Fundamentals of Java : 1

Authored by : Sangeeta Joshi ***Presented by: Sangeeta Joshi***

Agenda

- Basic class structure
- Class – Object Relationship
- Java Coding Conventions
- Accessors & Mutators
- Default Constructor
- Constructor Overloading
- “this”
- Types of variables

Standard Class Structure

```
class Date
{
      int dd,mm,yy;

    p void seDate(int d,int m,int y)
    {
          dd=d;
         mm=m;
          yy=y;
     }
    p void displayDate()
    {
     System.out,println(dd+mm+yy);
     }


}
```

```
Class DateDemo
 {
 public static void main(String []args)
      {
           Date d1 =new Date();

           d1.setDate(1,1,2001);

           Date d2=new Date();

           d1.setDate(2,2,2002);

            d1.displayDate();

             d2.displayDate();
       }
}
```

Class is a blueprint for Object

In software, a class is a description of an object:

- A class describes the data that each object includes.
- A class describes the behaviors that all objects exhibits.
- A class represents the structure of the object
- An object is called as an instance of class

Accessors & Mutators

- Data is encapsulated inside an object.
- Methods are required to ***set , access or to modify*** this data.
- Mutators or Setters : The methods to set the data into an object

  Naming convention:

  public void setXXX(-----){}

- Accessors or Getters :The methods to access the data from an object

  Naming convention :

  public datatype getXXX(){}

Example Accessors & Mutators

```
 class Date
{
       int dd,mm,yy;
      public void setDate(int d,int m,int y)  //setter or mutator
        {
        dd=d;
        mm=m;
        yy=y;
        }
      public int getDd()                        //getter or accessor
        {
         return dd;
        }

 }
```

Accessing Object Members

Accessing Object Members

The dot notation is: <object>.<member>
This is used to access object members, including attributes and methods.

Examples of dot notation are:

```
d.display();
d.age = 42;
```

Constructor

Constructor is a special method:

- Its name is same as class name
- Constructor does not have any return type (not even void)
- It gets invoked implicitly whenever a new object is created
- Constructors can be overloaded

The Default Constructor

The Default Constructor

- There is always at least one constructor for every class
- If the programmer does not supply any constructor explicitly, the default constructor will be created and executed implicitly
- The default constructor takes no parameters
- The default constructor body is empty.

Constructor with Parameter

You can pass parameters to a constructor.

Example:

```
public class MainClass
 {
       private int age;

     public MainClass(int age)
       {
               age = 42;
         }
}
```

This …..keyword

- “this” is a keyword in java
- it points to the current invoking object
- every class member gets a hidden reference – *"this"*
- For *d1.display()* or *d1.dd* :

  here current invoking object is “d1” so ‘this’ points to d1

Demo : ‘this’

```
Class DemoThis
{
    String name;
     DemoThis()
     {
        System.out.println(“default”);
      }
     DemoThis( String name)
     {
         this();                              //......constructor chaining
         this.name=name;
      }

}
```

Variables

A **variable** is a name given to memory location. That memory is associated to a data type and can be assigned a value.

```
int n;

float f1;

char ch;

double d;
```

Variables conti…

Assigning a value to a variable

Initialization of a variable with a primary value

```
1. int      n1;
2. n1 =21 ;                    // assignment
3. int   i2 = 18;              // initialization
4. char  ch = 'S';             // initialization
5. double d = 21.8;            // initialization
6. d = n1;                     // assignment
7. float f1 = 16.13F;
```

Types of variables in java

Instance Variables	: Copy exists per instance
Static Variables	: Class level variable i.e. copy exists per class
Local Variables	: Variables declared within methods or blocks. They are local to the block where they are declared

Any Questions?