National University of Computer and Emerging Sciences

Lab Manual
Object Oriented Programming

Course Instructor	Dr. Saira Karim
Lab Instructor (s)	Ms. Sonia Anum Ms. Mamoona Akbar
Section	BSCS 2A
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Copy Constructor
- ✓ Shallow vs deep copy constructor
- ✓ Overloaded function
- ✓ Operator overloading

Instructions

- ➢ Make sure memory is deallocated properly
- ➢ Make appropriate functions if you need them.

Task 01

Lamborghini is an international luxury sports car developer stationed in Italy. The company has a reputation for producing cars that are extremely expensive, powerful and rare. Lamborghini has developed a brand new model called the Diablo. The company produces a very limited numberof Diablo's each year. The company is producing the Diablo in only one colour called the "Hot Red". When the company has produced a Diablo, the car has a number of attributes like colour, cubic capacity, number of seats, year of manufacture, engine number, frame number and owner name. Out of these attributes the attributes that remain the same for all Diablo's being produced are colour, cubic capacity and number of seats.

Suppose you are working on a system specially designed for the Lamborghini Diablo. Follow the instructions below for creating the class and objects:

- Create default argument constructor.
- Create an object named "obj1" and initialize the object.
- Create a copy constructor that can copy all those attributes that remain the same for all cars.
- Create getter and setter functions to set and get the data of the data member
- Generate another object named "obj2" that is created by copying only those attributes that are the same from "obj1".
- Initialize the remaining attributes with values of your own.

Example input and output:

```
 Information about Obj1
Owner Name::abcd
Colour::White
Cubic Capacity::1000
Number of Seats::4
 Year::2012
 Engine Number::100
 Frame Number::5
Enter Name::xyz

 Enter Engine Number::100

 Enter frame Number::1234

 Enter year::2021

 Information about Obj2
Owner Name::xyz
Colour::White
Cubic Capacity::1000
Number of Seats::4
 Year::2021
 Engine Number::100
 Frame Number::1234
--------------------------------
```

Task 02:

a) Create a class Student
   - It has a data member integer array of 5 subjects marks
   - Create a default constructor that assigns dynamic memory to the data member
   - Make setter function named void set_marks(int marks,int index) that will receive marks and index location to set the marks(**note**: please validate the index location before setting the marks)
   - Create object student1 in the main and set the marks of five subjects by calling set_marks function through loop.
   - Create a function display( ) that will display the data of the objects
   - Now create another object student2 that will initialize with the existing object and display the data of both objects
   - Now make a destructor that will deallocate the dynamic memory of the data member
   - Deallocate the memory of student1 object and now display the data of object student2

   Now note the issue and give the reason why it is happening.

   Sample:

```
Info about object One
Data::20
Info about object two
Data::20
Now let's Modify the data of object 0ne
Now let's See the behavior of obj2 two
Info about object One
Data::30
Info about object two
Data::30
```

b) Once you have done **"a part"** and demonstrate the issue. The overload copy constructor and give the implementation of deep copy inside the overloaded copy constructor. Now call the display function to check whether the issue has been resolved or not.

Example:

```
Info about object One
Data::20
Info about object two
Data::20
Now let's Modify the data of object 0ne
Now let's See the behavior of obj2 two
Info about object One
Data::30
Info about object two
Data::20
--------------------------------
```

Task 03

Implement a class called **Complex**. The Complex class will have two data members:

- int real; // The real part of complex number
- int imaginary; // Imaginary part of the complex number.

You have to implement default constructor, overloaded constructor, copy constructor, destructor and overload the operators **+, - , <, >, ==,! =, prefix and postfix (addition and subtraction)**

Sample Run:

Driver.cpp	Output

```
int main()
{
	Complex C1;
	Complex C2(5,7);
	Complex C3(4,6);


	C1=C2+C3;

	C1=C2-C3;

	if(C1==C2)
		cout<<"C1 == C2"<<endl;
	else
		cout<<"C1 != C2"<<endl;

	if(C1!=C3)
		cout<<"C1 != C3"<<endl;
	else
		cout<<"C1==C3"<<endl;
	if(C2>C3)
		cout<<"C2 IS GREATER"<<endl;
	else
		cout<<"C3 IS GREATER"<<endl;
	if(C2<C3)
		cout<<"C2 is smaller"<<endl;
	else
		cout<<"C3 is smaller"<<endl;



	system("pause");
	return 0;
}
```

```
9 + 13i
C1 != C2
C1 != C3
C2 IS GREATER
C3 is smaller

Press any key to continue . . . . . .
```