Maharashtra State Board Of Technical Education

Government Polytechnic, Gondia

DEPARTMENT OF COMPUTER ENGINEERING

(2018-19)

MICRO PROJECT REPORT

ON

"CONSTRUCTOR AND ITS TYPE"

SUBMITTD BY:

- NAINA P. BANSOD
- BHUSHAN G. ASATI
- ROSHNI T. THAKUR
- KHUSHAL M. GANDHE
- TEJAS G. FASATE

Guided by : **Mr. A. G. Barsagade**

Head of Department : **Mr. J. M. Meshram**

Principal :**Dr. R. R. Wakodkar**

Certificate

This is to certify that the

“Micro-project title’’

“CONSTRUCTOR AND ITS TYPE“

Carried out under Guidance of **Mr. A. G. Barsagade Sir** lecturer Computer department and submitted to the department of Computer Engineering.

Name of Student:

- **NAINA P. BANSOD**
- **BHUSHAN G. ASATI**
- **ROSHNI T. THAKUR**
- **KHUSHAL M. GANDHE**
- **TEJAS G. FASATE**

As the Partial fulfillment of third subject Course of **“Object Oriented Programming Using C++**’’ During winter 2018.

Mr. A. G. Barsagade

Guide and lecturer:

Mr. J. M. Meshram

Head of Department:

Department of Computer Engineering Government Polytechnic, Gondia (Winter 2018)

Declaration

We are the student of Computer engineering hereby declare that the micro Project tittle

“CONSTRUCTOR AND ITS TYPE“

Submitted to the Computer engineering department GPG for the practical work of **“Object Oriented Programming Using C++’’**subject **(22316)** that the micro project has not previously formed the basic of any copyright work.

Submitted by:

Name of the student	Enrollment No	Signature
1. Naina P. Bansod	1612420071	
2. Bhushan G. Asati	1612420073	
3. Roshni T. Thakur	1612420077	
4. Khushal M. Gandhe	1612420078	
5. Tejas G. Fasate	1612420084	

Acknowledgement

This is completed micro project work comes as gift to us after all the efforts that has gone in to it has been a beautiful end over only because of the valuable guidance of our guide and well Wishers.…..

We wish extend our heart full gratitude of our guide **Mr. A. G. Barsagade** for his constant guidance , encouragement ,motivation for every stage of this work made this micro project a success…..

Finally we are proud to express our gratitude and respect to each member of this group……

Place: Government Polytechnic Gondia

Date:

Content

Introduction

- In class-based object-oriented programming, a constructor (abbreviation: ctor) is a special type of subroutine called to create an object. It prepares the new object for use, often accepting arguments that the constructor uses to set required member variables.
- A constructor resembles an instance method, but it differs from a method in that it has no explicit return type, it is not implicitly inherited and it usually has different rules for scope modifiers. Constructors often have the same name as the declaring class.
- They have the task of initializing the object's data members and of establishing the invariant of the class, failing if the invariant is invalid. A properly written constructor leaves the resulting object in a *valid* state. Immutable objects must be initialized in a constructor.
- Most languages allow overloading the constructor in that there can be more than one constructor for a class, with differing parameters. Some languages take consideration of some special types of constructors.
- Constructors, which concretely use a single class to create objects and return a new instance of the class, are abstracted by factories, which also create objects but can do so in various ways, using multiple classes or different allocation schemes such as an object pool.

What is the Purpose of a used Constructor in CPP?

- The constructor in C++ is a function that runs when an object is created. It is used to initialize the object.
- Types of constructors include default constructors (no arguments), copy constructor (one argument of same type as object), conversion constructors (one argument of some other type), and other constructors (all other cases).
- If you do not provide a default constructor, the object will not be initialized. If you do not provide a copy constructor, the compiler will blindly copy the attributes of the old object into the new object whenever a copy is made, such as in a function call with the object as an argument.
- This may or may not be safe, especially if any of the attributes are pointers, because that creates the situation of two pointers to the same region of memory. In that case, if that region of memory is an object, then when the object is destroyed, so will the pointed to object, and that will leave the original copied object in an invalid state, with its pointers referencing deleted memory.

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.
- Compiler identifies a given member function is a constructor by its name and the return type.
- Constructor has the same name as that of the class and it does not have any return type. Also, the constructor is always public.
- Constructors can be very useful for setting initial values for certain member variables.

Types of constructor:-

- Default Constructor
- Parameterized Constructor
- Copy Constructor

Default Constructor

- A constructor which does not expect any orgument and parameter is called default constructor.
- A default constructor is also called empty constructor because it does not have any orgument so it initialization or the object of class with same value in no default default constructor is define in the class then compiler while create a default constructor by its self with same initialization.
- 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.

Example:-

```
#include<iostream.h>
#include<conio.h>
Class number
{
int m,n;
Public:
number(int a, int b);
void display();
};
number :: number(int a, int b)
{
```

```
m=a;

n=b;

}

Void number :: void display()

{

Cout<<”m=”<<m;

Cout<<”n=”<<n;

}

void main()

{

Number n1(10,20);

n1.display();

getch();

}
```

Output:

```
M=10

N=20
```

Parameterized Constructor

- The constructor that can take orguments are called parameters. Parameterizes constructor when constructor has been parameterizes then object declaration much be different.
- 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.

1. Implicit call
2. Explicit way

Implicit call :- syntax :- number n1(10,20);

Explicit way :- syntax :- number n1=number (10,20);

Uses of Parameterized constructor:

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

Example:-

```
#include<iostream.h>

#include<conio.h>

class Point

{

   private:

   int x, y;

   public:

   Point(int x1, int y1)
```

```
    {
      x = x1;
      y = y1;
     }


      int getX()
      {
         return x;
      }
      int getY()
      {
         return y;
      }
   };
   void main()
{
   Point p1(10, 15);
   cout << "x = " << p1.getX()<<endl;
   cout<< ",y = " << p1.getY()<<endl;
   getch();
}
```

Output :

```
X=10  Y=15
```

Copy Constructor

- A constructor that initialization an object using values of another object pass two its as a parameter is called copy constructor.
- In CPP a new object of class can also be initialize with exiting object of same class its means a copy constructor is defined as a constructor which expect and all ready exacting object through reference of copy of data member value.

When is copy constructor called?
In C++, a Copy Constructor may be called in following cases:
1. When an object of the class is returned by value.
2. When an object of the class is passed (to a function) by value as an argument.
3. When an object is constructed based on another object of the same class.
4. When the compiler generates a temporary object.

Syntax

Class_name obj_name=obj_name;

Example

```
#include<iostream.h>

#include<conio.h>

Class student

{

int roll no;

float perc;

public:

student()

{

roll no=10;

perc=70;
}
Student(student & s)
{
roll no=s.roll no;
percent=s perc;
}
Void display()
{
Cout<<”roll no=”<<roll no;
Cout<<”perc=”<<perc;
}
};
Void main()
{
Student s1;
S1 display();
Student s2(s1);
S2 display();
Student s3=s1;
S3 display();
getch();
}
```

Output:

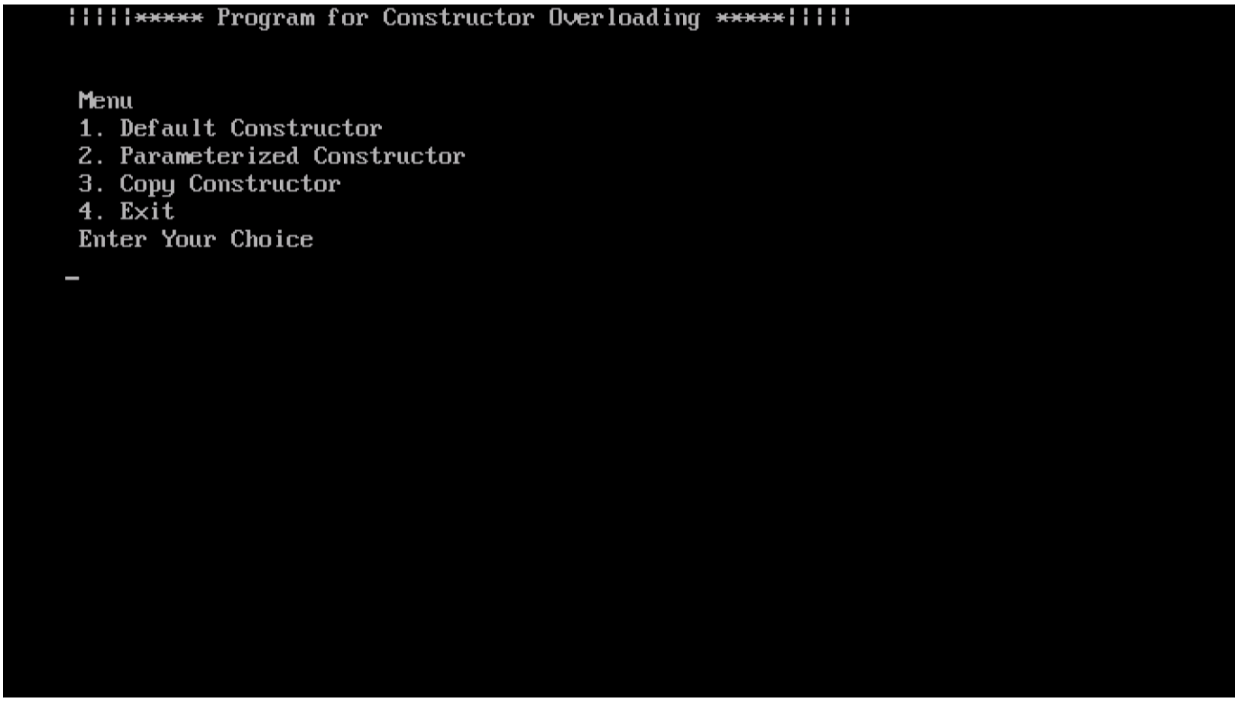
```
S1 Roll no 10
    Perc   70
S2 Roll no 10
     Perc 70
S3 Roll no 10
     Perc 70
```

Source Code

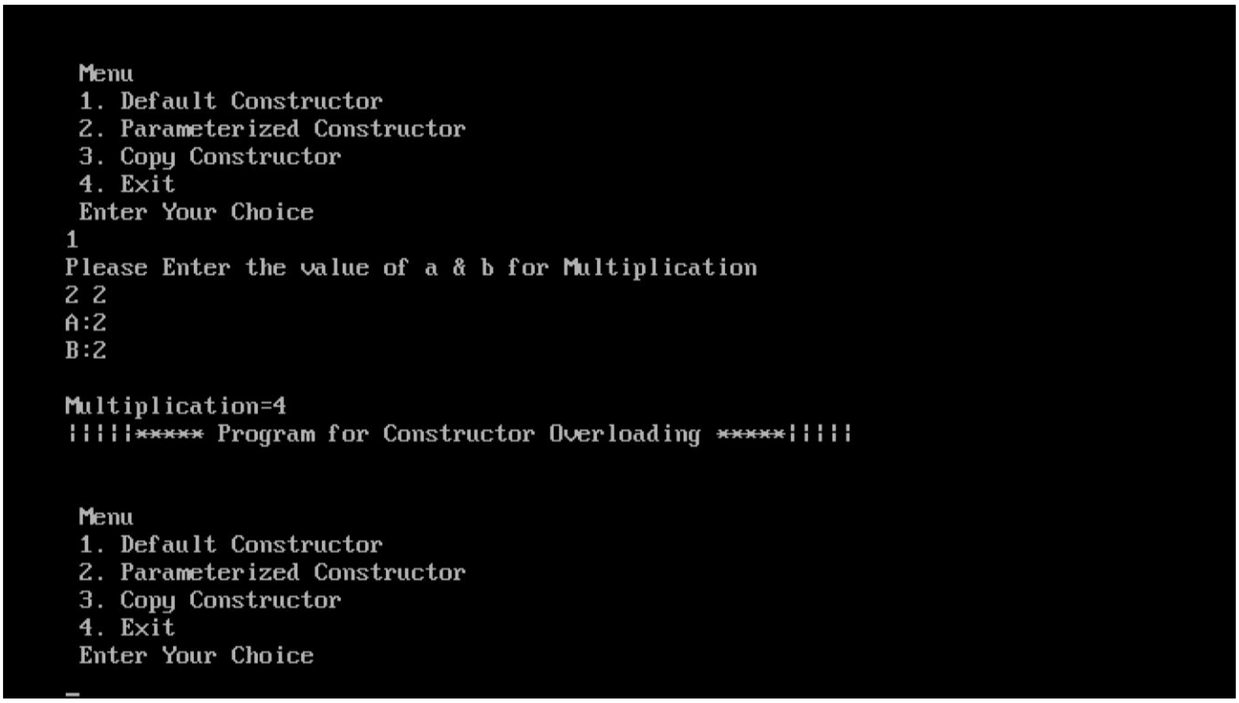
```
#include<iostream.h>
#include<conio.h>
class example
{
int a, b,c,d;
public:
 example()
 {
  cout<<"Please Enter the value of a & b for Multiplication "<<endl;
  cin>>a>>b;

  cout<<"A:"<<a<<endl;
  cout<<"B:"<<b<<endl;
  cout << "\nMultiplication="<<a*b<<endl;
  }
 example(int x, int y)
  {
  c = x;
  d= y;
  cout << "\n Multiplication"<<c*d<<endl;
  }
  example (example & e)
  {
  a=e.a;
  b=e.b;
  cout<<"Multiplication"<<a*b<<endl;
  }
  void display()
  {
```

```
cout << "C:" << a << endl;
cout <<"D:"<<b<<endl;
}
};

void main()
{
int choice;
clrscr();
do
{
cout<<"|||||***** Program for Constructor Overloading *****|||||"<<endl;
cout<<endl<<endl;
cout<<" Menu"<<endl;
cout<<" 1. Default Constructor "<<endl;
cout<<" 2. Parameterized Constructor "<<endl;
cout<<" 3. Copy Constructor "<<endl;
cout<<" 4. Exit "<<endl;
cout<<" Enter Your Choice "<<endl;
cin>>choice;
switch(choice)
{
case 1:
  example e1;
  break;
case 2:
  example e(10, 20);
  break;
case 3:
  e1.display();
  example e3(e1);
  break;
case 4:
  goto Bhushan;
  break;
default:
cout<<"Your Choice is Wrong"<<endl;
break;
}
}while(choice<=4);
Bhushan:
getch();
}
```

Output :-



```
    ¦¦¦¦¦***** Program for Constructor Overloading *****¦¦¦¦¦


     Menu
     1. Default Constructor
     2. Parameterized Constructor
     3. Copy Constructor
     4. Exit
     Enter Your Choice
    _
```



```

     Menu
     1. Default Constructor
     2. Parameterized Constructor
     3. Copy Constructor
     4. Exit
     Enter Your Choice
    1
    Please Enter the value of a & b for Multiplication
    2 2
    A:2
    B:2

    Multiplication=4
    ¦¦¦¦¦***** Program for Constructor Overloading *****¦¦¦¦¦


     Menu
     1. Default Constructor
     2. Parameterized Constructor
     3. Copy Constructor
     4. Exit
     Enter Your Choice
    _
```

Conclusions

1. We identify the various constructor used in program.
2. We use of constructor in a program succesfully.
3. When we call the object, then constructor is executed…..
4. A class can include a special function called a constructor which is automatically called whenever a new object of this class is created.
5. This constructor, function must have the same name as the class and cannot have any return type.

Reference

- The used OOPS (text max) books.
- And Refers Notes
- And Used Internet
- https://en.wikipedia.org/wiki/Constructor_(object-oriented_programming)#Types
- http://www.answers.com/Q/What_is constructor_in_c_plus_plus