

Second Unit test

(1)

Name : Aman Yadav

Roll no : 2020 7005

Subject : Programming in Java Class : B.Sc. cs 2nd Semester.

Date : 12-08-2021

Answers No - 4

Runnable interface :

Java.lang.Runnable is an interface that is to be implemented by a class whose instance are intended to be executed by a thread. There are two ways to start a new thread. Subclass Thread and implement Runnable. There is no need of subclassing thread when a task can be done by overriding only run() method of runnable.

Example:

```
public class RunnableDemo {
    public static void main(String[] s) {
        System.out.println("Main thread is -" + Thread.currentThread()
            .getName());
        Thread T1 = new Thread(new RunnableDemo().new RunnableImpl());
        T1.start();
    }
    private class RunnableImpl implements Runnable {
        public void run() {
            System.out.println(Thread.currentThread().getName()
                + ", executing run method!");
        }
    }
}
```

output:

Main thread is - main

Thread-0. Executing run() method!

Answer No - 6

i) Thread class methods: Thread class is main class on which java multithreading system is based. Thread class along with its companion interface Runnable will be used to create and run threads for utilizing multithread feature of java. It provide constructors and methods to support multithreading. It extends object class and implements runnable interface.

<u>methods</u>	<u>Description</u>
setName()	to give thread a name
getName()	return's thread's name
join()	wait for a thread to end
start()	start a thread by calling run().
run()	entry point for a thread.

ii) Garbage Collection: In java, garbage means unreferenced objects. Garbage collection is process of reclaiming the runtime unused memory automatically. In the other word it is a way to destroy the unused object. in c++ we use delete(). But, in java it is performed automatically, so, java, provides better management.

Example :-

```
public class Test Garbage {  
    public void finalize() {  
        System.out.println("object is garbage collected.");  
    }  
}
```

```

public static void main (String s[]) {
    TestGarbage t s1 = new TestGarbage ();
    TestGarbage t s2 = new TestGarbage ();
    s1 = null ;
    s2 = null ;
    System.gc ();
}
}

```

output: object is garbage collected
 object is garbage collected

Answer No - 2

Hierarchical Inheritance: ~~or~~ when two or more subclasses are derived from one superclass than this inheritance is known as hierarchical inheritance.

```

Example: class Animal {
    void eat() {
        System.out.println ("eating!");
    }
}
class Dog extends Animal {
    void bark() {
        System.out.println ("Barking!");
    }
}
class Cat extends Animal {
    void meow() {
        System.out.println ("Meowing!");
    }
}
class Test {

```

(4)

```
public static void main(String[] s) {
```

```
    Cat c = new Cat();
```

```
    c.meow();
```

```
    c.eat();
```

```
    c.bark(); // not accessible because it's not extended
```

```
}
```

```
}
```

output: Meowing!
eating!

Ans No - 1

A. (b) only II

B. (c) java.util.*

C. (b) throw