Hello… Everyone.

Welcome to week four. You are now about to learn how to Write a class and create an object

From that class. You already created an object from classes by Java Library.

Example Scanner. // Scanner input = new Scanner(System.in).

However, let us introduce an array in java. You know array from C language. However in Java array is

An Object, that means we must create an Object of array using "new"

Example:

```
    int [] a = new int [100];
double [] d = new double[100];
Book [] b = new Book[100];
Car [] c = new Car[100];
```

Notice int and double are primitive data type and Book, and Car are **class** that we must write

Java give you size of array, so you can print a.length, b.length it shows 100. Java give you an exception

If you go over the size. Above declarations let you index from **0 to a.lenght-1**

When using loop:

```
for (int i=0 ; I < a.lenght; i++){  }
```

Because array is an object. Its name is a reference like a, d, b and c. That means if you pass any object to

A method is will be call by reference. Also means you can change the content of object from method.

String was also an object that you should have read about it already and knowing that String in java is immutable, that cannot be change. Example String hi = "Hello". We cannot insert any character to this String. If you need to you must create object of StrinBuffer or StringBuilder. One place you should go always is try to GOOGLE String.java and look for methods. Also you may use any ClassName.java and

See its methods and constructors.

Class and Object:

Suppose I want to write a class for Die. Dice has six sides. When we throw it, we look at face. It may have color too. So let write a class for Die:

```
/**
 *
```

```
 * @author dehkhoda
 */
public class Die {
  private int face;// this is called instance data or property or
member of Die
  private String color;

  public Die() // an Empty Constructor
  {
  }
  public Die(String c)// an overload constructor
  {
      color = c;
  }
  public void setColor(String c)
  {
    color = c;
  }
  public String getColor()
  {
      return color;
  }
  public void rollDie()
  {
      face =(int)(Math.random()* (6-1+1))+1;
  }
  public int  getFace()
  {
      return face;
  }
  public void  setFace(int n)
  {
      face = n;
  }

  // always write the following method for any
  // class you write
  public String toString()
  {
      return "Color: "+color+"\n"
             +"Face: "+ face+"\n";
  }
}
```

The Die class has two member name face and color.

Two constructor that is use to create an object of Die. Note make members always private.

Sets of setters and getters also method **toString()** that returns state of object.

This is how to create a Die object.

Die d1 = new Die()

Or

Die d2 = new Die(“Red”);

There is no limit how many object of Die we create.

System.out.println(d1.toString()); shows Color: null

Face: 0

System.out.println(d2.toString()); shows Color: Red

Face: 0

You can use setter: d1.setFace(5) , d1.setColor(“Blue”);

Also d1.rollDie();

Now we may want to have array of Die.

```
Die [] dd= new Die();
for (int i =0 ; i < dd.lenght; i++)
{
  dd [i] = new Die();
  dd[i].rollDie(); dd[i].setColor(“Red”+i);
}
```

Here is how to print using class MyIO:

```
String out=””;
for (Die x : dd)
 out+=x.toString();
MyIO.display(out);
```

You should type above program to understand better.

When we write new class to make sure all setters and getters are working, we write

TestDie (sometimes refer to driver class)

Example:

```
/**
 *
 * @author dehkhoda
 */
public class TestDice {
 public static void main(String [] args)
 {
     Sample();
    System.exit(0);
  }
 public static void Sample()
 {
   Die   []  ard  = new Die[5];
    String out = "List of throw Dice\n";
  for(int i = 0 ; i < ard.length; i++)
  {
     ard[i]= new Die();
     ard[i].rollDie();
     ard[i].setColor("Red"+(i+1));
     out+=ard[i].toString();
  }
        MyIO.display(out,Color.GREEN,Color.BLACK,32);
     }
    }
```