॥ श्री ॥

Name: Hemant Vikas Patil
SE II Batch - B

Page No. 1
Date

Principles of Programming Languages

UNIT V

Q1. Define a package in Java?

→ a) Package can be created by simply including the package statement with package name at beginning of java source file. All class in that source file will be placed in the package.

b) In IDE's like eclipse, package name can be specified while creating the class if not specified all the class will belong to default package.

- A default package has no name. It is advised that we should always place our classes in a named package.

e.g. package mypackage;

d) From operating system's point of view, packages are nothing but folders in file system where classes files are placed.

- Java package statement can be seperate by dot (.) that will create package hierarchy.

eg. package mypack.java.mygames;

Q2. Describe methods overriding in Java

→ In Java method in a subclass can override method of a superclass. This happens when method in a subclass has some name type signature as in superclass.

b) In case of method overriding, when the method is called from within a subclass it will refer to method defined in sub so, the overridden method defined in superclass will be hidden.

c) If we want to call supperclass version of overriden method from subclass super can be used as
super.method - with same signature

d) Method overriding is considered valid only if name & signature of method remains same in superclass & subclass If signature differs it is considered overloading.

Q3. Illustrate Inheritance in Java

→ i) In Java a class can inherit other class by using extends keyword then, the definition of superclass is incorporated in subclass.

ii) So subclass can use members of super class. For example If class Y is created by extending X i.e. X superclass & Y subclass Then Y can access all the public members of X.

iii) The subclass object is used to directly access field in super class. Since superclass public members are inherited in subclass.

iv) Inheritance doesn't put any restriction in child & parent class Both can be treated independently in isolation as complete normal class.

v) The caller may create only superclass object & not for subclass if not needed & viceversa.

Q.4. Show how to call super class constructor.

→ i) A subclass can use super to call the super class constructor.

super(arg-list)

ii) The argument specified in parantheses is passed to super class constructor

iii) So we have to pass the exact number & type of arguments needed by super class constructor.

iv) Consider the program:

```
class X
{
    int a, b;
    public X(int a, int b)
    {
        this.a = a;
        this.b = b;
    }
    public void display()
    {
        System.out.println("Super displaying : a"
                +a + "b:" + b);
    }
}
```

```
class Y extends X
{
    int c, d;
    public Y (int c, int d, int a, int b)
    {
        super (a, b);
        this.c = c;
        this.d = d;
    }
    public void display ()
    {
        System.out.println ("sub display : c " + c +
                            "d :" + d );
    }
}
public class InherController
{
    public static void main (String args[])
    {
        Y y = new Y (3, 4, 1, 2);
        y.display ();
    }
}
```

o/p : sub display : c : 3 d : 4

Q.5. Demonstrate constructor execution sequence with example.

→ i) In Java, a constructor call sequence is different from execution sequence. i.e. it is reverse.

ii) Consider classes X, Y, Z class Z extends Y & Y extends X.

iii) If we create object of class Z, then Z class constructor calls Y class constructor similarly for Y and X.

iv) Hence X class constructor finishes execution first & returns to Y class constructor, similarly for Y & Z.

v) Constructor completes its execution in order of derivation through the constructors are called in reverse.

```
eg. class X
    {
      X()
       { System.out.println("Inside X"); }
    }
    class Y extends X
    { Y()
      { System.out.println("Inside Y"); }
    }
    class Z extends Y
      { Z()
      { System.out.println("Inside Z"); }
    }
    public class InherController
    {
```

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

O/P :- Inside X
Inside Y
Inside Z

Q6. Summarize access modifiers : private, default, public, protected.

→ i) Java provides access control mechanism to provide different level access to methods & data.
ii) When member of class (data or meth is declared public, it can be accessed anywhere in the code, by any classes It remains completely open for direct addressing using dot operator.
iii) When member is declared private, it can only be accessed by other members of same class.
iv) Protected access mode comes in picture when inheritance is involved
v) When no access modifier is specified then the member has default accessibility that is public in the same package also called as package accessibility.