Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	3
Student Name:	Siddhant Kedar
Roll No :	E-05

Title: Write a program to demonstrate default, overloading and copy constructors and application of destructor.

Learning Objective:

- Students will be able implement constructor and destructor in C++.

Learning Outcome:

- Understanding the default, overloading and copy constructor in C++.

Theory:

Constructor: 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.

1) Default Constructor: Default constructor is the constructor which doesn't take any argument. It has no parameter.

2) Copy Constructor: These are special type of Constructors which takes an object as argument, and is used to copy values of data members of one object into other object.

3) Constructor overloading: Just like other member functions, constructors can also be overloaded. Infact when you have both default and parameterized constructors defined in your class you are having Overloaded Constructors, one with no parameter and other with parameter.

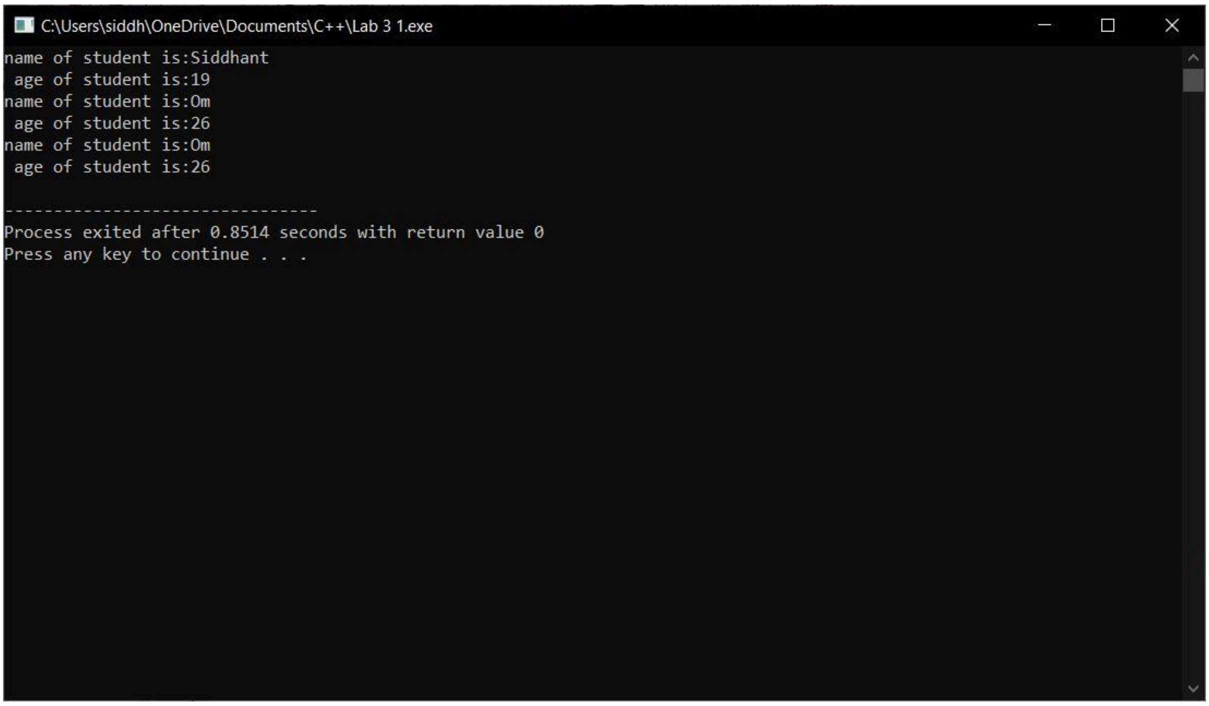
Destructor: Destructors in C++ are members functions in a class that delete an object. They are called when the class object goes out of scope such as when the function ends, the program ends, a delete variable is called etc.

Destructors are different from normal member functions as they don’t take any argument and don’t return anything.

Program 1: Default, overload and copy constructor

```
#include <iostream>

using namespace std;

#include <string.h>


class student{

    private:

        char name[20];

        int age;

    public:

        student(){};

        student(char *n)

        {

            strcpy(name,n);

            age=0;

        }

        student(char *n, int a)

        {

            strcpy(name,n);

            age=a;

        }

        student(student &s)

        {

            strcpy(name,s.name);

            age=s.age;

        }

        void show();

};


void student:: show()

{

    cout<< "name of student is:"<<name<<endl;
    cout<<" age of student is:"<<age<<endl;
}


int main()
{
    student s2("Siddhant",19);
    student s3("Om",26);
    student s4(s3);
    s2.show();
    s3.show();
    s4.show();
    return 0;
}
```

Output:



```
name of student is:Siddhant
 age of student is:19
name of student is:Om
 age of student is:26
name of student is:Om
 age of student is:26

--------------------------------
Process exited after 0.8514 seconds with return value 0
Press any key to continue . . .
```

Program 2:

Algorithm:

1) Start

2) Create a class called Car

3) Declaring attributes brand, model and year

4) Declaration of constructor

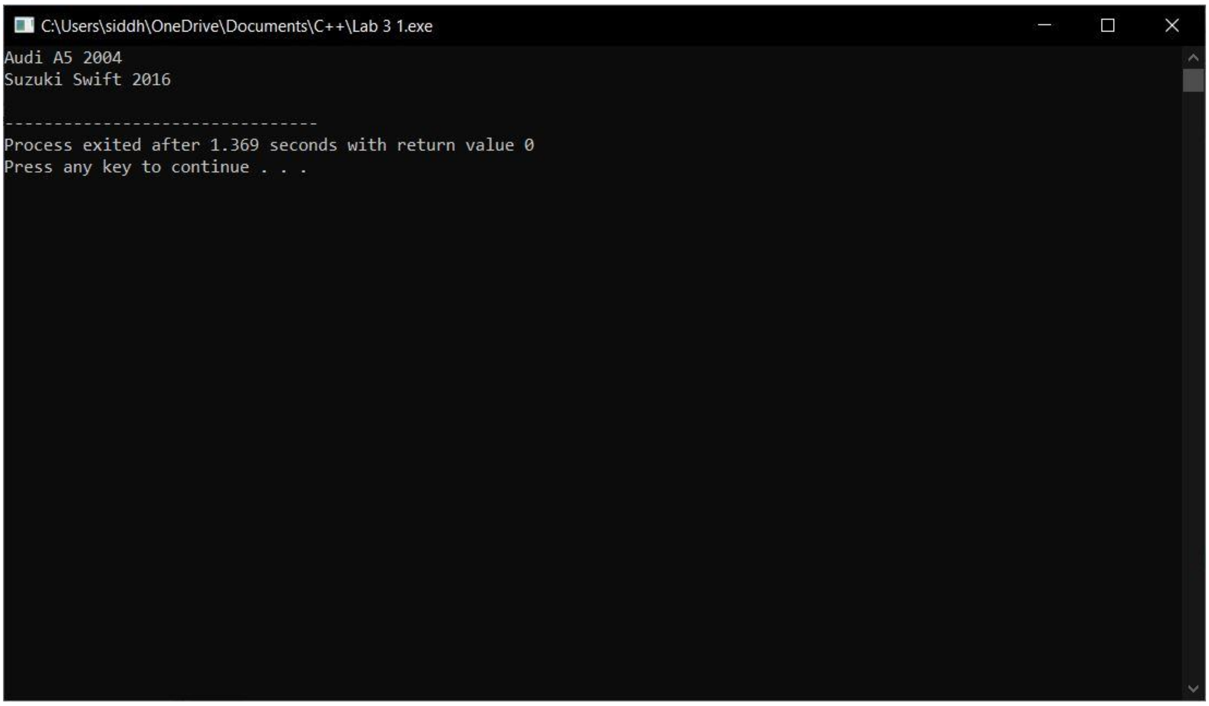
5) Defining constructor outside the class

6) Displaying output

7) End

```
#include <iostream>
using namespace std;
class Car {        // The class
  public:          // Access specifier
    string brand;  // Attribute
    string model;  // Attribute
    int year;      // Attribute
    Car(string x, string y, int z); // Constructor declaration
    ~Car(){
    }
};
// Constructor definition outside the class
Car::Car(string x, string y, int z) {
  brand = x;
  model = y;
  year = z;
}
int main() {
  // Create Car objects and call the constructor with different values
  Car carObj1("Audi", "A5", 2004);
  Car carObj2("Suzuki", "Swift", 2016);
 cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
  cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
  return 0;
}
```

Output:



```
Audi A5 2004
Suzuki Swift 2016

--------------------------------
Process exited after 1.369 seconds with return value 0
Press any key to continue . . .
```