LNMIIT/B.Tech./CSE/CORE/2018-19/ODD/AP/EndTerm

LNMIIT
The LNM Institute of Information Technology

Advanced Programming (AP)
End Term Exam
Date: November 30, 2018

Time: 3 Hours

Max. Marks: 40

NOTE: *No query will be entertained.*
There are two parts. The part A should be answered before part B.

PART A

Q1. In an inheritance hierarchy, superclass say *Employee* has been declared as an abstract class and subclasses such as *Manager*, *clerk* etc are normal classes. In this situation, show how a Java program will implement the concept of run-time polymorphism?(Hint: each class may have a method named *showDetails()*)? . **[4]**

Q2. Complete the code given below (in the blank box [ ]) to produce given desired output? **[4x1 = 4]**

```
interface A {
        void meth1();
}
interface B extends A {
        void meth2();
}
class Outer [   i   ] {
    [      ii      ]
    int outer_x = 100;
    void display() {
        [      iii      ]
    }
    class Inner {
            int inner_x=50;
            void display() {
                [      iv      ]
            }
    }
}
class Demo{
        public static void main(String arg[]) {
                Outer ob = new Outer();
                ob.meth1();
                ob.meth2();
                ob.display();
        }
}
```

Desired output:

```
Inside method meth1
Inside method meth2
Display method of Inner class display: outer_x = 100
Display method of Outer class display: inner_x = 50
```

PART B

Q3. What will happen if we don't override *run()* method of thread in Java? **[1]**

Q4. Why JVM terminates the daemon thread if no user threads are remaining? **[1]**

Q5. In multithreading environment, what if we don't want to make the entire method synchronized? **[1]**

Q6. A user program is getting NullPointerException and NumberFormatException. However, the program is not handling any kind of above-mentioned exceptions and still the program has compiled and executed successfully (with exception messages). Why? **[1]**

Q7. What are the possible ways to properly stop a thread in JAVA? Explain with example. [4]

Q8. In the following example, is it possible for multiple threads to access the methods simultaneously? Justify your answer. [2]

```
public class Counter {
    private int count = 0;
    public static synchronized int getCount(){
        return count;
    }
    public synchronized setCount(int count){
        this.count = count;
    }
}
```

Q9. Which type of exception the following tasks might throw [.5x6=3]

A. Access an array beyond its length

B. Opening and reading the contents of file stored on your system

[✓ C.] Closing a file

D. Dividing a number by zero

[✓ E.] Passing an illegal argument to a method

F. Accessing a method through null object

Which of the above tasks might throw an exception that the compiler would care about? Write the possible ways to handle that type of exception. [1+2= 3]

Q10. Differentiate the following: [write any four differences] [2x2=4]

A. HashMap and Hashtable

B. ArrayList and Vector

Q11. Write the output/error of the following programs with reason. [2+2=4]

(A)

```
import java.util.*;
public class Example{
public static void main(String args[]){
   Set<Integer> set=new HashSet<Integer>();
   set.add(9);
   set.add(7);
   set.add(9);
   set.add(7);
   set.add(10);
   set.add(7);
   Iterator<Integer> itr=set.iterator();
   while(itr.hasNext()){
       System.out.println(itr.next());
   }
} }
```

(B)

```
import java.util.*;
public class EndTerm{
public static void main(String args[]){
   HashSet<String> hset=new HashSet<String>();
   hset.add("Object");
   hset.add("Oriented");
   hset.add("Programming");
   ListIterator<String> itr=hset.listIterator();
   while(itr.hasNext()){
       System.out.println(itr.next());
   }
} }
```

Q12. On a given database, say Employee, user wants to perform select, insert and update queries. How the user can execute all the queries in one execute? Only write all the necessary steps. [6]

Q13. Create an object of a *statement* class to produce a *ResultSet* having following properties: [2]

- Scrollable in both forward and backward direction
- Sensitive to changes made by others to the database that occur after the result set was created
- Updatable