	AIR UNIVERSITY
	DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING
	EXPERIMENT NO 3

Lab Title: Destructors, Parameterized Constructors and Copy Constructors

Student Name: _____ **Reg. No:** _____

Objective: _____

_____

LAB ASSESSMENT:

Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Ability to Conduct Experiment					
Ability to assimilate the results					
Effective use of lab equipment and follows the lab safety rules					

Total Marks: _____ Obtained Marks: _____

LAB REPORT ASSESSMENT:

Attributes	Excellent (5)	Good (4)	Average (3)	Satisfactory (2)	Unsatisfactory (1)
Data presentation					
Experimental results					
Conclusion					

Total Marks: _____ Obtained Marks: _____

Date: _____ Signature: _____

LAB TASK # 1

```
#include <iostream>
using namespace std;


class Invoice
{
int price,quantity;
string part_des,part_num;
public:
Invoice(int price_func,int quantity_func,string part_des,string part_num)
{
if (price<0) //Parametarized constructor
{
price=0;
}
if (quantity<0)
{
quantity=0;
}
cout<<"\n\nThe Price is: "<<price_func<<endl;
cout<<"The Quantity is: "<<quantity_func<<endl;
cout<<"The part description is: "<<part_des<<endl;
cout<<"The part number is: "<<part_num<<endl;
price=price_func;
quantity=quantity_func;
}
Invoice()
{
cout<<"\n\nEnter Price:";
cin>>price;
cout<<"\nEnter Quantity:";
cin>>quantity;
cout<<"\nEnter Part description:";
cin>>part_des;
cout<<"\nEnter Part number:";
cin>>part_num;
if (price<0)
{
price=0;
}
if (quantity<0)
{
quantity=0;
}
cout<<"\n\nThe Price is: "<<price<<endl;
```

```
cout<<"The Quantity is: "<<quantity<<endl;
cout<<"The part description is: "<<part_des<<endl;
cout<<"The part number is: "<<part_num<<endl;
}
int getInvoiceAmount()
{
return quantity*price;
}
~Invoice()
{
cout<<"\n\nDestructor called\n\n";
}
};
int main()
{
cout<<"\n\nEnter the details of hardware "<<endl;
int price_2=23,quantity_2=60;
string part_des_2="Silver",part_num_2="00EN700";
Invoice part_1,part_2(price_2,quantity_2,part_des_2,part_num_2);

/*Part 1 without arguments
and part 2 with arguments*/
int value_1 = part_1.getInvoiceAmount();
int value_2=part_2.getInvoiceAmount();
cout<<"\n\nThe invoice amount for part 1 will be: "<<value_1<<endl;
cout<<"\nThe invoice amount for part 2 will be: "<<value_2<<endl;
cout<<"\n\n\n";
return 0;

}
```

OUTPUT

```
Enter the details of hardware


Enter Price:23

Enter Quantity:3

Enter Part description:Alumunium

Enter Part number:2001A


The Price is: 23
The Quantity is: 3
The part description is: Alumunium
The part number is: 2001A


The Price is: 23
The Quantity is: 60
The part description is: Silver
The part number is: 00EN700


The invoice amount for part 1 will be: 69

The invoice amount for part 2 will be: 1380




Destructor called


Destructor called
```

LAB TASK #2

```
#include <iostream>
using namespace std;

class copy_concatenate
{
char x;
public:
copy_concatenate(const char *str)
{
x = *str;
}

copy_concatenate(const copy_concatenate &c1)
{
x=c1.x;
```

```
}

~copy_concatenate()
{
cout<<"\nDestructor called\n";
}
void display()
{
cout<<"The concatenated string will be : " <<x;
}
};

int main()
{
copy_concatenate obj1("S");
copy_concatenate obj2= obj1;
cout<<"\n\nFor object #1: ";
obj1.display();
cout<<"\n\nFor object #2: ";
obj2.display();
}
```

OUTPUT

```
For object #1: The concatenated string will be : S

For object #2: The concatenated string will be : S
Destructor called

Destructor called
```

Q: Can we set constructors private?
Yes we can set constructors private but it is not recommended to set constructors private.

Q: Why is copy constructor needed?
When we want to create more than one same objects from a class, then at that time copy constructors are needed. So that we won't need to pass the same parameters again and again.

Q: SavingAccount g=v; results in call of which constructor?
If we pass this in int main(), then a copy constructor will be called which will assign all the values of object v to object g.

Q: Why copy constructor receives a constant object of same class as an argument?
This is because it creates the copy of the object and that why the copy constructor receives a constant object of same class as an argument.

Conclusion:

I learned how to create and use copy constructors, destructors in classes. I also learned how to create more then one constructors and how to create parameterized constructors.