

## Experiment No 5

**Aim:** Implementing constructors in CPP

**Theory:** To customize how a class initializes its members, or to invoke functions when an object of your class is created, define a *constructor*. A constructor has the same name as the class and no return value. You can define as many overloaded constructors as needed to customize initialization in various ways. Typically, constructors have public accessibility so that code outside the class definition or inheritance hierarchy can create objects of the class. But you can also declare a constructor as protected or private. Constructors can optionally take a member initializer list. It's a more efficient way to initialize class members than assigning values in the constructor body.

**Definition:** A constructor is a special member function that is called automatically when an object is created.

**Syntax:**

```
class CLASS_NAME
{
    .....
public :
    CLASS_NAME([parameter_list]) //constructor definition
{
    .....
}

    //other member functions
};
```

**Defining out of class:**

```
class CLASS_NAME
{
    .....
public :
    CLASS_NAME([parameter_list]) //constructor Declaration
{
    .....
}

    //other member functions
};
```

```
CLASSNAME: :CLASSNAME([parameter_list]) //Constructor Definition
{
    .....
}
```

}

### Points to remember about constructors in C++:

- Constructors invoke simultaneously whenever an object of a class creates.
- The compiler will automatically generate a default constructor if no constructor is defined explicitly.
- Preferably, constructors use to declare the data members(variables). Usually, not utilized for generating input and output.
- They also allocate memory at run time using the 'new' operator in C++.
- More than one constructor can be declared as well. It's "Constructor Overloading."

### Constructors types in C++

There are 3 types of Constructors in C++. Let's learn about them.

#### 1. Default Constructor

A **constructor with no arguments** (or parameters) in the definition is a default constructor. It is the type of constructor in C++ usually used to initialize data members (variables) with real values.

Note: The compiler automatically creates a default constructor without data member (variables) or initialization if no constructor is explicitly declared.

#### Syntax:

```
class CLASS_NAME
{
    .....
    public :
    CLASS_NAME() //Default constructor
    {
        .....
    }
    //other member functions
};
```

#### 2. Parameterized Constructor

Unlike the Default constructor, It contains parameters (or arguments) in the constructor definition and declaration. More than one argument can also pass through a parameterized constructor.

Moreover, this type of constructor in C++ is used for overloading to differentiate between multiple constructors.

#### Syntax:

```
class class_Name
```

```
{  
public:  
    class_Name(datatype variable) //Parameterized constructor  
    {  
        .....  
    }  
};
```

### 3. Copy Constructor

A copy constructor is the third type among various types of constructors in C++. The member function initializes an object using another object of the same class. It helps to copy data from one object to another.

Example:

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

```
// class name: Rectangle
```

```
class Rectangle {
```

```
private:
```

```
    double length;
```

```
    double breadth;
```

```
public:
```

```
    // parameterized constructor
```

```
    Rectangle(double l, double b) {
```

```
        length = l;
```

```
        breadth = b;
```

```
    }
```

```
// copy constructor with a Rectangle object as parameter copies data of the obj parameter
```

```
    Rectangle(Rectangle &obj) {
```

```
        length = obj.length;
```

```
        breadth = obj.breadth;
```

```
    }
```

```
    double calculateArea() {
```

```
        return length * breadth;
```

```
    }
```

```
};
```

```
int main() {
```

```
// create objects to call constructors
Rectangle obj1(10,6);
Rectangle obj2 = obj1; //copy the content using object

//print areas of rectangles
cout << "Area of Rectangle 1: " << obj1.calculateArea();
cout << "Area of Rectangle 2: " << obj2.calculateArea();

return 0;
}
```

### Practical Related Questions:

- 1.What is constructor? Explain its uses.
- 2.Explain Copy constructor with proper example.
- 3.What is constructor overloading in CPP explain with example.

### Programs:-

1. Design a class student having data members as name and percentage. Write a constructor to initialize these data members. Accept and display for 1 object.
2. Define class salary which will contain member variables Basics, TA, DA, HRA : Write a program using constructor with the default values for DA and HRA, and calculate the salary of the employee.
3. Write a program to declare the class time having data members hrs,min and sec . Write a constructor to accept data and display for 2 objects.

### Conclusion :-

Hence, we learnt to implement the constructors in CPP.

### Practical 5

**1. Design a class student having data members as name and percentage. Write a constructor to initialize these data members. Accept and display for 1 object.**

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class student
{
char name[20];
int per;
public:
student(){
    strcpy (name," ");
    per = 0;
}
void accept()
{
    cout<<"\n Enter name and percentage : ";
    cin>>name>>per;
}
void display(){
    cout<<"\n Name : "<<name;
    cout<<"\n Percentage : "<<per;
}
};
void main()
{
    student s1;
    s1.accept();
    s1.display();
    getch();
}
```

**2. Define class salary which will contain member variables Basics, TA, DA, HRA : Write a program using constructor with the default values for DA and HRA, and calculate the salary of the employee.**

```
#include<iostream.h>
#include<conio.h>
class Time
{

int hrs, min, sec;
```

```
public:
    Time(int a, int b,int c);
    void display();
};
Time::Time(int a , int b, int c)
{
    hrs = a;
    min = b;
    sec = c;
}
void Time::display()
{
    cout<< "\nHours = "<<hrs;
    cout<<"\tMinutes = "<<min;
    cout<<"\tSeconds = "<<sec;
}
void main()
{
    Time T1(13,32,54);
    Time T2(50,10,00);
    clrscr();
    T1.display();
    T2.display();
    getch();
}
```

**3. Write a program to declare the class time having data members hrs,min and sec . Write a constructor to accept data and display for 2 objects.**

```
#include<iostream.h>
#include<conio.h>
class SALARY
{
    float BASIC, TA, DA, HRA;
public:
    SALARY(float, float, float, float);
    void display();
};
SALARY::SALARY(float ta, float basic, float da = 200, float hra = 400)
{
    BASIC = basic;
    DA = (BASIC * da) /100;
    HRA = (BASIC * hra) /100;
    TA = (BASIC * ta) /100;
}
```



PCET-NMVPM's

# Nutan College of Engineering and Research, Talegaon, Pune

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING



```
void SALARY::display()
{
    cout<< "\n Salary of employee : "<<(DA + HRA + TA + BASIC);
}
void main()
{
    SALARY S(110, 6000);
    S.display();
    getch();
}
```