Question
Consider below code:

```
public class Test {
    static Double d1;  // d1 =null
    static int x = d1.intValue();// null.intValue() ---> NullPointerException

    public static void main(String[] args) {
        System.out.println("HELLO");
    }
}
```

On execution, does Test class print "HELLO" on to the console?
A.Yes HELLO is printed on the console
B.NO Hello is not printed on the console

Answer: B

Question
Consider below code:

```
public class Test {
    static Double d1; // static variable ====> d1 =null
    int x = d1.intValue();//instance variable ====> only upon creating an object

    public static void main(String[] args) {
        System.out.println("HELLO");//HELLO
    }
}
```

On execution, does Test class print "HELLO" on to the console?
A.Yes HELLO is printed on the console
B.NO Hello is not printed on the console

Answer: A

Question:
What will be the result of compiling and executing Test class?

```
public class Test {
    public static void main(String[] args) {
        Error obj = new Error();
        boolean flag1 = obj instanceof RuntimeException; //Line n1
        boolean flag2 = obj instanceof Exception; //Line n2
        boolean flag3 = obj instanceof Error; //Line n3
        boolean flag4 = obj instanceof Throwable; //Line n4
        System.out.println(flag1 + ":" + flag2 + ":" + flag3 + ":" + flag4);
    }
}
```

A. Compilation Error
B. false:false:true:true
C. false:true:true:true
D. true:true:true:true
E. false:true:true:false

Note: Error and RunTimeExcpetion no relation in hierarchy as parent and child
Error and Exception no relation in hierarchy as parent and child

Answer: A

```
String s = "sachin";
class Student{}
Student s1 = new Student();
System.out.println( s instanceof String);//true
```

```
System.out.println( s instanceof StringBuffer);//CE(String and StringBuffer no
relationship)
System.out.println( s1 instanceof Runnable);//false
System.out.println( null instanceof Stringbuilder);//false
```

Fill in the blanks for the definition of java.lang.Error class:
public class java.lang.Error extends _____ {...}
A. RunTimeException
B. Exception
C. Throwable

Answer: C

Question>
Given code of Test.java file:

```
public class Test {
    public static void main(String[] args) {
        System.out.println(new RuntimeException()); //Line n1
        System.out.println(new RuntimeException("HELLO")); //Line n2
        System.out.println(new RuntimeException(new
RuntimeException("HELLO"))); //Line n3
    }
}
```

Does above code compile successfully?
A. Yes
B. No

Answer: A

Question>
Given code of Test.java file:

```
interface ILogger {
    void log();
}

public class Test {
    public static void main(String[] args) {
        ILogger [] loggers = new ILogger[2]; //Line n1  ==JVM ==> loggers[0] =null;
loggers[1] =null;
        for(ILogger logger : loggers)
            logger.log(); //Line n2 ====JVM====> NullPointerException
    }
}
```

What will be the result of compiling and executing Test class?
A. Line n1 causes compilation error
B. Line n2 causes compilation error
C. Exception is thrown at runtime
D. No output is displayed but program terminates succesfully.

Answer: C

valid
====

```
class MyThread1 implements Runnable{}
class MyThread2 implements Runnable{}
Runnable[] runnable = new Runnable[2];
runnable[0] =new MyThread1();
runnable[1] =new Mythread2();
```