JAVA Assignment : 2

20DCS103

1. Give java packages and it's significance.

→ Packages are used in JAVA in order to prevent naming conflicts, to control access, to make searching / locating and usage of classes, interfaces, enumerations and annotations easier, etc.

→ A package can be defined as a grouping of related types [classes, interfaces, enumerations and annotations] providing access protection and namespace management.

→ Ex:- java.lang - bundles the fundamental classes.
java.io - classes for input, output functions are bundled in this package.

→ While creating a package, one should choose a name for the package and include a package statement along with that name at the top of every source file that contains the classes, interfaces types that one wants to include in the package.

20DCS103

2. Is it important that a Try block should be followed by Catch block in terms of exception handling. ?

→ No, it is not mandatory that each try block must be followed by a catch block in java.

→ After try blocks we can use either catch block or finally block.

→ Generally, thrown exceptions should be declared in the thrown clause of the method.

3. What is the meaning of multithreading?

→ Multi threading is a process of executing multiple threads simultaneously.

→ A thread is a lightweight sub-process, the smallest unit of processing.

→ Multiprocessing and multi threading, both are used to achieve multi tasking.

→ It is mostly used in games, animation, etc.

4. Why the runnable interface is being used in java ?

→ The runnable interface is being used in java as it provides a standard set of rules for the instances of classes which wish to execute code when they are active.

→ The most common use case of the runnable interface is when we want to override the run method.

→ public void run() takes no arguments. When the object of a class implementing runnable class is used to create a thread, then the run method is invoked in the thread which executes seperately.

5. Give the 2 - ways in java implementing multi - threading.

→ 1) Multi threading by extending thread class.

→ Here, in this example the method - 1 is implemented :

```
class MultithreadingExample extends Thread
{
    public void run()
    {
      System.out.println("Thread is in
          running state.");
    }

    public static void main(String args[])
    {
      MultithreadingExample e1 = new
              MultithreadingExample();
      e1.start();
    }
}
```

Output:

Thread is in running state.

→ 2) Multithreading by implementing runnable interface.

```
class MultithreadingExample implements Runnable
{
    public void run()
    {
      System.out.println("Thread is in running
                          state.");
```

```
    }

    public static void main(String args[])
    {
        MultiThreadingExample e1 = new
                        MultiThreadingExample();
        Thread tobj = new Thread(e1);
        tobj.start();
    }
}
```

Output :-

Thread is in running state.

6. When there are many changes that being required in the data, which one should you use ? StringBuffer or String.

→ Java provides StringBuffer class as a replacement of strings in places where there is a necessity to make a lot of modifications to strings of characters are required.

→ One can modify/manipulate the contents of a StringBuffer over and over again

20DCS103

without leaving behind a lot of new unused objects.

7. There is garbage collection in java, how it is being done ?

→ The garbage collection is a process of reclaiming the runtime unused memory automatically.

→ To make it clear, it is a way to destroy the unused objects.

→ There are many ways to do it :

1) By nulling the reference.

```
Employee e = new Employee();
e = null;
```

2). By assigning reference to another.

```
Student s1 = new Student();
Student s2 = new Student();
s1 = s2;
```

→ Now, s2 is ready/available for garbage collection.

8. How you can make sure that a resource is not being used by numerous threads simultaneously ?

→ When multiple threads are working on the same data and the value of our data is changing, that scenario is not thread safe and we will get inconsistent results.

→ This is how one can make sure that a resource is not being used by numerous threads simultaneously.

9. How to use concept of synchronization?

→ Synchronization is the process of allowing only one thread at a time to complete the particular task.

→ It means when multiple threads executing simultaneously and want to access the same resource at the same time, then the problem of inconsistency will occur.

→ So, synchronization is used to resolve inconsistency problem by allowing only one

thread at a time.

→ Example :-

```
class A
{
    synchronized void sum (int n)
    {
        Thread t = Thread.currentThread();
        for (int i = 1 ; i <= 5 ; i++)
        {
            System.out.println ( t.getName() +
                    " : " + (n + i) );
        }
    }
}

class B extends Thread
{
    A a = new A();
    public void run()
    {
        a.sum(10);
    }
}
```

20DCS103

```
class Test
{
  public static void main (String [] args)
  {
     B b = new B();
     Thread t1 = new Thread (b);
     Thread t2 = new Thread (b);

     t1.setName ("Thread A");
     t2.setName ("Thread B");

     t1.start();
     t2.start();
  }
}
```

→ Output :-

```
Thread A : 11
Thread A : 12
Thread A : 13
Thread A : 14
Thread A : 15
Thread B : 11
Thread B : 12
Thread B : 13
Thread B : 14
Thread B : 15
```

10. How can interface be used to support multiple inheritance ?

→ When we use multiple inheritance concept without using interfaces, it will give a compile time error.

→ But we can implements more than one interfaces on a single class which will provide the facilities of multiple inheritance in java.

→ Example :-

```
interface studentInt
{
    void sRead();
}
interface teacherInt
{
    void tRead();
}

class Person implements studentInt, teacherInt
{
    void sRead()
    {
        System.out.println("Student is reading.");
    }
```

```
    void tRead()
    {
        System.out.println(" Teacher is reading.");
    }
}.

public class Example
{
    public static void main( String [] args)
    {
        Person p1 = new Person();
        p1.sRead();
        p1.tRead();
    }
}
```

→ Output :-

```
Student is reading.
Teacher is reading.
```