**Object Oriented Programming**

**Lab 5**

**Submitted To:**

Ma’am Amber Madeeha Zeb

**Submitted By:**

Manaal Waseem

FA18-BCE-074

**In Lab:**

**Task 1:**

**Write two different functions with same name that is Area to calculate the area of circle and triangle.**

**Code:**

1 #include <iostream>

2 #include <cmath>

3 **const float** pi=3.1415;

4

5 **using namespace std**;

6

7 **class** Area

8 {

9 **private**:

10 **float** a;

11

12 **public**:

13 Area()

14 {

15 a=0;

16 }

17

18 **void** area(**float** w, **float** h)

19 {

20 a=0.5\*w\*h;

21 }

22

23 **void** area(**float** r)

24 {

25 a=pi\*pow(r,2);

26 }

27

28 **void** show()

29 {

30 **cout**<<"Area:"<< a << **endl**;

31 }

32 };

33

34 **int** main()

35 {

36 Area a1;

37 Area a2;

38

39 **cout**<< "Area of Circle: " << **endl**;

40 a1.area(2.4);

41 a1.show();

42 **cout**<< "Area of Triangle: " << **endl**;

43 a2.area(2.5,3.6);

44 a2.show();

45

46 **return** 0;

47 }

**Output:**



**Task 2:**

**Write a definition of a Counter class having one private data member count of integer type. This class has following functions**

* **void inc\_count( ); // will increment the value of count by 1**
* **int get\_count ( ); // will return the value of count**

**this class has two constructors**

* **Counter( ); // that initialize count by 0**
* **Counter (int i); // that initialize the count by i**

**Create two objects of Counter class. Write a cout statement in constructor and then check whether that statement appear when two object are created. Then increment object 1 3 times and increment object 2 4 times and display their count values.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** Counter

6 {

7 **private**:

8 **int count**;

9 **public**:

10 Counter()

11 {

12 **count**=0;

13 **cout**<< "I am present!!!" << **endl**;

14 }

15 Counter(**int** i)

16 {

17 **count**=i;

18 **cout**<< "I am absolutely present!!!" << **endl**;

19 }

20 **void** inc\_count()

21 {

22 **count**++;

23 }

24 **void** get\_count()

25 {

26 **cout**<< "Count=" << **count** << **endl**;

27 }

28 };

29

30 **int** main()

31 {

32 Counter c1,c2(1);

33

34

35 c1.inc\_count();

36 c1.inc\_count();

37 c1.inc\_count();

38 c1.get\_count();

39 c2.inc\_count();

40 c2.inc\_count();

41 c2.inc\_count();

42 c2.inc\_count();

43 c2.get\_count();

44

45 **return** 0;

46 }

**Output:**

****

**Task 3:**

**Write a definition of class named Race. It has following private data member**

* **carNo (int)**
* **driverID (int)**
* **carModel (int)**

**The class has one constructor Race (int, int, int) that initializes the values of carNo, driverID and carModel. Race class has following member functions**

* **void InputValues( ) // this will be used to input values of data member of Book object from user**
* **void setValues(int cn, int di, int cm); // it will assign values of cn, di and cm to carNo, driverID and carModel respectively**
* **void display( ); // it displays the value of private**

**Create two object of Rave class. Assign values using InputValues and setValues function and display them using display function.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** Race

6 {

7 **private**:

8 **int** carNo;

9 **int** driverID;

10 **int** carModel;

11 **public**:

12 Race(**int** a=0, **int** b=0, **int** c=0)

13 {

14 carNo=a;

15 driverID=b;

16 carModel=c;

17 }

18

19 **void** InputValues()

20 {

21 **cout**<< "Enter Car Num: " << **endl**;

22 **cin**>> carNo;

23 **cout**<< "Enter Driver ID: " << **endl**;

24 **cin**>> driverID;

25 **cout**<< "Enter Car Model: " << **endl**;

26 **cin**>> carModel;

27 }

28

29 **void** SetValues(**int** n, **int** i, **int** m)

30 {

31 carNo=n;

32 driverID=i;

33 carModel=m;

34 }

35

36 **void** display()

37 {

38 **cout**<< "Car Num: " << carNo << **endl**;

39 **cout**<< "Driver ID: " << driverID << **endl**;

40 **cout**<< "Car Model: " << carModel << **endl**;

41 }

42

43 };

44

45 **int** main()

46 {

47 Race r1,r2(2,4);

48

49 **cout** <<"Object 1: " <<**endl** <<**endl**;

50 r1.SetValues(5,6,7);

51 r1.display();

52 **cout** <<**endl**;

53 **cout** <<"Object 2 before user input: " <<**endl** <<**endl**;

54 r2.display();

55 **cout** <<**endl**;

56 **cout** <<"Object 2: " <<**endl** <<**endl**;

57 r2.InputValues();

58 r2.display();

59

60 **return** 0;

61 }

**Output:**

****

**Task 4:**

**Write a definition of a distance class as shown in the example 4.2 above. Make all the appropriate function constant. Include a constant data member called id of integer type.**

**Create two object constant and non-constant. Assign values and display them. Also check what happens**

* **If you try to modify private data member of an object from the definition of const function.**
* **If you try to modify the private data member of const object from the definition of non-constant function.**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** add //Specifies the class

6 {

7 **private**:

8 **int** iNum1, iNum2, iNum3; //Member data

9

10 add(**int** a=0, **int** b=0)

11 {

12 iNum1=a;

13 iNum2=b;

14 }

15 **public**:

16 **void** input(**int** iVar1, **int** iVar2) //Member function

17 {

18 **cout**<<"Functions to assign values to the member data"<<**endl**;

19 iNum1=iVar1; iNum2=iVar2;

20 }

21 **void** sum(**void**) //Member function

22 {

23 **cout**<<"Functions to find the sum of two numbers"<<**endl**;

24 iNum3=iNum1+iNum2;

25 }

26 **void** disp(**void**) //Member function

27 {

28 **cout**<<"The sum of the two numbers is "<<iNum3<<**endl**;

29 }

30 };

31

32 **int** main()

33 {

34 add A1;

35 **int** iX, iY;

36 **cout**<<"Input two numbers"<<**endl**;

37 **cin**>>iX;

38 **cin**>>iY;

39 A1.input(iX, iY);

40 A1.sum();

41 A1.disp();

42 //system("pause");

43

44

45 **return** 0;

46 }

**Output:**

****

When trying to access through non-constant function

****

When trying to access through constant function

**Conclusion:**

In this task, we concluded that a constant objects’ data members may not be modified from either constant or non-constant member functions.

**Post Lab:**

**Task 1:**

**Write a definition of class named Date that contains three elements the month, the day of the month, and the year, all of type int.**

* **Write two constructors, a default constructor (that initialize each data element of object with zero) and a constructor that takes three parameters (the month, the day of the month, and the year) and initialize the data member of the object with these parameters.**
* **Write a function void printDate() that displays the data elements of the object.**
* **Write a function void setDate(int, int, int) that takes three parameters (he month, the day of the month, and the year) and initialize the data member of the object with these parameters.**

**Write a main function create two object of class Date, the data member of one object is initialized with zero through default constructor. The data member of second object is initialized with some values using a constructor that takes three parameters. Prompt the user to input date (the month, the day of the month, and the year) in a main function, assign these values to the first object (using function setDate) and then display the value of the data members of two objects using function printDate().**

**Code:**

1 #include <iostream>

2

3 **using namespace std**;

4

5 **class** date

6 {

7 **private**:

8 **int** month;

9 **int** day;

10 **int** year;

11

12 **public**:

13 date()

14 {

15 month=0;

16 day=0;

17 year=0;

18 }

19

20 date(**int** a, **int** b, **int** c)

21 {

22 month=a;

23 day=b;

24 year=c;

25 }

26

27 **void** setDate(**int** r, **int** s, **int** t);

28

29 **void** printDate();

30 };

31

32 **void** date::setDate(**int** r, **int** s, **int** t)

33 {

34 month=r;

35 day=s;

36 year=t;

37 }

38

39 **void** date::printDate()

40 {

41 **cout** <<"Date: " <<month <<"/" <<day <<"/" <<year;

42 }

43

44 **int** main()

45 {

46 date dt1, dt2(2,3,2018);

47 **int** m,d,y;

48

49 **cout** <<**endl**;

50 dt1.printDate();

51 **cout** <<**endl**;

52

53 **cout** <<**endl**;

54 dt2.printDate();

55 **cout** <<**endl**;

56

57 **cout** <<**endl**;

58 **cout** << "Enter Month: " <<**endl**;

59 **cin** >> m;

60 **cout** << "Enter Date: " <<**endl**;

61 **cin** >> d;

62 **cout** << "Enter Year: " <<**endl**;

63 **cin** >> y;

64 **cout** <<**endl**;

65 dt1.setDate(m,d,y);

66 dt1.printDate();

67 **cout** <<**endl**;

68

69 **return** 0;

70 }

**Output:**

****

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**THE END**