JAVA – Access specifiers

02 Encapsulation

```
class Car
{
    int mileage;
    int max_speed;
    public:
        void display();
}
```

Data Encapsulation - Wrapping the data and functions in one single unit

03 Abstraction

Abstraction - hiding irrelevant details from the user.

Access specifiers are the main pillar of implementing abstraction.

Java Package:

- Is a group of similar types of classes, interfaces and sub-packages
- built-in package and user-defined package
- built-in packages such as java, lang, awt, javax, swing, net, io, util, sql etc.(14 packages)

Simple example of java package

```
package mypack;
public class Simple
{
                public static void main(String args[])
                {
                        System.out.println("Welcome to package");
                }
}
```

Access package from another package:

Three ways to access the package from outside the package.

- import package.*;
- import package.classname;
- fully qualified name.

Using packagename.*

```
//save by A.java
package pack;
public class A
{
        public void msg()
        {
                System.out.println("Hello");
        }
}
```

```
//save by B.java
package mypack;
import pack.*;
class B
{
        public static void main(String args[])
        {
                A obj = new A();
                obj.msg();
        }
 }
```

```
Output: Hello
```

Using packagename.classname

```
//save by A.java
package pack;
public class A
{
        public void msg()
        {
                System.out.println("Hello");
        }
}

//save by B.java
package mypack;
import pack.A;
class B
{
 public static void main(String args[])
        {
                 A obj = new A();
                obj.msg();
         }
 }
```

Output: Hello

Using fully qualified name

```
//save by A.java
package pack;
public class A
{
     public void msg()
     {
             System.out.println("Hello");
     }
}
```

```
//save by B.java
package mypack;
class B
{
  public static void main(String args[])
     {
             pack.A obj = new pack.A();  //using fully qualified name
              obj.msg();
      }
}
```

```
Output: Hello
```

Access Specifiers

All members of a class can be accessed:

Within the class - No restriction.

From outside the class - ?

Access Specifiers

- Defines the access control

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

- **Access specifiers** define how a member's variables and member's functions of a class can be accessed **from outside the class**.

Access Specifiers

Private

Access Denied - Outsiders

```
class A
{
private int data;
int a;
public int b;
private void msg(){
        System.out.println("Hello java");    }
}
 public class Simple
{  public static void main(String args[]){
       A obj=new A();
        System.out.println(obj.data);
                 //Compile Time Error
        obj.msg();//Compile Time Error
        obj.data=40; // compile time error
        obj.a=10;
        obj.b= 20;
    }
}
```

Private

Access Denied - Outsiders

Access Specifiers

✔ Any object or function outside the class cannot access the private members.

Access Specifiers

Private

Access Denied - Outsiders

- ✔ Any object or function outside the class cannot access the private members.
- ✔ **Can be accessed only by the functions inside the class.**

```
public class A
{
    private String name;
     public String getName()
    {
        return name;
    }
    public void setName(String name)
    {
        this.name = name;
    }
 }
```

```
 public class Main
{
   public static void main(Stringargs[])
   {
      A obj=new A();
      obj.setName("Face");
      System.out.println(obj.getName());
  }
}
```

Output:

```
Face
```

Private

Access Denied - Outsiders

Access Specifiers

- ✔ Any object or function outside the class cannot access the private members.
- ✔ Can be accessed only by the functions inside the class.
- ✔ **By default all the members of a class would be private.**

Access Specifiers

Proctected

- ✔ A protected member variable or function is very similar to a private member
- ✔ But they can be accessed by any subclass (derived class) of that class.

```
//save by A.java
package pack;
public class A
{
   protected void msg()
   {
        System.out.println("Hello");
   }
}
//save by B.java
package mypack;
import pack.*;
class B extends A
{
    public static void main(String args[])
    {
        B obj = new B();
        obj.msg();
    }
}
```

Output:
Hello

Access Specifiers

Public

```
class A
{
    public void msg();
    {
       System.out.println("Hello");
    }
}
class B
{
  public static void main(String args[])
   {
         A obj = new A();
         obj.msg();
   }
}
```

Output:

Hello

Access Specifiers

- ✔ All the class members declared under public will be available to everyone.
- ✔ Public members can be accessed by other classes too.
- ✔ Can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

Accessing Public Data Members:

```
class A
{
     public void msg();
     {
        System.out.println("Hello");
     }
}
class B
{
   public static void main(String args[])
    {
           A obj = new A();
           obj.msg();
    }
}
```

- Accessed using the direct member access (.) operator with the **object** of that class.

Access Specifiers

● ● ● ● ●

Default

Accessing Default Data Members:

```
//save by A.java
package pack;
class A
{
    void msg()
    {
        System.out.println("Hello");
    }
}
//save by B.java
package mypack;
import pack.*;
class B
{
    public static void main(String args[])
    {
        A obj = new A();//Compile Time Error
        obj.msg();//Compile Time Error
    }
}
```

- Access only the members of the same package.

1. Which methods can access to private attributes of a class?

a. Only Static methods of the same class
[✓ b. Only instances of the same class]
c. Only methods those defined in the same class
d. Only classes available in the same package.

2. What is the output of the following program?

```
class Area {
int width;
int length;
int area;
void area(int width, int length)
{
this.width = width;
this.length = length;
}
}
```

```
class Output {
public static void main(String  args[])
{
Area obj = new Area();
obj.area(5 , 6);
System.out.println(obj.length
+ " " + obj.width);
}
}
```

a. 0 0 b. 5 6 c. 6 5 d. 5 5

✓

3. Which of the following statements are incorrect?

a. Default constructor is called at the time of declaration of the object if a constructor has not been defined.

b. Constructor can be parameterized.

c. finalize() method is called when a object goes out of scope and is no longer needed. ✓

d. finalize() method must be declared protected.

4. What is the output of this program?

```
class ControlAccess
{
public int x;
private int y;
void cal(int a, int b)
{
x =a + 1;
y =b;
}
}
```

```
class Access_Control {
public static void main(String args[])
{
access obj = new access();
obj.cal(2, 3);
System.out.println(obj.x + " " + obj.y);
}
}
```

a. 3 3   b. 2 3   c. Runtime Error   [✓ d. Compilation Error]

5. The main method should be static for the reason

a. It can be accessed easily by the class loader.

b. It can be accessed by every method or variable without any hindrance.

[✓ c. It can be executed without creating any instance of the class.]

a. None of the above

6. Find the output of following program?

```
class Access{
public int x;
private int y;
void cal(int a, int b){
x =a + 1;
y =b;
}
void print() {
system.out.println(" " + y);
}
}
```

```
class Access_Control {
public static void main(String args[])
{
access obj = new access();
obj.cal(2, 3);
System.out.println(obj.x);
obj.print();
}
}
```

a. 2 3 b. 3 3 c. Runtime Error [✓ d. Compilation Error]

8. What is the output of this program?

```
class Output {
static void main(String args[])
{
int x , y = 1;
x = 10;
if (x != 10 && x / 0 == 0)
System.out.println(y);
else
System.out.println(++y);
}
}
```

a. 1   b. 2   c. Runtime Error   [✓ d. Compilation Error]

9. Predict the output

```
public class InitDemo
{
static int i = demo();
Static
{
System.out.print(i);
}
InitDemo(){
System.out.print("hello1");
}
public static void main
(String... args)
{
System.out.print("Hello2");
}
static int demo(){
System.out.print("InsideDemo");
return 10;
}
}
```

a. Compilation error.

b. Illegal Argument Exception is thrown at runtime.

c. InsideDemo 10 Hello2

[✓ d. Hello2 InsideDemo 10]

10. What will be the output for the below code?

```
public class Test{
static{
int a = 5;
}
public static void main(String[] args){
System.out.println(a);
}
}
```

a. [✓ Compile with error] b. 5 c. 0

b. d. Runtime Exception

11. Find the output of the following program?

```
class Area
{
int width;
int length;
int volume;
area()
{
width=5;
length=6;
}
void volume()
 {
volume = width*length*height;
}
}
class Cons_method {
public static void main(String args[])
{
Area obj = new Area();
obj.volume();
System.out.println(obj.volume);
}
}
```

a. 0 b. 1 c. 30 [✓ d. error]

12. What is the return type of Constructors?

a. int

b. float

c. void

d. [✓ None of the above]

14.Find the output of the following program?

```
class Box {
int width;
int height;
int length;
}
class Mainclass {
public static void main(String args[])
{
Box obj = new Box();
System.out.println(obj);
}
}
```

a. 0 b. 1 c. Runtime error d. Garbage value

✓

15. Which of the following is a valid declaration of an object of class Coffee?

a. [✓ Coffee barista = new Coffee();]

b. Coffee barista = new Coffee;

c. Coffee = new Coffee();

d. new Coffee barista;

Thank you