**Time allowed: 90 Minutes Max. Marks: 40**

**General Instructions:**

* **Follow the instructions given in each section.**
* **Make sure that you attempt the questions in order.**

**SECTION-A (10\*1 mark=10 marks)**

***(All questions are compulsory)***

1) What is multithreaded programming?

a) It’s a process in which two different processes run simultaneously

**b) It’s a process in which two or more parts of same process run simultaneously**

c) It’s a process in which many different process are able to access same information

d) It’s a process in which a single process can access information from many sources

2) Which of these are types of multitasking?

a) Process based

b) Thread based

**c) Process and Thread based**

d) None of the mentioned

3) Thread priority in Java is?

**a) Integer**

b) Float

c) double

d) long

4) Which of the following classes can catch all exceptions which cannot be caught?

a) RuntimeException

**b) Error**

c) Exception

d) ParentException

5) Which of the following operators is used to generate instance of an exception which can be thrown using throw?

a) thrown

b) alloc

c) malloc

**d) new**

6) If a class inheriting an abstract class does not define all of its function then it will be known as?

**a) Abstract**

b) A simple class

c) Static class

d) None of the mentioned

7) An abstract class with 100% abstract methods is equivalent to \_\_\_\_\_

a) Concrete class

b) Virtual Class

**c) Interface**

d) All the above

8) What is the maximum number of Java Class files that can be kept inside a single Java Package?

a) 8

b) 64

c) 128

**d) Unlimited**

9) The name of a package is the name of the \_\_\_ in Java.

a) folder

b) All parent folders separated by DOT symbols

c) All parent packages separated by DOT symbols

**d) All the above**

10) Which of these methods is used to print stack trace?

a) obtainStackTrace()

**b) printStackTrace()**

c) getStackTrace()

d) displayStackTrace()

**SECTION-B (5\*2 mark=10 marks)**

***(All questions are compulsory)***

11) Determine output of the following program code?

public class Test{

public static void main(String args[]){

int i;

try{

i = calculate();

System.out.println(i);

}catch(Exception e){

System.out.println("Error occured");

}

}

static int calculate(){

return (7/2);

}

}

**a) 3**

b) 3.5

c) Error occured

d) Compilation Error

12) public class Test{

public static void main(String args[]){

try{

int a = Integer.parseInt("four");

}

}

}

Which exception could be handled by the catch block for above?

a) IllegalStateException

**b) NumberFormatException**

c) ClassCastException

d) ArrayIndexOutOfBoundsException

13) What will be the output of the following Java program?

class String\_demo

{

public static void main(String args[])

{

char chars[] = {'a', 'b', 'c'};

String s = new String(chars);

System.out.println(s);

}

}

a) a

b) b

c) c

**d) abc**

14) What is the output of the below Java program with an Interface?

interface Car

{

int basePrice=1000;

}

public class InterfaceTest2 implements Car

{

void changePrice()

{

basePrice = 2000;

System.out.print(basePrice);

}

public static void main(String[] args)

{

new InterfaceTest2().changePrice();

}

}

a) 1000

b) 2000

**c) Compiler error**

d) None of the above

15) What kind of exception is being thrown if Wait(), Pulse() or PulseAll() is called from code that is not within synchronized code?

a) System I/O Exception

b) DivideByZero Exception

**c) SynchronizationLockException**

d) All of the mentioned

**SECTION-C(Coding Question) (2x5 marks=5 marks)**

Q16) Given two Lists of strings s1 and s2, you have to count the number of strings in s2 which is either a suffix or prefix of at least one string of s1.

**Input**:

s1 = ["cat", "catanddog", "lion"]

s2 = ["cat", "dog", "rat"]

**Output**:

2

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | s1 = ["jrjiml", "tchetn", "ucrhye", "ynayhy",  "cuhffd", "cvgpoiu", "znyadv"]  s2 = ["jr", "ml", "cvgpoi", "gpoiu", "wnmkmluc",  "geheqe", "uglxagyl", "uyxdroj"] | s1 = [  "rainbow", "rain", "train"]  s2 = ["train", "rain", "bow"] | s1 =  ["pizza", "sandwich", "subway"]  s2= ["runway", "maza", "burger"] |
| **Output** | 4 | 3 | 0 |

Solution :

**import java.util.\*;**

**import java.io.\*;**

**public class MyClass**

**{**

**public static void main(String args[])throws IOException**

**{**

**String s1[]={"cat", "catanddog", "lion"};**

**String s2[]={"cat", "dog", "rat"};**

**Solution ob=new Solution();**

**int ans=ob.prefixSuffixString(s1,s2);**

**System.out.println(ans);**

**}**

**}**

**class Solution**

**{**

**public int prefixSuffixString(String s1[],String s2[])**

**{**

**int count = 0;**

**for (String s : s2) {**

**for (String word : s1) {**

**if (word.startsWith(s) || word.endsWith(s)) {**

**count++;**

**break;**

**}**

**}**

**}**

**return count;**

**}**

**}**

Q17) Given two strings s and t. Return the minimum number of operations required to convert s to t.

The possible operations are permitted:

* Insert a character at any position of the string.
* Remove any character from the string.
* Replace any character from the string with any other character.

**Input**:

s = "piza", t = "pizza"

**Output**: 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | s = "piza", t = "pizza" | s = "samsung", t = "rang" | s = "panda", t = "grandma" |
| **Output** | 1 | 4 | 3 |

Solution :

**import java.util.\*;**

**import java.lang.\*;**

**import java.io.\*;**

**public class MyClass {**

**public static void main(String[] args) throws IOException {**

**String s = "piza";**

**String t ="pizza";**

**Solution ob = new Solution();**

**int ans = ob.editDistance(s, t);**

**System.out.println(ans);**

**}**

**}**

**class Solution {**

**public int editDistance(String s, String t) {**

**int m=s.length();**

**int n=t.length();**

**int prev[]=new int[n+1];**

**int curr[]=new int[n+1];**

**for(int j=0; j<=n; j++){**

**prev[j]=j;**

**}**

**for(int i=1; i<=m; i++){**

**curr[0]=i;**

**for(int j=1; j<=n; j++){**

**if(s.charAt(i-1)==t.charAt(j-1)){**

**curr[j]=0+prev[j-1];**

**}**

**else{**

**curr[j]=1+Math.min(curr[j-1],Math.min(prev[j],prev[j-1]));**

**}**

**}**

**prev=curr.clone();**

**}**

**return prev[n];**

**}**

**}**

**SECTION-D (Coding Question)(1x10 mark=10 mark)**

Q18) Given an integer N, the task is to write Java Program to print the first N natural numbers in increasing order using two threads.

**Input**: N = 10

**Output**: 1 2 3 4 5 6 7 8 9 10

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | N = 18 | N = 5 | N = 7 |
| **Output** | 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 | 1 2 3 4 5 | 1 2 3 4 5 6 7 |

Solution :

**public class JavaProgram {**

**// Starting counter**

**int counter = 1;**

**static int N;**

**// Function to print odd numbers**

**public void printOddNumber()**

**{**

**synchronized (this)**

**{**

**// Print number till the N**

**while (counter < N) {**

**// If count is even then print**

**while (counter % 2 == 0) {**

**// Exception handle**

**try {**

**wait();**

**}**

**catch (**

**InterruptedException e) {**

**e.printStackTrace();**

**}**

**}**

**// Print the number**

**System.out.print(counter + " ");**

**// Increment counter**

**counter++;**

**// Notify to second thread**

**notify();**

**}**

**}**

**}**

**// Function to print even numbers**

**public void printEvenNumber()**

**{**

**synchronized (this)**

**{**

**// Print number till the N**

**while (counter < N) {**

**// If count is odd then print**

**while (counter % 2 == 1) {**

**// Exception handle**

**try {**

**wait();**

**}**

**catch (**

**InterruptedException e) {**

**e.printStackTrace();**

**}**

**}**

**// Print the number**

**System.out.print(**

**counter + " ");**

**// Increment counter**

**counter++;**

**// Notify to 2nd thread**

**notify();**

**}**

**}**

**}**

**public static void main(String[] args)**

**{**

**// Given Number N**

**N = 10;**

**// Create an object of class**

**JavaProgram mt = new JavaProgram();**

**// Create thread t1**

**Thread t1 = new Thread(new Runnable() {**

**public void run()**

**{**

**mt.printEvenNumber();**

**}**

**});**

**// Create thread t2**

**Thread t2 = new Thread(new Runnable() {**

**public void run()**

**{**

**mt.printOddNumber();**

**}**

**});**

**// Start both threads**

**t1.start();**

**t2.start();**

**}**

**}**