CS 5 (Programming in Java) Unit II Short Answer Type

Q) Define package. Write down its advantages & types of packages.

A **java package** is a group of similar types of classes, interfaces and sub-packages.

Advantages of Java Packages:-

1) Java package is used to categorize the classes and interfaces so that they can be easily maintained.

2) Java package provides access protection.

3) Java package removes naming collision.

Types of Packages:-

Package in java can be categorized in two

1. built-in package
2. user-defined package

Q) Write the syntax to create and import Packages.

Creating a package:

Whenever we create user defined package in a Java program, we must use **package** statement . It must be the first statement in the program.

Syntax:

package pack1[.pack2[.pack3……[.packn]…..]];

Here, **package** is a keyword for creating user defined packages, pack1 represents top level package and pack2 to packn represent sub packages.

Examples:

```
package   p1;
package    p1.p2;
```

Importing a Package or Class:

To use a class or a package from the library, we need to use the import keyword:

Syntax

import *packagename.Class*; // Import a single class
import *packagename*.*; // Import the whole package

Example:- To use the Scanner class, **which is used to get user input**, write the following code:

a) **import java.util.Scanner;**

Here:
→ **java** is a top level package
→ **util** is a sub package
→ **Scanner** is a class which is in the sub package **util**.

b) **import java.util.*;**
Here:
→ **java** is a top level package
→ **util** is a sub package
→ * indicates all the classes & interfaces defined in util subpackage.

Q) Write about Built-in Packages

The Java API is a library of prewritten classes, that are free to use.

There are many built-in packages. Some of them are as follows :

Package Name	Functionality
java.lang	Basic language fundamentals
java.util	Utility classes & data structure classes
java.io	File handling operations
java.math	Mathematical functions
java.sql	JDBC to access databases

java.awt	AWT for native GUI components
javax.swing	Lightweight GUI components

Q) Explain about Access protection in Java
Or
Explain about Access modifiers in Java

Access modifiers define the scope of the class, variables, methods, and constructors.

For example, private members are accessible within the same class .

Java provides many levels of security that provides the use of variables and methods within the classes, subclasses, and packages.

There are four types of Java access modifiers:

- **private :** applied to variables, constructors, methods, and inner classes (not top-level classes)
- **default (No Modifier) :** applied to variables, constructors, methods, and classes
- **protected :** applied to variables, constructors, methods, and inner classes (not top-level classes)
- **public :** applied to variables, constructors, methods, and classes

	Private	No Modifier	Protected	Public
Same class	Yes	Yes	Yes	Yes
same package subclass	No	Yes	Yes	Yes
same package non - subclass	No	Yes	Yes	Yes
Different package subclass	No	No	Yes	Yes
Different package mon-subclass	No	No	No	Yes

Q) Explain about wrapper classes in Java.

The **wrapper classes in Java** provide the mechanism *to convert primitive datatypes into objects and objects into primitive types*.

The eight classes of the *java.lang* package are known as wrapper classes.

Primitive Type	Wrapper class
boolean	Boolean
char	Character
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double

Boxing

The conversion of primitive data type into its corresponding wrapper class

Example :

```
int a = 12;

Integer c = a;    //  autoboxing
```

Unboxing

The conversion of object (wrapper type) into the primitive type.

Example :

```
Integer  a = 5;
int  b = a;    //  auto  unboxing
```

Q) Define string. Explain how to create strings.

String is a sequence of characters.
E.g. “Hello” is a string of 5 characters.

In java, String is an immutable object which means it is constant and cannot be changed once it has been created.

Creating a String

There are two ways to create a String:

1. String literal
2. Using new keyword

1. **String literal**

Assign a String literal to a String instance:

```
String   str1 = "Welcome";
```

2. **Using New Keyword**

We can create strings using new operator like shown below

```
String str1 = new String("Welcome");
```

Q) Write the methods of StringBuffer class.

Java StringBuffer class is used to create mutable (modifiable) string. The StringBuffer class in java is same as String class except it is mutable i.e. it can be changed.

Important methods of StringBuffer class:

1. append() method concatenates the given argument with this string.

2.replace() method replaces the given string from the specified beginIndex to endIndex-1.

3.reverse() method reverses the current string.

4.capacity() method returns the current capacity of the buffer.

Q) What are the types of Exceptions.

There are mainly two types of exceptions: checked and unchecked. Here, an error is considered as the unchecked exception. According to Oracle, there are three types of exceptions:

1. Checked Exception
2. Unchecked Exception
3. Error

1.Checked Exception

The classes which directly inherit Throwable class except RuntimeException and Error are known as checked exceptions e.g. IOException, SQLException etc. Checked exceptions are checked at compile-time.

2) Unchecked Exception

The classes which inherit RuntimeException are known as unchecked exceptions e.g. ArithmeticException, NullPointerException, ArrayIndexOutOfBoundsException etc. Unchecked exceptions are not checked at compile-time, but they are checked at runtime.

3) Error

Error is irrecoverable e.g. OutOfMemoryError, VirtualMachineError, AssertionError etc.

Q) Explain briefly about Java Exception Keywords.

There are 5 keywords which are used in handling exceptions in Java.

Keyword	Description
try	The "try" keyword is used to specify a block where we should place exception code. The try block must be followed by either catch or finally. It means, we can't use try block alone.
catch	The "catch" block is used to handle the exception. It must be preceded by try block which means we can't use catch block alone. It can be followed by finally block later.
finally	The "finally" block is used to execute the important code of the program. It is executed whether an exception is handled or not.
throw	The "throw" keyword is used to throw an exception.
throws	The "throws" keyword is used to declare exceptions. It doesn't throw an exception. It specifies that there may occur an exception in the method. It is always used with method signature.

Q) Define thread .

A **thread** is a light-weight sub-process, smallest part of processing.

One thread can **run concurrently** with the other smallest parts(other threads) of the same process.

The process of executing multiple threads simultaneously is known as **multithreading.**

Threads are independent that's the reason if an exception occurs in one thread, it doesn't affect the execution of other threads.

Advantages of Java Multithreading

1. Multithreading provides simultaneous execution of two or more parts of a program to **maximum utilize the CPU time**.

2. All threads of a process share the common memory.

Q) What are Thread Life-Cycle States?

A thread can be in one of the following states:

NEW – The thread is created but not started.

```
ThreadDemo obj=new ThreadDemo( );
```

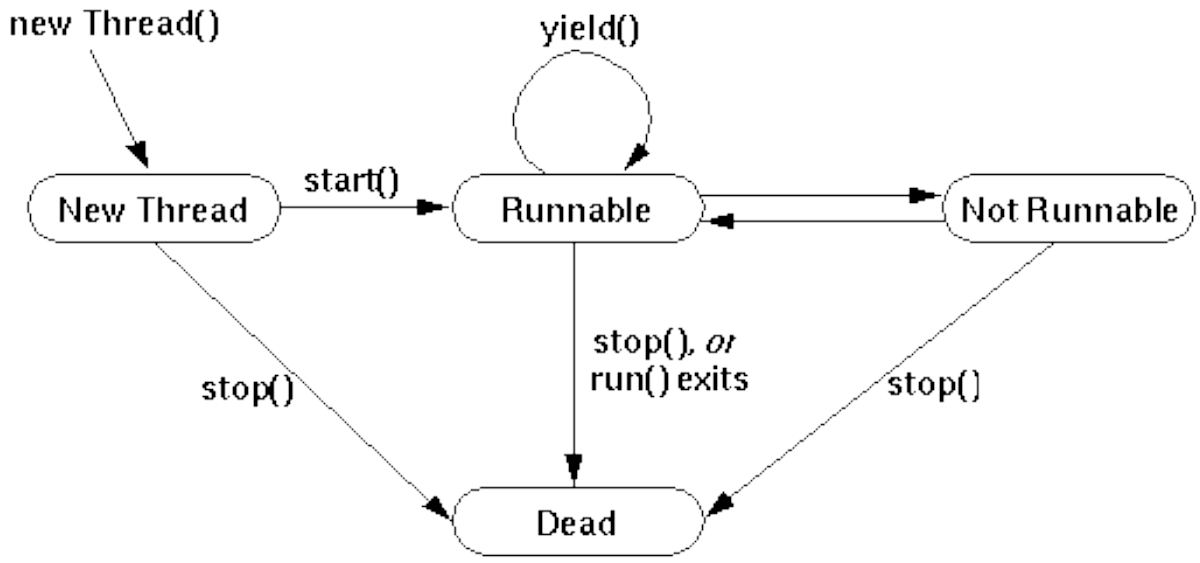
RUNNABLE – The thread is ready for execution by JVM. But it is waiting in the queue for getting the processor. A thread enters this state after start() method is called.

```
obj.start( );
```

NOT RUNNABLE - A thread is said to be in NOT RUNNABLE state if it is in any of the following 3 states - WAITING, TIMED_WAITING, BLOCKED.

TERMINATED – This state is reached when the thread has finished its execution.

A thread can be in only one state at a given point of time.



Q) List the methods in Thread class.

Java provides **Thread class** in java.lang package to achieve thread programming. Thread class provides constructors and methods to create and perform operations on a thread. Some of them are given below:

- getName() : It is used for Obtaining a thread's name
- getPriority() : Obtain a thread's priority
- isAlive() : Determine if a thread is still running
- join() : Wait for a thread to terminate
- run() : Entry point for the thread
- sleep() : suspend a thread for a period of time
- start() : start a thread by calling its run() method

Q) Explain about main thread.

Even though programmer does not create a thread, every Java program has a thread called the **main thread**.

When a Java program starts executing, JVM created the **main thread** and calls the main( ) method.

The **main thread** spawns the other threads. These spawned threads are called **child threads**.

The **main thread** is always the last to finish executing because it is responsible for releasing the resources such as network connections.

The programmer can control the **main thread** using the **currentThread( )** method. This method is defined in java.lang.Thread class.

Sample code:

```
Thread obj=Thread.currentThread( );
System.out.println("Current thread: " + obj);
```

Output :-
Current thread: Thread[main, 5, main]

Q) Briefly write about Thread Priorities.

Every thread has a priority.

The thread scheduler determines the order in which threads are scheduled based on thread priorities. The high priority threads will usually run before and more frequently than lower priority threads.

Thread priority is defined by 3 constants in **Thread** class as:

- MAX_PRIORITY, which is 10
- NORM_PRIORITY, which is 5
- MIN_PRIORITY, which is 1

The priority of a thread lies between 1 to 10. Default priority of each thread is NORM_PRIORITY, which is 5.

We can get the priority of thread using

int Thread.getPriority() method.

We can set the priority of thread using

Thread.setPriority(int priority**)** method.

Sample Code:

```
System.out.println("running thread priority is:"+Thread.currentThread().getPriority());
ThreadDemo t1=new ThreadDemo ();
t1.setPriority(Thread.MIN_PRIORITY);
```

Note : Write the sample program given below only if it is asked in the question.

Sample Program

```
class PriorityDemo extends Thread
{
    public void run()
    {
    System.out.println("running thread name is:"+Thread.currentThread().getName());
    System.out.println("running thread priority is:"+Thread.currentThread().getPriority());
    }
    public static void main(String args[])
    {
            PriorityDemo t1=new PriorityDemo ();
            PriorityDemo t2=new PriorityDemo ();
            t1.setPriority(Thread.MIN_PRIORITY);
            t2.setPriority(Thread.MAX_PRIORITY);
            t1.start();
            t2.start();
    }
```

}

Output:-

running thread name is:Thread-0
running thread priority is:10
running thread name is:Thread-1
running thread priority is:1

Q) What is thread synchronization?

At times when more than one thread try to access a shared resource, we need to ensure that resource will be used by only one thread at a time. The process by which this is achieved is called **synchronization**.

Synchronization is a mechanism *to control the access of multiple threads to any shared resource*.

synchronized method:

Synchronized method is used to lock an object for any shared resource.

Q) Write about Java I/O.

- java.io package provides separate classes & interfaces for reading & writing data.
- So, if we use any I/O class, we need to write **import java.io.*;** statement at the top of the program.
- Java I/O facility is based on streams.
- Java I/O package has both byte stream classes & character stream classes.
- java.io package contains two top level byte stream abstract classes:
    - java.io.InputStream (for reading bytes data)
    - java.io.OutputStream (for writing bytes data)

- It also contains two top level character stream abstract classes:
    - java.io.Reader (for reading character data)
    - java.io.Writer (for writing character data)

- The subclasses of the above classes are actually used for reading & writing data.

Q) List the methods of File class:

The File class has many useful methods for creating and getting information about files. Some of them are :

Return Type	Method	Description
boolean	createNewFile()	It atomically creates a new, empty file if and only if a file with this name does not exist.
boolean	canWrite()	It tests whether the file can be modified or not
boolean	canExecute()	It tests whether the file can be executed or not
boolean	canRead()	It tests whether the file can be read or not
boolean	isDirectory()	It tests whether the file is a directory or not.
boolean	isFile()	It tests whether the file is a normal file or not.
String	getName()	It returns the name of the file.
boolean	exists()	Tests whether the file exists
long	length()	Returns the size of the file in bytes
boolean	mkdir()	It creates the directory named by this abstract pathname.

Q) Write about Scanner Class and its methods.

The Scanner class is used to get user input i.e., to read values from keyboard and store them in variables.
It is defined in the java.util package. So, if we use Scanner class, we need to write **import java.util.Scanner;** statement at the top of the program.

Useful methods

Method	Description
nextBoolean()	Reads a boolean value from the user
nextByte()	Reads a byte value from the user
nextDouble()	Reads a double value from the user
nextFloat()	Reads a float value from the user
nextInt()	Reads a int value from the user
nextLine()	Reads a String value from the user
nextLong()	Reads a long value from the user
nextShort()	Reads a short value from the user

Sample Program :

```
import java.util.Scanner;

class ScannerDemo {
```

```
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);

    System.out.println("Enter name, age and salary");

    // String input
    String name = myObj.nextLine();

    // Numerical input
    int age = myObj.nextInt();
    double salary = myObj.nextDouble();

    System.out.println("Name: " + name);
    System.out.println("Age: " + age);
    System.out.println("Salary: " + salary);
  }
}
```

Output:

```
Enter name, age and salary
bbcit
27
50000.00
Name:bbcit
Age:27
Salary:50000.00
```


Q) Write about BufferedInputStream Class & its methods.

It is used for buffering the input.

Method	Description
int available()	Returns the number of bytes that can be read.

int read()	Reads a byte of data from the input stream.
int read(byte[] b)	Reads up to **b.length** bytes of data from the input stream and stores it in ***b*** array.
int read(byte[] b, int off, int len)	Reads up to **len** bytes of data from the input stream into ***b*** array starting at ***off***.
long skip(long x)	Skips over and discards x bytes of data from the input stream.
void close()	Closes the stream.

Q) Write about BufferedOutputStream Class & its methods

It is used for buffering the output.

Method	**Description**
void flush()	Flushes the output stream.
void write(byte[] ary, int off, int len)	Writes **len** bytes from the byte array ary starting at **off** to the file output stream.
void write(int b)	Writes the specified byte to the file output stream.