

CHAPTER-6
MORE ABOUT CLASSES AND LIBRARIES

Brief Summary of the Chapter:

In this chapter the way access of members of a class i.e. about access specifier will be discuss. Java include predefined classes in the form of packages which are also called Java class Library. Some of the used packages are: java.lang, java.util, java.io, java.swing, java.awt, java.applet etc.

Key points:

- The public member of object are accessed through .(dot) operator.
- The private members are accessible only inside their own class.
- The protected members are accessible inside their own class, sub class and packages.
- The default members are accessible inside their own class as well to classes in the same package.
- Related classes and interfaces are grouped together in the form of package.
- Packages and class are imported through import command.

SOLVED QUESTIONS

1. Which keyword can protect a class in a package from accessibility by the classes outside the package?

(a)private (b)protected (c) final (d) None of these

Ans: (d) None of these.

2. We would like to make a member of a class visible in all subclasses regardless of what package they are in. Which one of the following keywords would achieve this?

(a)private (b)protected (c) final (d) public (e) None of these

Ans: (b) protected.

3. Which of the following keywords are used to control access to a class member?

(a) default (b) abstract (c)protected (d) interface (e) public.

Ans: (c) and (e) public

4. The public members of objects are accessed through which operator.

(a) arrow (b) dot (c) this (d) none of these

Ans: (b) dot

5. The private members are accessible only inside their _____ class.

(a) own (b) sub (c) super (d) none of these

Ans: (a) own

6. Which command is used to import packages and their classes?

(a) include (b) import (c) public (d) inline

Ans: (b) import

7.Which statement is used to create a package in Java?

(a) Class (b) super (c) this (d) package

Ans:(d) package

8. In Java, all strings are objects?

(a) True (b) False (c) don't say

Ans: (a) True

9. What do you understand by Package in Java?

Ans: A group of classes is called package

10. Given a package named EDU. Student, how would you import a class named Test contained in this package? Write one line statement.

Ans: import EDU.Student.Test;

11. What will be the output of the following code

```
StringBuffer city = new StringBuffer("Madras");
```

```
StringBuffer string = new StringBuffer();
string.append(new String(city));
string.insert(0,"Central");
string.out.println(string);
```

Ans: CentralMadras.

12. Give the output of the following program:

```
class MainString
{ public static void main( String args[])
    { StringBuffer s = new StringBuffer("String");
      if(s.length() > 5) && (s.append("Buffer").equals("x"));
        System.out.println(s);
    }
}
```

Ans: StringBuffer.

13. What is the output of the following code fragment if "abc" is passed as argument to the func()?

```
Public static void func(string s1)
{
    String s = s1 + "xyz";
    System.out.println("s1=" + s1);
    System.out.println("s = " +s);
}
```

Ans: s1= abc

s =abcxyz

14. What are the access specifiers in Java? Explain.

Ans: The Access Specifiers control access to members of class from / within Java Program. Java supports various Access Specifiers to control the accessibility of class members.

☐ Private : A variable or method declared as private, may not be accessed outside of the class. Only class member can access them, since they are private to others.

☐ Protected: Protected members can be accessed by the class members and subclasses (derived classes) and current package, but they are not accessible from beyond package or outside.

☐ Public: Class members declared as public, are accessible to any other class i.e. everywhere , since they are public.

☐ Package (default): If no any specifier is mentioned, default or friendly access is assumed. Class member may be accessed by any other Class members available in the same package, but not accessible by the other classes outside the package, even subclasses.

15. What do you meant by private, public, protected, package(friendly) access specifiers?

Ans: **Private Access Specifier**

Members declared as private are accessible by the members of the same class, since they are private.

A private key word is used to specify.

//e.g to demonstrate private specifier.//

class abc

```

{ private int p;
private void method1()
{ p=10;
system.out.print("I am Private method");
}
}
class xyz
{.....
void method2()
{ abc x = new abc();
x.p =10;
x.method1() ;
}
}

```

Protected Access Specifier

Protected members are accessible by all the classes in the same package and sub-classes (same of different packages). A protected key word is used to specify.

Package mypackage;

```

class abc
{ protected int p;
protected void method1()
{ p=10;
system.out.print("Protected
method");
}
}
class xyz
{.....
void method2()
{ abc x = new abc();
x.p =10;
x.method1() ;
}
}

```

Lets another Package...

```

package yourpackage;
import mypackage.*;
class pqr extends abc
{ void method3()
{ abc x=new abc();
pqr y=new pqr();
x.p=10;
x.method1();
y.p=10;
y.method1();
}
}

```

Public Access Specifier

Public Members can be access at anywhere i.e. same or different package. A public key word is used to specify.

```

packagemypackage;
class abc
{ public int p;
public void method1()

```

```
{ p=10;
system.out.print("Public method");
}
}
```

```
package yourpackage;
import mypackage.* ;
class xyz
{.....
void method2()
{ abc x = new abc();
x.p =10;
x.method1() ;
}
}
```

Package (friendly) Access Specifier

If no specifier is explicitly specified, Java assumes default (friendly) access i.e. all the members are accessible in all other classes of the same package only, since they are trusted or friends. This is called Package level access. No any key word is used to specify default access.

```
package mypackage;
class abc
{ int p;
void method1()
{ p=10;
system.out.print("Package method");
}
}
class xyz
{.....
void method2()
{ abc x = new abc();
x.p =10;
x.method1() ;
}
}
```

16. What do you understand by Library in Java?

Ans: A library is readymade and reusable component/codes that can be used in a program to perform predefined task.

☐ Some commonly used Java libraries are Math Library, String Library, Utility Library and IO Library etc.

☐ You can use import statement at the top of the program to include the Java libraries.

```
import java.io.*;
```

☐ The java.lang is the default imported library in your program without writing import statement.

String Library & its commonly used methods

1. boolean equals(str) - Compare this (current) string to given string and returns true if both are true otherwise false. e.g. `boolean test=str1.equals(str2);`

2. int compareTo(str1,str2) - Compare two strings in alphabetical order.

boolean equalsIgnoreCase(str) - Compare this string to given string but ignores case difference.

3. int length() -Returns the length of this string.

e.g. `int x=str1.length();`

Math Library & its commonly used methods

☐ Java provides math library, which available under java.lang package.

□ In order to use functions/methods of math library, you need to invoke function using math keywords before the function.

e.g. `x=math.abs(-7.5);`

1. **pow(num1,num2)** - It computes num1 num2 , where num1 and num2 are numbers.

e.g. `syste.out.print(""+math.pow(2,3);`

2. **round(num1)** - It rounds off a given number to its nearest integer. It can take float/double as argument.

e.g.

`system.out.print(""+math.round(1.5)); 2`

`system.out.print(""+math.round(-1.5)); -1`

Using Dates & Times in JAVA

□ Java offers two classes in java.util package to manipulate date and time.

1. java.util.Date 2. java.util.Calendar

□ In order to use Date & calendar, you need to import java.util package. E.g. `import java.util.*;`

Date d=new Date(); -It returns system date in the given format.

Tue Jul 20 17:30:22 GMT+05:30 2010

UNSOLVED QUESTIONS

1. What are the different types of access specifier supported by java?
2. Which is the default package of java?
3. What is friendly access of class member?
4. How does a class enforce information hiding?
5. Define an abstract class and abstract methods.
6. What is an interface? What is the use of Interface.

CHAPTER 7

CONCEPT OF INHERITANCE IN JAVA

Brief Summary of the Chapter:

This chapter talks about Inheritance, the capability of one class to derive properties from another class. Here we can learn how the inheritance is implemented in Java.

KEY POINTS

- **Inheritance:** Inheritance is the capability of one class to inherit properties from an existing class. Inheritance supports reusability of code and is able to simulate the transitive nature of real life objects.
- **Derived/Sub and Base/Super classes**

A class from which another class is inheriting its properties is called base class and the class inheriting properties is known as a sub class and derived class.

1. Single (1:1)

→ when a class inherits from a single base class.

2. Hierarchical (1:M)

→ when several classes inherit from the same class.