Experiment No 03

Aim: - Create constructor and destructor for Laptop Customization System.

Objectives:-

1) To understand the concept of constructors and destructor.

2) To identify the difference types of constructors.

Theory:-

What is constructor?

A constructor is a member function of a class which initializes objects of a class. In C++, Constructor is automatically called when object (instance of class) create. It is special member Function of the class.

How constructors are different from a normal member function?

A constructor is different from normal functions in following ways:

- Constructor has same name as the class itself.
- Constructors don't have return type
- A constructor is automatically called when an object is created.

If we do not specify a constructor, C++ compiler generates a default constructor for us (expects no parameters and has an empty body).

⇨ The constructor functions have some special characteristics. These are:
  - They should be declared in the public section.
  - They are invoked automatically when the objects are created.
  - They do not have return types, not even void and therefore, and they cannot return values.
  - They cannot be inherited, though a derived class can call the base class constructor.
  - Like other C++ functions, they can have default arguments.
  - Constructors cannot be virtual.
  - We cannot refer to their addresses.
  - An object with a constructor (or destructor) cannot be used as a member of a union.

- They make 'implicit calls' to the operators new and delete when memory allocation is required.

Types of Constructors:

1. Default constructor

2. Parameterized constructor

3. Copy constructor

1. Default Constructors: Default constructor is the constructor which doesn't take any Argument. It has no parameters.

```
// Cpp program to illustrate the concept of Constructors
#include <iostream>
using namespace std;
class construct {
public:
int a, b;
// Default Constructor
construct()
{
a = 10;
b = 20;
}
};
int main()
{
// Default constructor called automatically when the object is created
construct c;
cout << "a: " << c.a << endl
<< "b: " << c.b;
return 1;
```

```
}
```

Output:

a: 10

b: 20

Note: Even if we do not define any constructor explicitly, the compiler will automatically Provide a default constructor implicitly.

2. Parameterized Constructors: It is possible to pass arguments to constructors. Typically, these Arguments help initialize an object when it is created. To create a parameterized constructor, Simply add parameters to it the way you would to any other function. When you define the Constructor's body, use the parameters to initialize the object.

```
// CPP program to illustrate parameterized constructors
#include <iostream>
using namespace std;
class Point {
private:
int x, y;
public:
// Parameterized Constructor
Point(int x1, int y1)
{
x = x1;
y = y1;
}
int getX()
{
return x;
```

```
}
int getY()
{
return y;
}
};
int main()
{
// Constructor called
Point p1(10, 15);
// Access values assigned by constructor
cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();
return 0;
}
```

Output:

p1.x = 10, p1.y = 15

When an object is declared in a parameterized constructor, the initial values have to be passed as

arguments to the constructor function. The normal way of object declaration may not work. The

constructors can be called explicitly or implicitly.

Example e = Example(0, 50); // Explicit call

Example e(0, 50); // Implicit call

Uses of Parameterized constructor:

- It is used to initialize the various data elements of different objects with different values when they are created.
- It is used to overload constructors.

3. Copy Constructor:

A copy constructor is a member function which initializes an object using another object of the same class. A copy constructor has the following general function prototype:

```
ClassName (const ClassName &old_obj);
```

Following is a simple example of copy constructor.

```
#include<iostream>

using namespace std;

class Point

{

private:

int x, y;

public:

Point(int x1, int y1) { x = x1; y = y1; }

// Copy constructor

Point(const Point &p2) {x = p2.x; y = p2.y; }

int getX() { return x; }

int getY() { return y; }

};

int main()

{

Point p1(10, 15); // Normal constructor is called here

Point p2 = p1; // Copy constructor is called here

// Let us access values assigned by constructors

cout << "p1.x = " << p1.getX() << ", p1.y = " << p1.getY();

cout << "\np2.x = " << p2.getX() << ", p2.y = " << p2.getY();

return 0;

}
```

Output:

p1.x = 10, p1.y = 15

p2.x = 10, p2.y = 15

Destructors:

What is destructor?

Destructor is a member function which destructs or deletes an object.

- A destructor, as the name implies is used to destroy the object that have been created by a constructor.
- Like a constructor, the destructor is a member function whose name is the same as the class name but is preceded by a tilde.
- For example, the destructor for the class integer can be defined as shown below: ~SY ( ) { }

Example:

```
#include<iostream>

using namespace std;

class SY { // The class

public: // Access specifier

SY() { // Constructor

cout << "SY IT!"<<endl;

}

~SY(){

cout<<"destroyed the string";

};

};

int main() {

SY S;

}
```

Output:

SY IT!

Destroyed the string

Constructor details for Laptop Customization System:

- User () {} default constructor is created which takes input for users name and last name and then displays the full name of a user.

When is destructor called?

A destructor function is called automatically when the object goes out of scope:

1. The function ends
2. The program ends
3. A block containing local variables ends
4. A delete operator is called.

How destructors are different from a normal member function?

Destructors have same name as the class preceded by a tilde (~)

Destructors don't take any argument and don't return anything.

Source Code for Laptop Customization System using constructor:

```
#include<iostream>
#include<stdio.h>

// Assignment-3

/* Here we have used the Constructor Features which takes input name and
   last name and displays full name.
   In this part of project we have used the default constructor */
using namespace std;
```

```
class User
{
private:
   string name;
   string lastname;
   string fn;

 public:
     User()//default constructor used
     {
         cout<<"\n * Enter the First name : ";
         cin>>name;
         cout<<"\n * Enter the lastname : ";
         cin>>lastname;
     }
     string fullname()
     {
        fn=name+" "+lastname;
        return fn;
     }
     void display()
    {
         cout<<"\n\n **Hello "<<fn<<" Welcome to our Program !!!!"<<endl;
    }
};
class Features : public User //class named Features
{
   int os;
    int processor;
    int ram;
public:
```

```
    void select_os();
    void select_processor();
    void select_ram();
    void display_features();
};
void Features::select_os() // member function
{
    cout<<"\n ** Please Select The Operating system you want in your Laptop : \n"<<endl;
    cout<<"\n 1.Windows "<<endl;
    cout<<"\n 2.MAC Os "<<endl;
    cout<<"\n 3.Exit"<<endl;
    cout<<"\n Enter Your Choice : ";
    cin>>os;
    switch(os)
    {
    case 1:
        cout<<"\n * You have Selected Your Os ==> windows Os"<<endl;
        break;

    case 2:
        cout<<"\n * You have Selected Your Os ==> MAC Os"<<endl;
        break;

    case 3:
        cout<<"\n Terminated Successfully !!"<<endl;
        cout<<"\n Thank You Visit Again !!!!\n"<<endl;
        exit(0);
        break;

    default :
```

```
        cout<<"\n Enter valid choice ";
        exit(0);
    }
}
void Features::select_processor()
{
if(os==1){
    cout<<"\n\n **Select the Processor : \n"<<endl;
    cout<<"1. Intel i3"<<endl;
    cout<<"2. Intel i5"<<endl;
    cout<<"3. Intel i7"<<endl;
    cout<<"4. Intel i9"<<endl;
    cout<<"5. AMD Ryzen 5"<<endl;
    cout<<"6. AMD Ryzen 7"<<endl;
    cout<<"7. Exit"<<endl;
    cout<<"\n Enter Your Choice : ";
    cin>>processor;
    switch(processor)
    {
    case 1:

        cout<<"\n * You have Selected Your Processor ==> Intel i3 "<<endl;
        break;
    case 2:

        cout<<"\n * You have Selected Your Processor ==> Intel i5 "<<endl;
        break;
    case 3:

        cout<<"\n * You have Selected Your Processor ==> Intel i7 "<<endl;
        break;
```

```
case 4:

   cout<<"\n * You have Selected Your Processor ==> Intel i9 "<<endl;
   break;
case 5:

   cout<<"\n * You have Selected Your Processor ==> AMD Ryzen 5 "<<endl;
   break;
case 6:

   cout<<"\n * You have Selected Your Processor ==> AMD Ryzen 7 "<<endl;
   break;
case 7:
   cout<<"\n Terminated Successfully !!"<<endl;
   cout<<"\n Thank You Visit Again !!!!\n"<<endl;
   exit(0);
   break;

default :
   cout<<"\n Enter valid choice ";
   exit(0);
}
}

else if(os==2)
{
cout<<"\n\n **Select the Processor : \n"<<endl;
cout<<"1. Intel i3"<<endl;
cout<<"2. Intel i5"<<endl;
cout<<"3. Intel i7"<<endl;
cout<<"4. Intel i9"<<endl;
```

```
cout<<"5. Exit"<<endl;
cout<<"\n Enter Your Choice : ";
cin>>processor;
switch(processor)
{
case 1:
   cout<<"\n * You have Selected Your Processor ==> Intel i3 "<<endl;
   break;

case 2:
   cout<<"\n * You have Selected Your Processor ==> Intel i5 "<<endl;
   break;

case 3:
   cout<<"\n * You have Selected Your Processor ==> Intel i7 "<<endl;
   break;

case 4:
   cout<<"\n * You have Selected Your Processor ==> Intel i9 "<<endl;
   break;

case 5:
   cout<<"\n Terminated Successfully !!"<<endl;
   cout<<"\n Thank You Visit Again !!!!\n"<<endl;
   exit(0);
   break;

default :
   cout<<"\n Enter valid choice ";
   exit(0);
}
```

```
    }
}

void Features::select_ram()
{

if(os==1){
    cout<<"\n\n **Select the RAM : \n"<<endl;
    cout<<"1. 4 GB"<<endl;
    cout<<"2. 8 GB"<<endl;
    cout<<"3. 16 GB"<<endl;
    cout<<"4. 32 GB"<<endl;
    cout<<"5. Exit"<<endl;
    cout<<"\n Enter Your Choice : ";
    cin>>ram;
    switch(ram)
    {
    case 1:

        cout<<"\n * You have Selected Your RAM ==> 4 GB "<<endl;
        break;
    case 2:

       cout<<"\n * You have Selected Your RAM ==> 8 GB "<<endl;
        break;
    case 3:

        cout<<"\n * You have Selected Your RAM ==> 16 GB "<<endl;
        break;
    case 4:
```

```
        cout<<"\n * You have Selected Your RAM ==> 32 GB "<<endl;
        break;
    case 5:
        cout<<"\n Terminated Successfully !!"<<endl;
        cout<<"\n Thank You Visit Again !!!!\n"<<endl;
        exit(0);
        break;
    default :
        cout<<"\n Enter valid choice ";
         exit(0);
    }
}

if(os==2)
{
    cout<<"\n\n **Select the RAM : "<<endl;
    cout<<"1. 8 GB"<<endl;
    cout<<"2. 16 GB"<<endl;
    cout<<"3. Exit"<<endl;
    cout<<"\n Enter Your Choice : ";
    cin>>ram;
    switch(ram)
    {
    case 1:

       cout<<"\n * You have Selected Your RAM ==> 8 GB "<<endl;
        break;
    case 2:

        cout<<"\n * You have Selected Your RAM ==> 16 GB "<<endl;
        break;
```

```
    case 3:
        cout<<"\n Terminated Successfully !!"<<endl;
        cout<<"\n Thank You Visit Again !!!!\n"<<endl;
        exit(0);
        break;
    default :
        cout<<"\n Enter valid choice ";
         exit(0);
    }
}
}

void Features::display_features()
{
    cout<<"\n\n";
    cout<<"\n * Your Selected Specifications are ==> \n"<<endl;

 //OS
    if(os==1){
        cout<<"\n Operating System ==> Windows Os"<<endl;



    //Processor
    if(processor==1)
        cout<<"\n Processor ==> Intel i3"<<endl;
    else if(processor==2)
        cout<<"\n Processor ==> Intel i5"<<endl;
    else if(processor==3)
        cout<<"\n Processor ==> Intel i7"<<endl;
    else if(processor==4)
```

```
        cout<<"\n Processor ==> Intel i9"<<endl;
    else if(processor==5)
        cout<<"\n Processor ==> Ryzen 5"<<endl;
    else if(processor==6)
        cout<<"\n Processor ==> Ryzen 7"<<endl;

     //RAM
    if(ram==1)
        cout<<"\n RAM ==> 4 GB"<<endl;
    else if(ram==2)
        cout<<"\n RAM ==> 8 GB"<<endl;
    else if(ram==3)
        cout<<"\n RAM ==> 16 GB"<<endl;
    else if(ram==4)
        cout<<"\n RAM ==> 32 GB"<<endl;
}
    //for mac
    else if(os==2){
        cout<<"\n Operating System ==> MAC Os "<<endl;
    if(processor==1)
        cout<<"\n Processor ==> Intel i3"<<endl;
     else if(processor==2)
        cout<<"\n Processor ==> Intel i5"<<endl;
     else if(processor==3)
        cout<<"\n Processor ==> Intel i7"<<endl;
     else if(processor==4)
        cout<<"\n Processor ==> Intel i9"<<endl;

     //ram mac
    if(ram==1)
       cout<<"\n RAM ==> 8 GB"<<endl;
```

```
    else if(ram==2)
      cout<<"\n RAM ==> 16 GB"<<endl;
    }
}
int main()
{
    cout<<"\t\t\t\t\t\t**********************************"<<endl;
    cout<<"\t\t\t\t\t\t*   Laptop Customization System   *"<<endl;
    cout<<"\t\t\t\t\t\t**********************************"<<endl;
    Features f; //object created
    f.fullname();
    f.display();
    f.select_os();
    f.select_processor();
    f.select_ram();
    system("cls");
    f.display_features();
    cout<<"\n\n * Thank you Visit Again :) !!!! "<<endl;
}
```

Output:

```
            **********************************
            *        Laptop Customization System        *
            **********************************

 * Enter the First name : Piyush
 * Enter the lastname : Sonawane
```

**Hello Piyush Sonawane Welcome to our Program !!!!

** Please Select The Operating system you want in your Laptop :

1.Windows

2.MAC Os

3.Exit

Enter Your Choice : 1

* You have Selected Your Os ==> windows Os

**Select the Processor :

1. Intel i3
2. Intel i5
3. Intel i7
4. Intel i9
5. AMD Ryzen 5
6. AMD Ryzen 7
7. Exit

Enter Your Choice : 3

* You have Selected Your Processor ==> Intel i7

**Select the RAM :

1. 4 GB
2. 8 GB
3. 16 GB
4. 32 GB
5. Exit

Enter Your Choice : 2

* Your Selected Specifications are ==>

Operating System ==> Windows Os

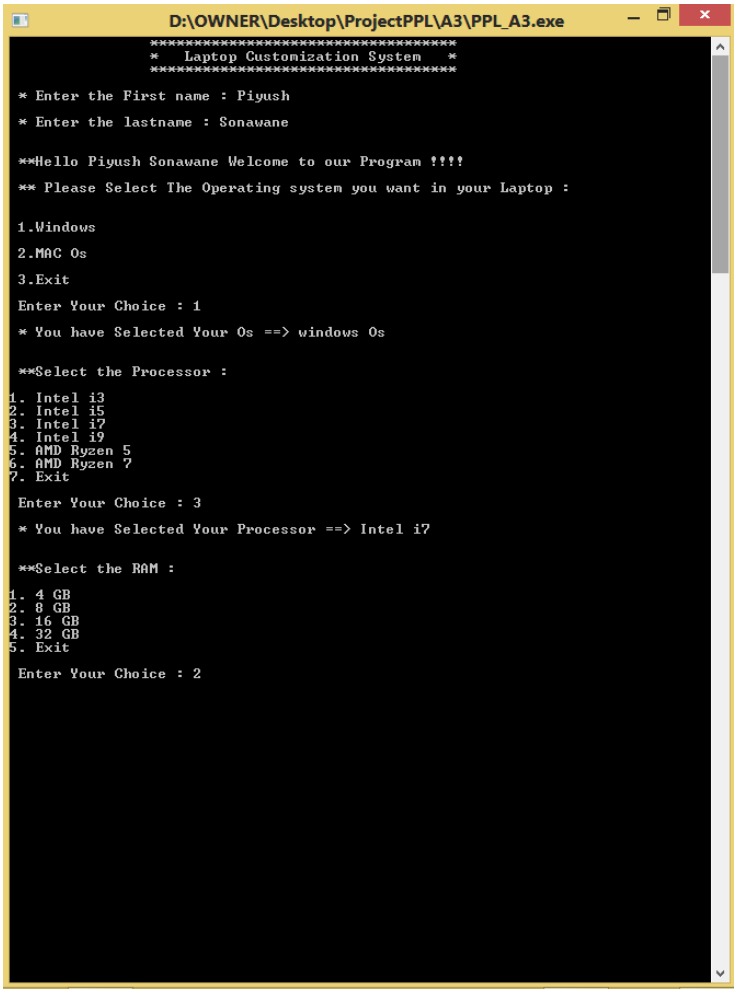
Processor ==> Intel i7

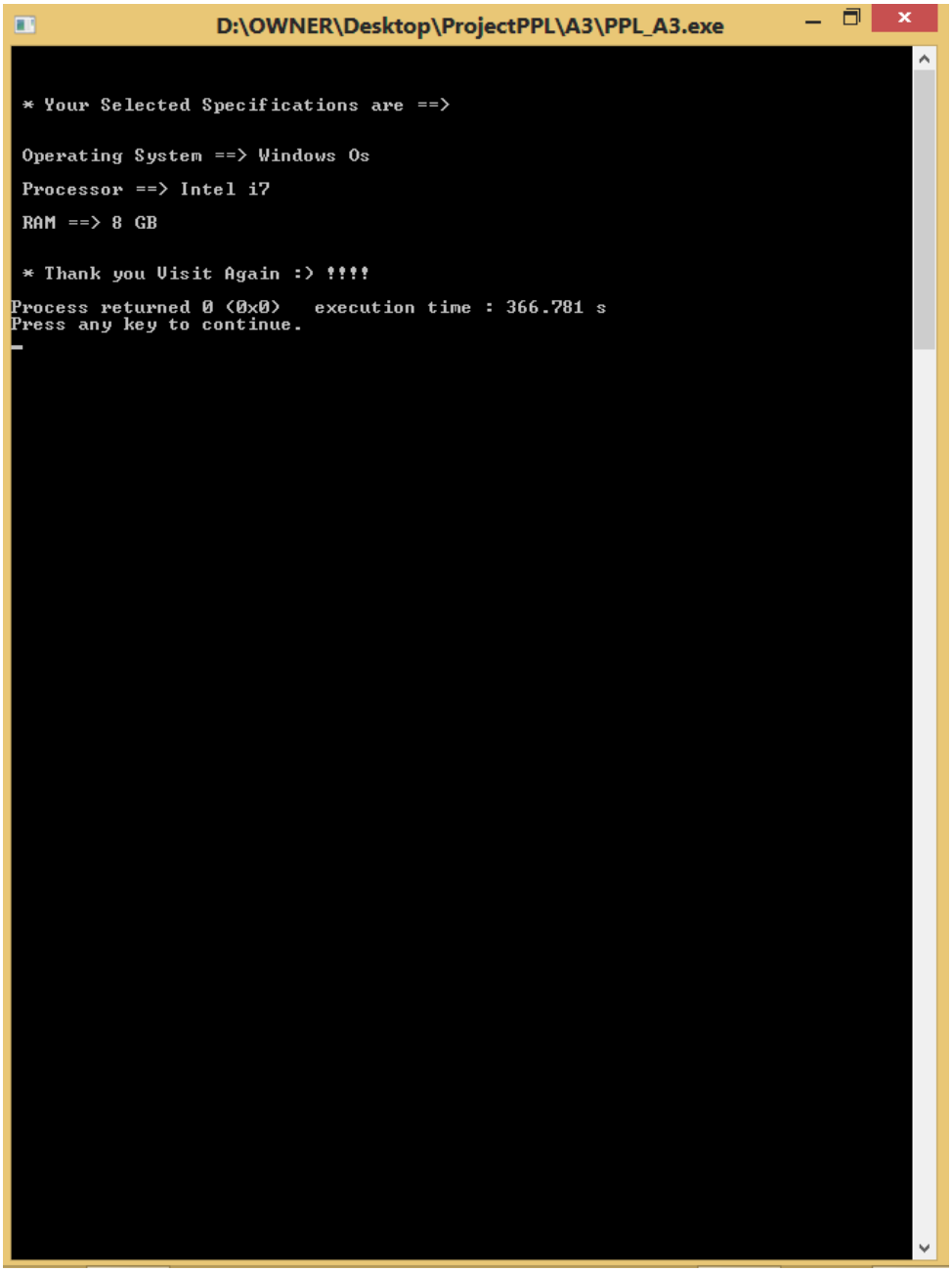
RAM ==> 8 GB

* Thank you Visit Again :) !!!!

Process returned 0 (0x0) execution time : 25.369 s

Press any key to continue.





Conclusion: Thus we studied and implemented Laptop Customization System using Concept of Constructor.