Constructors In C++

PRESENTED BY-

NEELAM SINGH

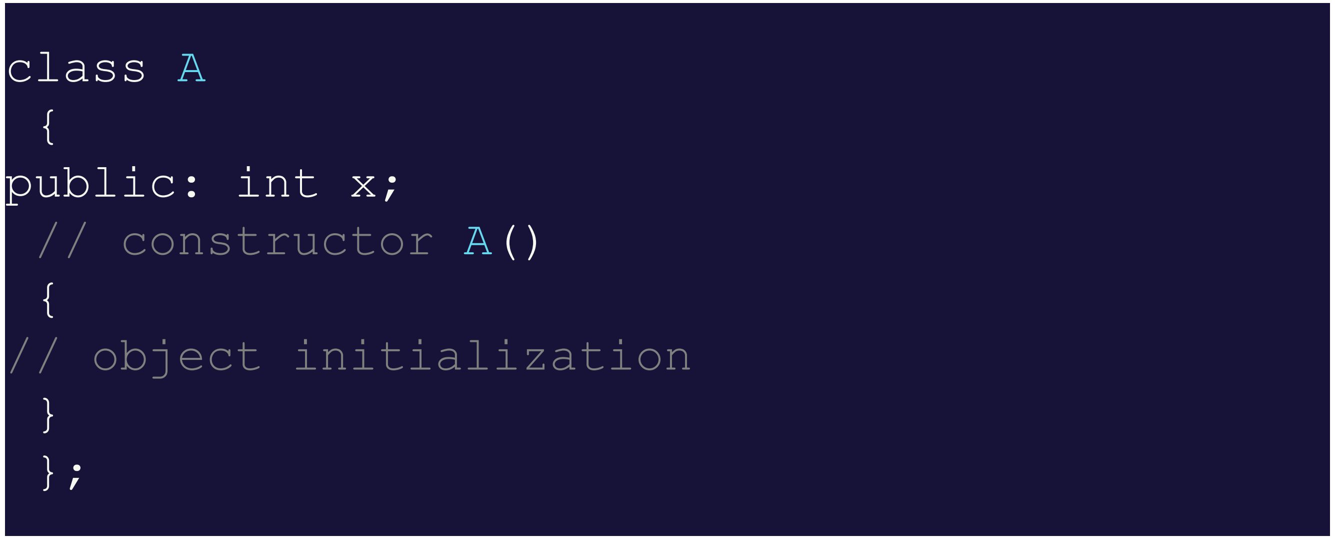
DEPARTMENT OF COMPUTER SCIENCE

INTRODUCTION

- Constructors are special class functions which performs initialization of every object. The Compiler calls the Constructor whenever an object is created. Constructors initialize values to object members after storage is allocated to the object.
- Whereas, Destructor on the other hand is used to destroy the class object.
- Before moving forward with Constructors and Destructors in C++ language, check these topics out to understand the concept better:

1. Function in C++
2. Class and Objects in C++
3. Data Members

- Let's start with Constructors first, following is the syntax of defining a constructor function in a class:



```
class A
 {
public: int x;
 // constructor A()
 {
// object initialization
 }
 };
```

While defining a contructor you must remeber that the **name of constructor** will be same as the **name of the class**, and contructors will never have a return type.

Constructors can be defined either inside the class definition or outside class definition using class name and scope resolution :: operator.

```
class A
{
   public:
 int i;
  A(); // constructor declared
};

// constructor definition
A::A()
{
   i = 1;
}
```

Types of Constructors in C++

Constructors are of three types:

1. Default Constructor
2. Parametrized Constructor
3. Copy COnstructor

Default Constructors

- Default constructor is the constructor which doesn't take any argument. It has no parameter.

SYNTAX-

```
class_name(parameter1, parameter2, ...)
{
   // constructor Definition
}
```

EXAMPLE

```
class Cube
{
   public:
 int side;
   Cube()
  {
     side = 10;
   }
};

int main()
{
   Cube c;
   cout << c.side;
}
```

OUTPUT – 10

Parameterized Constructors

- These are the constructors with parameter. Using this Constructor you can provide different values to data members of different objects, by passing the appropriate values as argument.

EXAMPLE

```
class Cube
{
  public:
  int side;
 Cube(int x)
 {
side=x;
 }
};
int main()
{
  Cube c1(10);
 Cube c2(20);
Cube c3(30);
cout << c1.side;
cout << c2.side;
cout << c3.side;
}
```

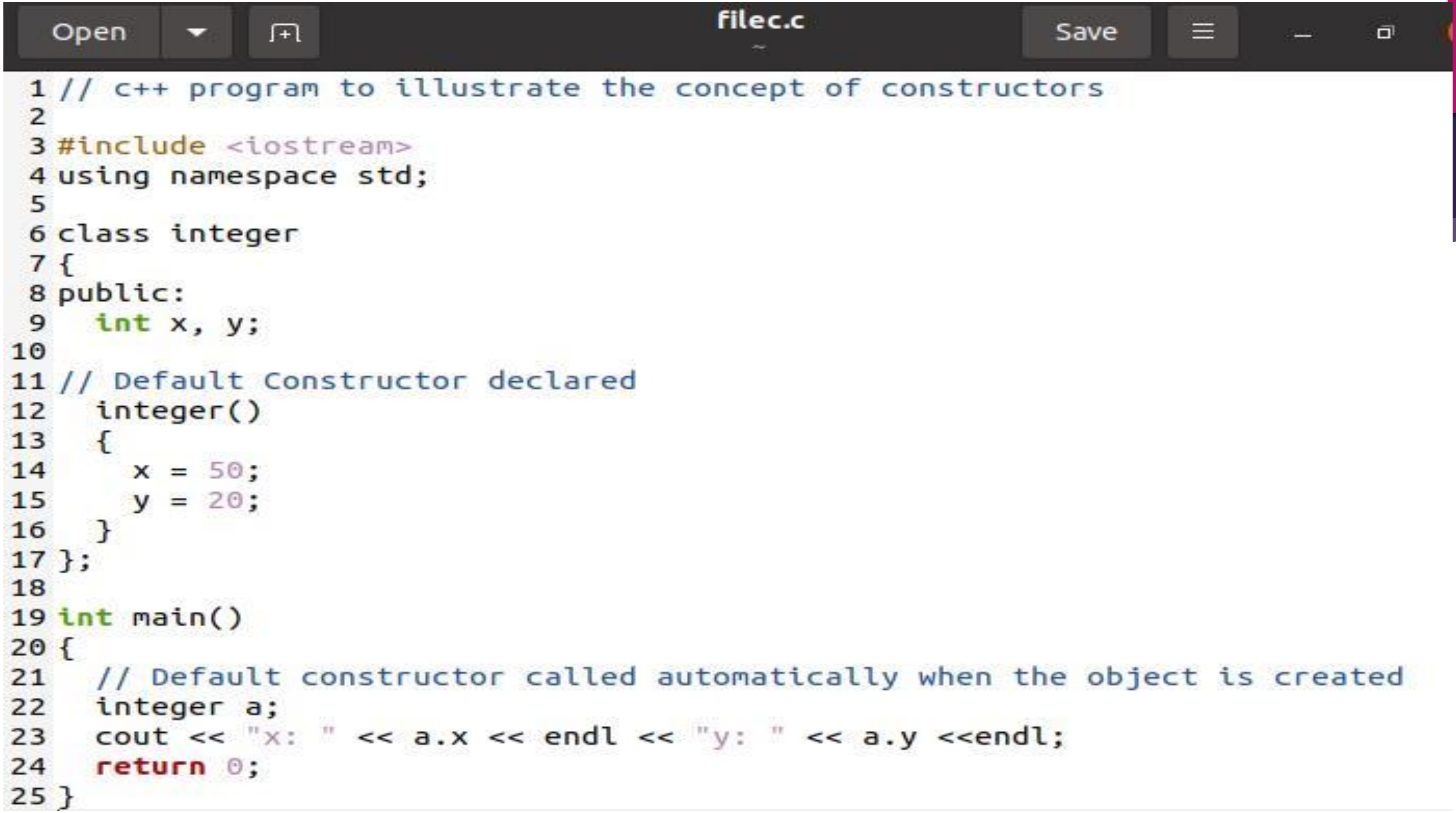
```
OUTPUT
              10
              20
              30
```

Copy Constructors

- 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. We will study copy constructorS in detail later.

Constructor Overloading in C++

- 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.
- You can have any number of Constructors in a class that differ in parameter list.



```cpp
// c++ program to illustrate the concept of constructors

#include <iostream>
using namespace std;

class integer
{
public:
  int x, y;

// Default Constructor declared
  integer()
  {
    x = 50;
    y = 20;
  }
};

int main()
{
  // Default constructor called automatically when the object is created
  integer a;
  cout << "x: " << a.x << endl << "y: " << a.y <<endl;
  return 0;
}
```

Conclusion

- The constructor may be defined as the special feature of the programming languages which is used to make the program effective and efficient. It can also be considered as a special type of method that has the same name as that of the class and can be invoked whenever the object of that class is created. Based on the requirement of the constructor once can choose between the default and the parameterized constructor. It has to be understood that it can only be used in the case when there is something that has to be called immediately right after the instance of the class has been created.

THANK YOU