Access modifiers: GETTERS & SETTERS

Access modifiers

Access modifiers specify where a property is accessible. There are 4 types of access modifiers.

1) Private
2) Default
3) Protected
4) Public

Access modifier	within class	within package	Outside package by subclass only	Outside package
Public	✓	✓	✓	✓
protected	✓	✓	✓	N
Default	✓	✓	N	N
Private	✓	N	N	N

Getters & Setters

~~Getter → Returns a value (accessor)~~
~~Setter → Sets/ Updates a value (mutator)~~

~~For e.g.~~:
Private access modifier can be ~~used~~ accessed within the class only.
So let's try to access it outside the class.

```
class Employee {

    private int id;
    private String name;

}
```

```
public class AM {

    p s v m (String[] args) {

        Employee emp1 = new Employee();
        emp1.id = 3;
        emp1.name = "Shubham";

    }
}
```

o/p:

id has private access in employee

But, We can access private access modifier outside the class using Getters & Setters

→ Getter → Returns the value (accessor)

→ Setter → Sets/Updates the value (mutator)

For e.g.

```
class Employee {

private int id;
private String name;

public String getName() {
        return name;
    }
```

```
public void setName(String n){
    name = n;
}

public void setId(int i){
    id = i;
}

public int getId(){
    return id;
}
}

public class CWH {
    p s v m (String[] args){

Employee emp1 = new Employee();

emp1.setName("Shubham");
Sout.ln(emp1.getName());
emp1.setId(1);
Sout.ln(emp1.getId());

}
}

o/p
Shubham
1
```

Constructors in Java

→ Constructors are similar to methods but they are used to initialize an object.

→ Constructors do not have any return type (not even void)

→ Every time we create an object by using the new() keyword, a constructor is called.

→ If we do not create a constructor by ourself, then the default constructor (created by Java compiler) is called.

Rules for creating a Constructor:

1) The class name & constructor name must be the same

2) It must have no explicit return type

3) It cannot be abstract, static, final & synchronized.

Types of Constructors in Java

1) Default Constructor:

A constructor with zero parameters is known as default constructor.

→ Syntax

<class_name>(){
// code to be executed on the execution of the constructor
}

Eg.

```
class CWH{
    CWH(){
        System.out.println("This is a default constructor of CWH class");
    }
}

public class CWH_constructors{
    p s v m( ){
        CWH obj1 = new CWH();
    }
}
```

o/p
This is the default constructor of CWH class.

In the above code, CWH() is the constructor of class CWH. The CWH() constructor is invoked automatically with the creation of ~~obj~~ obj1.

2) Parameterized constructor:

A constructor with some specified number of parameters is called parameterized constructor.

```
<class-name>(<data-type> par1,<data-type>p2)
{
// Code
}
```

E.g.

```
class CWH{
    CWH(String s, int b){
        SoutIn("This is the " + b + "th video
                of" + " " + s);
    }
}

public class CWH_constructor{
    p s v m ( ){
        CWH obj1 = new CWH("CWH", 42);
```

o/p

This is the 42th video of CWH

In the above e.g., CWH() constructor accepts 2 parameters

Constructor Overloading in Java:

Just like methods, constructors can also be overloaded in Java. We can overload the Employee constructor like below.

E.g.

```
public Employee(String n){
    name = n;
}
```

Note:

1) Constructors can take parameters without being overloaded

2) There can be more than 2 overloaded constructors.

For e.g.

```
class Employee{
  // First constructor
  Employee(String s, int i){
  System.out.println("The name of 1st employee is: "+s);
  System.out.println("The id of 1st employee is: "+i);

  }
  // Constructor overloaded

  Employee(String s, int i, int salary){
```

```
sout.ln("The name of second employee is: "+s);
sout.ln("The id of  "  ——————  "  ——  : "+i);
sout.ln("  —  salary-"  ——————————————  "+salary);

}

public class CWH_const{

    psvm(  ){
        Employee ~~Student~~
        Employee ~~Shubham = new Employee()~~
    Employee shubham = new Employee("Shubham,
                                    1);
Employee harry = new Employee("Harry",2,70000);

    }

}
```

o/p
The name of first employee is: Shubham
The id of the first employee is: 1
The name of the second employee is: Harry
The salary of second employee is: 70,000
The id of " —————— is: 2

Practice set on constructors & Access Modifiers

Q1 Create a class cylinder & use getter & setter to set its radius & height, calculate its surface area & volume
Use constructor & repeat

```
package com.company

class cylinder {

    private float height;
    private float radius;

    public float getRadius() {
        return radius;
    }

    public void setRadius(float radius)
    {
        this.radius = radius
    }

Same for height

    public float surfaceArea() {

        return 6.28f * radius * height * (radius +
                                          height);
    }

    public float volume() {
        return 3.14f * radius * radius * height;
    }

    public cylinder(float radius, float height)
    {   this.radius = radius;
        this.height = height;
    }
}
```

```
public class Accessmod {

    psvm( ) {
    cylender obj = new cylender(4.5f, 7.6f);

    Sout("The height of cylender is:" +
                                obj.getHeight());
Sout("The radius of cylender is:" +
                                obj.getRadius());

Soutln("The surface area is:" + obj.surfaceArea());

Soutln("The volume is:" + obj.volume());

    }
```

Output

```
The height of cylender is: 7.6
The radius of cylender is: 4.5
The surface area is: 259.7898
The volume is: 483.246
```