الجامعة المستنصرية / كلية العلوم
قسم علوم الحاسوب

OOP

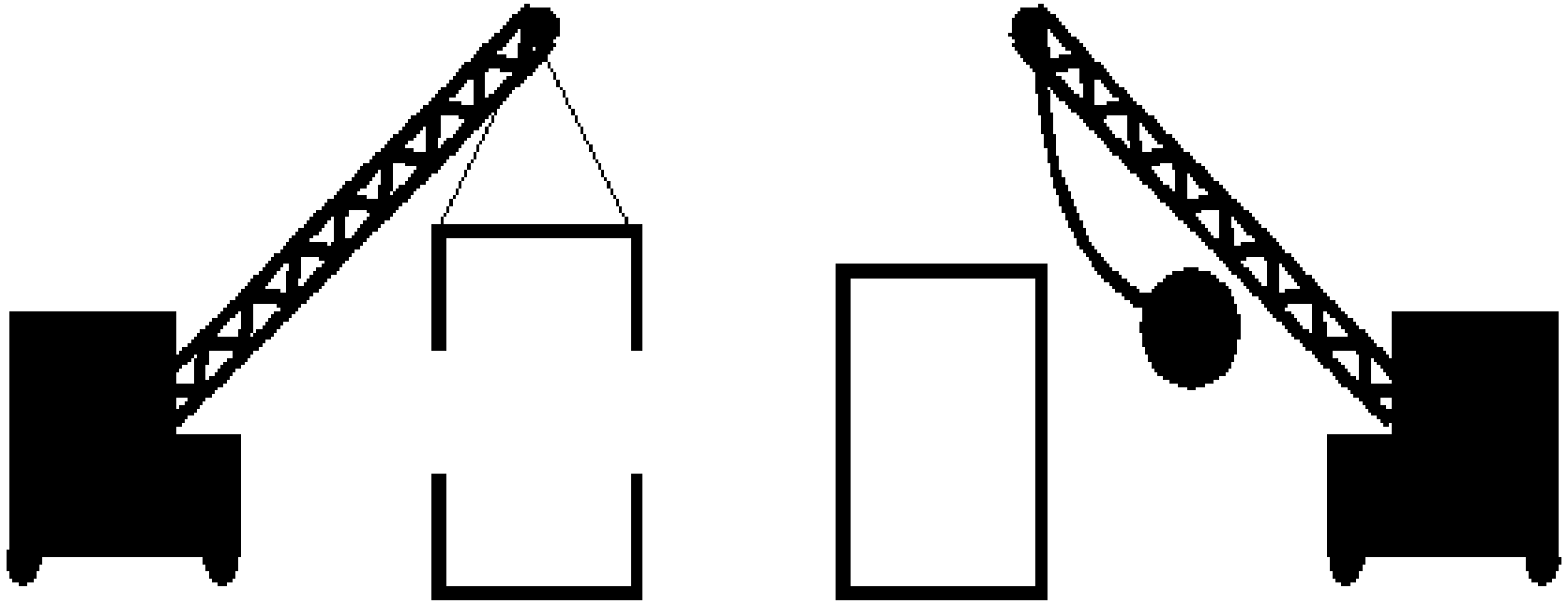
OBJECT-ORIENTED PROGRAMMING

5

CONSTRUCTOR AND DESTRUCTOR

A constructor is a **special method** that is used to initialize objects to initialize fields(data members)

it is called when an object of a class is created.



Constructor

Destructor

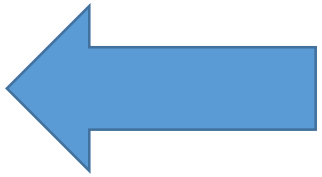
```
MyClass *MyObjPtr = new MyClass();
```

```
delete MyObjPtr;
```

Important Notes About Constructor :

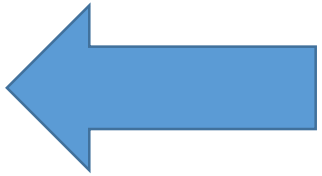
- A constructor has the same name as the class.
- Has no data type - A constructor can never return anything, which is why you don't have to define a return type for it.
- If no constructor defined then the CLR(Common Language Runtime) will provide an implicit constructor which is known as a *Default Constructor*.
- Constructors can be overloaded. - class can have any number of constructors and they vary with the number of arguments that are passed

```
public double findarea()
{
   ..............
}
```



method

```
public        findarea()
{
   ..............
}
```



constructor

```
class student
{
      private string name;
      private double av;

      public student()
      {
        Console.WriteLine(“STUDENT CONSTRUCTED");
      }

      public void DisplayInfo()
      {
            Console.WriteLine(“STUDENT’S NAME IS :  " + name + " AND AVERAGE IS : " + av);
      }
}
```

attributes

constructor

method

```
static void Main(string[] args)
      {
        student stu = new student();
        …
        …
        Console.ReadLine();
      }
```

اي ايعاز في (constructor) سيتنفذ مباشرة

The Output =

Constructor without Parameter

```
class student
{
      private string name;
      private double av;

      public student()
      {
        Console.WriteLine(“STUDENT CONSTRUCTED");
      }

      public void DisplayInfo()
      {
            Console.WriteLine(“STUDENT’S NAME IS :  " + name + " AND AVERAGE IS : " + av);
      }
}
```

attributes

constructor

method

```
using System;
program P5Constroctor0
{
    class student
     {
      private string name;
      private double av;

      public student()
      {
        Console.WriteLine(“STUDENT CONSTRUCTED");
      }

      public void DisplayInfo()
      {
            Console.WriteLine(“STUDENT’S NAME IS :  " + name + " AND AVERAGE IS : " + av);
      }
    }

static void Main(string[] args)
        {
          student stu = new student();     // ++++++ create object ++++++++++
          stu.DisplayInfo();
          Console.ReadLine();
        }
```

The Output =

Constructor with Parameter

```
class student
{
        private string name;
        private double av;

        public student(string n, double a)
        {
                name = n;
                av = a;
        }

        public void DisplayInfo()
        {
                Console.WriteLine("STUDENT'S NAME IS :  " + name + " AND AVERAGE IS : " + av);
        }
}
```

attributes

constructor

method

```
using System;
program P5Constroctor0
{
    class student
     {
      private string name;
      private double av;

       public student(string n, double a)
     {
            name = n;
            av = a;
     }
      public void DisplayInfo()
     {
           Console.WriteLine("STUDENT'S NAME IS :  " + name + " AND AVERAGE IS : " + av);
      }
    }

static void Main(string[] args)
       {
         student stu = new student("Ali" , 90);          // ++++++ create object ++++++++++
         stu.DisplayInfo();
         Console.ReadLine();
       }
```

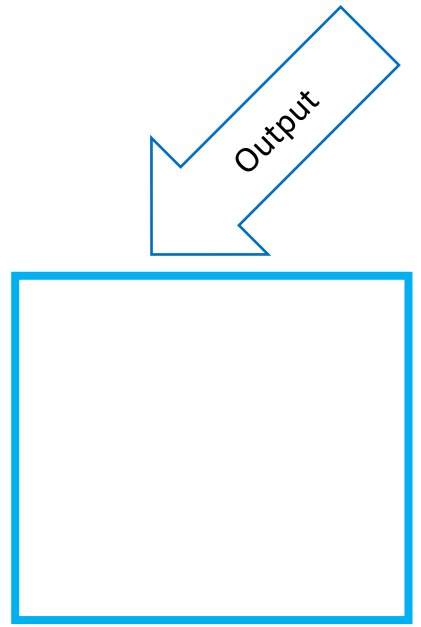
The Output =

COMPLETE PROGRAM student's average

```
using System;
namespace program6
{
  class student
  {
     private string name;
     private int d1,d2,d3;
     private double av;
     public student(string a, int b,c,d)    // with parameters
     {
        name = a;
        d1=b; d2=c; d3=d;
     }
    public double findav()
    {
      av=(d1+d2+d3)/3;
      return av;
     }
  }
```

constructor called once the instance of class created

```
  static void Main(string[] args)
  {
    student st = new student("Ali“,20,40,100);

    double ava= st.findav();
    console.writeline(ava);
    Console.ReadLine();
  }
 }
}
```



Multiple Constructor

```
class student
{
   private string name;
   private int d1,d2,d3;
   private double av;
   public student(string a, int b,c,d)
   {
      name = a;
      d1=b; d2=c; d3=d;
   }

   public student()
   {
      name = a;
      d1=100; d2=90; d3=99;
   }

   public double findav()
   {
     av=(d1+d2+d3) / 3;
     return av;
    }
}
```

```
static void Main(string[] args)
{
  student st = new student("Ali“,20,40,100);

  double ava= st.findav();
  console.writeline(ava);
  Console.ReadLine();
}
```

OUTPUT

```
static void Main(string[] args)
{
  student st = new student();

  double ava= st.findav();
  console.writeline(ava);
  Console.ReadLine();
}
```

OUTPUT

Destructors:

Destructor is a method inside the class used to destroy instance of that class when it is no longer needed. garbage cleanup is automatic system, (framework will free the objects that are no longer in use)

BUT there may be times where we need to do some manual cleanup. In this case we can use Destructor, which is used to destroy the objects that we no longer want to use (free or cleanup resources used by the object)

A destructor method called once an object is disposed.

```
class Student
{
    public Student()
    {
        //Constructor
        // Your code
    }

    ~Student()
    {
        //Destructor
        // Your code
    }
}
```

Once the class object is instantiated, *Constructor* will be called

and when object is collected by the garbage collector, *Destructor* method will be called.

The purpose of the destructor method is to remove unused objects and resources.

- Destructors are not called directly in the source code but during garbage collection.
- A destructor is invoked at an undetermined moment.
- More precisely a programmer can't control its execution; rather it is called by the Finalize () method.
- Like a constructor, the destructor has the same name as the class except a destructor is prefixed with a tilde (~).

- Destructors are parameterless.
- A Destructor can't be overloaded. (only one function)
- Destructors are not inherited.

```
class student
{
    private string name;
    private int d1,d2,d3;
    private double av;

    public student()
    {
        Console.WriteLine("+ Constructor was called +");
    }

    public double findav()
    {
        av=(d1+d2+d3)/3;
        return av;
    }

    ~ student()
    {
        Console.WriteLine("- Destructor was called -");
    }
}
```

```
static void Main(string[] args)
{
    student st = new student();

    Console.ReadLine();
}
```

OUTPUT

```
using System;
program P5Constructor1
{
class traingle
    {
        private int tbase;
        private int thight;
        private double ar;

        public traingle()                   // constructor without parameters
        {
            Console.WriteLine(" THE OBJECT HAS BEEN CREATED ");

            thight = 10;                                    // init variables
            tbase  = 20;

            Console.Write("input hight : ");                // input variables
            thight = Convert.ToInt32(Console.ReadLine());
            Console.Write("input base : ");
            tbase = Convert.ToInt32(Console.ReadLine());
        }

        public void printinfo() // print information
        {
            Console.WriteLine("BASE = " + tbase.ToString() );
            Console.WriteLine("HIGHT= " + thight.ToString());
        }

        public double findarea()            // find area
        {
            ar = 0.5 * tbase * thight;
            return ar;                  //  we should return the area
        }
    }
```

```
    static void Main(string[] args)
    {
        traingle tshape = new traingle();

        //tshape.printinfo();

        double tar;
        tar = tshape.findarea();

        Console.WriteLine("-----------------");
        Console.WriteLine("AREA="+"{0:N2}",tar);
        Console.WriteLine("-----------------");

        Console.ReadLine();
    }

}
```

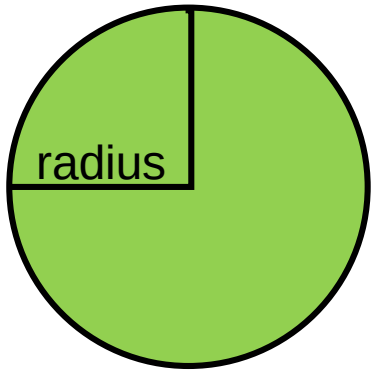
```
using System;
program P5Constructor1
{
class triangle
      {
         private int tbase;
         private int thight;
         private double ar;

         public triangle(int a, int b) // constructor with parameters
         {
            thight = a;          // initialize variables when create object
            tbase  = b;
         }

         public void printinfo() // print information
         {
            Console.WriteLine("BASE = " + tbase.ToString() );
            Console.WriteLine("HIGHT= " + thight.ToString());
         }

         public double findarea()          // find area
         {
            ar = 0.5 * tbase * thight;
            return ar;                 //  we should return the area
         }
      }
```

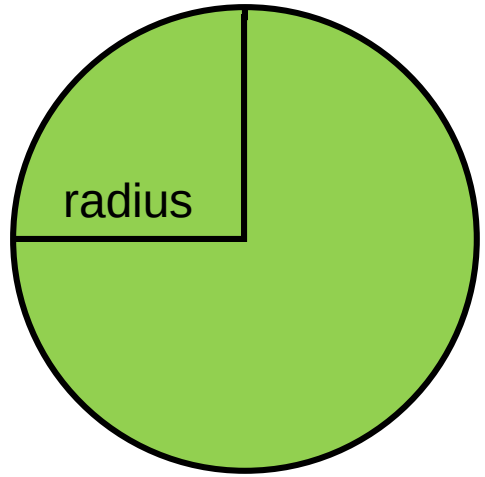
```
    static void Main(string[] args)
    {
        traingle tshape = new traingle( 6, 12 );

        //tshape.printinfo();

        double tar;
        tar = tshape.findarea();

        Console.WriteLine("-----------------");
        Console.WriteLine("AREA="+"{0:N2}",tar);
        Console.WriteLine("-----------------");

        Console.ReadLine();
    }

}
```

H.W.

Create a class to calculate area of circle with all necessary attributes and methods and constructors

Write the complete program to create the instance object for this class and call it twice if the radius = 10 and 20 cm





H.W.

Create a class to calculate factorial of a number N with all necessary attributes and methods and constructors

Write the complete program to create the instance object for this class and call it for N=4 and use the result to call it as N for the second object

N = 4
F = 25

N = 25
F =
15511210043330
985984000000

طرق الوقايا من فايروس كورونا

احرص دائما على تطهير يديك بعد
ملامسة الأسطح في الأماكن العامة

غسل اليدين بالماء والصابون لمدة
لا تقل عن 20 ثانية بشكل متكرر

عند السعال والعطس قم بوضع منديل
والتخلص منه عند الإنتهاء في سلة المهملات

ارتدِ قناعا واقيا كإجراء وقائي
في المستشفيات والأماكن المغلقة

قم بتطهير وتنظيف الأسطح التي تلامسها بشكل مُتكرر