NAME: Novojit Das Dipta
ID: CSE 066 07773
Batch: 66E
Course: Advanced Programming

Question 1:

What is a package? How can you define your own package?

Answer: Packages are containers for classes, where some of the classes are accessible are exposed and others are kept for internal purpose. Not only classes, subclasses and interface can be used within package.

To define a package, we have to you a keyword ' **package** '

Example:
package pkg; // *where, pkg is the name of the package*

We can also create **hierarcy** of package.

Syntax: package *pkg1*[.*pkg2*[.*pkg3*]];
Example: Package result.dept.cse; // *Where cse is pathed under the directory*
// *Result/Dept/CSE*

Java Runtime found these package in 3 ways -
1) Java's **Default** Directory
2) Using **CLASSPATH** environmental variable
3) **-classpath** option with **java** and **javac**

Question 2:

How a class or an interface is added to a package? Show with an example, how you can organize two classes in two different files into a single package?

Answer:

Part 1

```
package newpack;     // newpack is a package name
 public interface survival_guide;     // where survival guide is an interface name
{
     public abstract void book();
}

package newpack;     // newpack is our same package
 public class random implements survival guide;      // random is the class name

{
      public static void main (strings[] args);

     System.out.ln("book count");
}
```

Part 2

Protection.java FILE

```
package p1;  // p1 is package

public class Protection  //protection is the class name here

{

Int n = 1;
private int n_pri = 2;
protected int n_pro = 3;
public int n_pub = 4;
public Protection()
```

```
 {
System.out.println("base constructor");
System.out.println("n = " + n);
System.out.println("n_pri = " + n_pri);
System.out.println("n_pub = " + n_pub);
}
        }

                System.out.println("n_pro = " + n_pro);
```

Derived.java File

```
package p1;
class Derived extends Protection {
Derived() {
System.out.println("derived constructor");
System.out.println("n = " + n);   // class only
System.out.println("n_pri = "4 + n_pri);
System.out.println("n_pro = " + n_pro);
System.out.println("n_pub = " + n_pub);
}

}
```

Question 3:

Discuss the relative merits of using protected access vs. private access in superclass?

Answer:

protected members can be accessed by classes from the same package and subclasses of the declaring class from other packages.

It means that subclass from other package cannot access protected members of arbitrary instances of their superclasses, they can only access them on instances of their own type (where type is a compile-time type of expression, since it's a compile-time check). Using protected access enables the subclass to manipulate the protected members without using the access methods of the

superclass. If the superclass members are private, the methods of the superclass must be used to access the data. This may result in a decrease in performance due to the extra method calls.

Question 4:

i. Which instance variables can be accessed using object ob1 in the main() method?

ii. Which instance variables can be accessed using object ob2 in the main() method?

You should state the reasons on behalf of your answers.

Answer:

(I)

In this question we can find 2 packages named p and q. we already know that only public modifiers can be accessed between different packages. But if we use no modifiers (sometimes we call it default) then we can only access through in same package not different.

So after importing all from p package (**import p.*;** // *in the E.java code*) can access D.java and through that file it also can find B.java (Because of this code - **public class D extends B** in the D.java). Another thing, in the B.java file we can see a protected variable int m. Therefore, in the D.java class D is the subclass of class B. so we can easily say, obj 1 will get access that protected variable (through same package).

But they are in different package so we can only access public variables and protected member in B.java files. In the obj 1

(II)

Now its time to see whats going on obj 2 (in E.java file).

In the obj 2, The class C was imported through the package p. we all already know that we can only access the public modifiers in the different package. If there is a subclass. Then we only can access protected members through subclass. But

in the obj 2 there is no subclass access. So we can access the C.java file's protected members. We can only access the public member of that class.

Question 5:

What is an exception? Why do we need to handle exception in Java?

answer:

An exception is an event, which occurs during the execution of a program, that disrupts the normal flow of the program's instructions.

Java exception handling is managed via five keywords: try, catch, throw, throws, and finally. Briefly, here is how they work. Program statements that you want to monitor for exceptions are contained within a try block. If an exception occurs within the try block, it is thrown. In our code can catch this exception (using catch) and handle it in some rational manner. System-generated exceptions are automatically thrown by the Java runtime system. To manually throw an exception, use the keyword throw. Any exception that is thrown out of a method must be specified as such by a throws clause. Any code that absolutely must be executed after a try block completes is put in a finally block.