double height;

  public:
    // parameterized constructor to initialize variables
    Wall(double len, double hgt) {
      length = len;
      height = hgt;
    }

    double calculateArea() {
      return length * height;
    }
};

int main() {
  // create object and initialize data members
  Wall wall1(10.5, 8.6);
  Wall wall2(8.5, 6.3);

  cout << "Area of Wall 1: " << wall1.calculateArea() << endl;
  cout << "Area of Wall 2: " << wall2.calculateArea();

  return 0;
}
```

C++ Access Specifiers:

In C++, there are 3 access modifiers:

1. public
2. private
3. protected (will be covered in the upcoming labs)

1. Public Access Specifiers:

● The public keyword is used to create public members (data and functions). ● The public members are accessible from any part of the program.

2. Private Access Specifiers:

● The private keyword is used to create private members (data and functions). ● The private members can only be accessed from within the class.

Submission Instructions:

1. Now create a new folder with name *ROLLNO_SEC_LAB01* **e.g. i22XXXX_A_LAB09**
2. You need to display your roll no and name before the output of each question.
3. Move all of your .**cpp and .h files** to this newly created directory and compress it into a **.zip file**.
4. Now you have to submit this zipped file on Google Classroom.
5. If you don't follow the above-mentioned submission instruction, you will be marked **zero.**
6. Plagiarism in the Lab Task will result in **zero** marks in the whole category.

Q1. Consider class named Employee

1. Following static private data members
    - no_employee (int)
2. Following public data members
    - name (of type string)
    - empID (of type integer): use no_employee to assign EmpID
    - EmpType (string) (permanent or hourly)
    - Hours_worked
2. Write a default Constructor
3. Write a parameterized Constructor
4. Write a function static int getNumberOfEmployees()
5. Write a function calculate_the_income(). For hourly employee, per hour income is Rs150. For permanent employee he gets paid the salary for exact 240 hours, no matter how many actual hours he or she worked.Again, one hour salary is Rs. 150.

Example:

1. Create a Employee object with name="abc" type="permanent" and hours=10
2. Display details of the employee along with the EmpID
3. Calculate and display his income
4. Create a Employee object with name="def" type="hourly" and hours=10
5. Display details of the employee along with the EmpID
6. Calculate and display his income
7. Display the number of employees

Q 2. Write a class **Distance** with the following private data members:

- int feet
- int inch

Then implement member functions;

1. Write a default constructor
1. Overloaded constructors
    - Distance(int f, int i) // parameterized constructor
    - Distance(Distance &copy) // copy constructor
2. Create setter and getter methods for all data members
3. Write following member function in the class

    -Distance addDistance(Distance &d2)

    It add both distance and returns newly generated distance.

    void display()

    It prints the distance.

Output:

1. **Example 1:**

    Use Copy Constructor to assign values

    - D1(15,8), D2(10,7)
    - Addition : (26,3)

Q3. Write a class named as Calculator that holds data in three dimensions. Class should have following private
member variables.

- operand1:A int that holds the value of an operand1
- operand2:A int that holds the value of an operand2

- Write a default constructor that initializes each member variable of class with 0 Calculator()
- Write a one argument constructor that initializes each member variable of class with value n Calculator(int n)
- Write a two argument constructor that initializes operand1 with n1, operand2 with n2 and operand3 with 0 Calculator(int n1,int n2)
- you are required to generate getter setters for each operand.
  - void setOperand1(int op1)
  - int getOperand1()
  - void setOperand2(int op2)
  - int getOperand2()

- Write a function inside class Calculator, named as Add_two. you are required to add first two opperands and return their sum as an integer value

int Add_two ()

- Write a global function fill2DArray which takes three arguments, an array arr of type Calculator of length int size and an array of int sum . You are required to assign values to each Calculator in arr using constructors.
  - for each object you need to call two argument constructor of class Calculator
  - you have to set values for each object such that first generate a random number and assign it to operand1, now operand2 should be like such that
    Example1: if operand1=1000 then operand2=100
    Example2: if operand1=3500 then operand2 =350
  - call its related Add function which takes two argument and store its result in sum array
    Int* fill2DArray (Calculator arr[], int size,int sum[])
    arr is sent by reference

Output format:

```
ARRAY 1:
operand 1: 41 operand 2: 43operand 3: 0
 SUM=45
operand 1: 467 operand 2: 463operand 3: 0
 SUM=513
operand 1: 334 operand 2: 333operand 3: 0
 SUM=367
operand 1: 500 operand 2: 503operand 3: 0
 SUM=550
```