Object Oriented Programming using Java

Prepared By:
Suyel, PhD
Assistant Professor
Dept. of CSE, NIT Patna

Outline

1. this Keyword

2. Usages of this Keyword

this Keyword

- In Java, **this** is a reference variable that refers to the current object inside a method or a constructor..
- Java does not allow to declare two or more variables having the same name inside a scope (class scope or method scope). In such a situation, we use **this** keyword.
- If the name of the parameter and instance variable is different, the compiler automatically appends **this** keyword.
- The main purpose of using **this** keyword in Java is to remove the confusion between class attributes and parameters that have same names.

this Keyword (Cont…)

- Usages of this keyword
  - It can be used to refer the current class instance variable.
  - It can be used to invoke the current class method (implicitly).
  - this() can be used to invoke the current class constructor.
  - this keyword can be passed as an argument in the method call.
  - this can be passed as an argument in the constructor call.
  - It can be used to return the current class instance from the method.

Usages of this Keyword

- To refer the current class instance variable

```
class This1
{
   int ID;
   String name;

   This1(int ID, String name)
   {
      this.ID = ID;
      this.name = name;
   }

   void disp()
   {
      System.out.println("ID: " + ID + ", Name: " + name);
   }

   public static void main(String args[])
   {
      This1 obj1 = new This1(1010,"Gopi");
      This1 obj2 = new This1(2020,"Yuvraj");
      obj1.disp();
      obj2.disp();
   }
}
```

Usages of this Keyword (Cont…)

- To refer the current class instance variable (Cont…)
  - Output

```
ID: 1010, Name: Gopi
ID: 2020, Name: Yuvraj
```

Usages of this Keyword (Cont…)

- To invoke the current class method (implicitly)

```
class This2
{
   int ID;
   String name;

   void get(int x, String str)
   {
      ID = x;
      name = str;
      System.out.println("ID: " + ID + ", Name: " + name);
   }

   void disp()
   {
      System.out.println("Hello, How are you?");
      this.get(1010, "Ayushi");
      System.out.println("This is the line after get method?");
   }

   public static void main(String args[])
   {
      This2 obj = new This2();
      obj.disp();
   }
}
```

Usages of this Keyword (Cont…)

- To invoke the current class method (implicitly) (Cont…)
  - Output

```
Hello, How are you?
ID: 1010, Name: Ayushi
This is the line after get method?
```

Usages of this Keyword (Cont…)

- To invoke the current class constructor by this()
  - this() must be the first statement in constructor.

```
class This4
{
   int ID;
   String name;

   This4()
   {
      this(1010, "Ashish");
      System.out.println("We are in Default Constructor");
   }

   This4(int ID, String name)
   {
      this.ID = ID;
      this.name = name;
      System.out.println("We are in Parameterized Constructor");
      System.out.println("ID: " + ID + ", Name: " + name);
   }

   public static void main(String args[])
   {
      This4 obj = new This4();
   }
}
```

Usages of this Keyword (Cont…)

- To invoke the current class constructor by this() (Cont…)
  - Output

```
We are in Parameterized Constructor
ID: 1010, Name: Ashish
We are in Default Constructor
```

Usages of this Keyword (Cont…)

- To invoke the current class constructor by this() (Cont…)

```
class This5
{
   int ID;
   String name;
   String dept;
   float mark;

   This5(int ID, String name, String dept)
   {
      this.ID = ID;
      this.name = name;
      this.dept = dept;
   }

   This5(int ID, String name, String dept, float mark)
   {
      this(ID, name, dept);
      this.mark = mark;
   }

   void disp()
   {
      System.out.println("ID: " + ID + ", Name: " + name + ", Department: " + dept + ", Mark: " + mark);
   }

   public static void main(String args[])
   {
      This5 obj = new This5(1010, "Ankit", "CSE", 80);
      This5 obj1 = new This5(2020, "Vikas", "ECE");
      obj.disp();
      obj1.disp();
   }
}
```

Usages of this Keyword (Cont…)

- To invoke the current class constructor by this() (Cont…)
  - Output

```
ID: 1010, Name: Ankit, Department: CSE, Mark: 80.0
ID: 2020, Name: Vikas, Department: ECE, Mark: 0.0
```

Usages of this Keyword (Cont…)

- To pass as an argument in the method
  - It is mainly used in event handling.
  - When we have to provide reference of a class to another one, then we use this keyword.
  - It is used to reuse one object in many methods.

Usages of this Keyword (Cont…)

- To pass as an argument in the method (Cont…)

```
class This6
{
   void disp(This6 obj)
   {
      System.out.println("Method is invoked");
   }

   void get()
   {
      disp(this);
   }

   public static void main(String args[])
   {
      This6 obj = new This6();
      obj.get();
   }
}
```

Usages of this Keyword (Cont…)

- To pass as an argument in the method (Cont…)
  - Output

    Method is invoked

Usages of this Keyword (Cont…)

- To pass as an argument in the method (Cont…)

Test.java

```
class Test
{
   int x;
   int y;
   int z;
   Test(int x, int y)
   {
      this.x = x;
      this.y = y;

      System.out.println("Before passing this to sum() method:");
      System.out.println("x = " + this.x + ", y = " + this.y);

      sum(this);

      System.out.println("After passing this to sum() method:");
      System.out.println("x = " + this.x + ", y = " + this.y);
   }

   void sum(Test o)
   {
      x = x + 2;
      y = y + 3;
      z = x + y;
      System.out.println("Sum: "+ z);
   }
}
```

This7.java

```
class This7
{
   public static void main(String args[])
   {
      Test obj = new Test(5, 10);
   }
}
```

Usages of this Keyword (Cont…)

- To pass as an argument in the method (Cont…)
  - Output

```
Before passing this to sum() method:
x = 5, y = 10
Sum: 20
After passing this to sum() method:
x = 7, y = 13
```

Usages of this Keyword (Cont…)

- To pass as an argument in the constructor call
  - It is useful, if we have to use one object in multiple classes.

```
class Test1
{
   This8 x;

   Test1(This8 ob)
   {
      this.x = ob;
   }

   void disp()
   {
      System.out.println(x.y); //Printing using data member of This8 class
   }
}

class This8
{
   int y = 5;

   This8()
   {
      Test1 obj1 = new Test1(this);
      obj1.disp();
   }

   public static void main(String args[])
   {
      This8 obj = new This8();
   }
}
```

Usages of this Keyword (Cont…)

- To pass as an argument in the constructor call (Cont…)
  - Output

```
5
```

Usages of this Keyword (Cont…)

- To return the current class instance from the method
  - To **return** this keyword as an statement from the method, return type of the method must be the class type (non-primitive).
  - For example:

```
return_type method_name()
{
    return this;
}
```

Usages of this Keyword (Cont…)

- To return the current class instance from the method (Cont..)

```
class Test2
{
   Test2 get()
   {
      return this;
   }

   void disp()
   {
      System.out.println("How are you?");
   }
}

class This9
{
   public static void main(String args[])
   {
      Test2 obj1 = new Test2(); //Below lines are same as "new Test2().get().disp();"
      Test2 obj2 = obj1.get();
      obj2.disp();
   }
}
```

Usages of this Keyword (Cont…)

- To return the current class instance from the method (Cont..)
  - Output

    How are you?

Usages of this Keyword (Cont…)

- To return the current class instance from the method (Cont..)

```
class This10
{
   int x;
   int y;

   This10()
   {
      x = 5;
      y = 10;
   }

   This10 get()
   {
      return this;
   }

   void disp()
   {
      System.out.println("x: " + x + ", y: " + y);
   }

   public static void main(String args[])
   {
      This10 obj = new This10();
      obj.get().disp();
   }
}
```

Usages of this Keyword (Cont…)

- To return the current class instance from the method (Cont..)
  - Output

    x: 5, y: 10

thank you . . .

Slides are prepared from various sources, such as Book, Internet Links and many more.