**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) Which is the less restrictive access modifier in Java?

**a) public**

b) private

c) protected

d) default

2) Which is the most restrictive access modifier in Java?

a) public

**b) private**

c) protected

d) default

3) To access a protected variable or method of a Class outside the package, you need to \_\_\_\_ in Java.

a) Create an instance and call the protected variable or method

**b) Create a Subclass and call the protected variable or method**

c) A and B

d) None of the above

4) What happens when we access the same variable defined in two interfaces implemented by the same class?

a) Compilation failure

b) Runtime Exception

c) The JVM is not able to identify the correct variable

**d) The interfaceName.variableName needs to be defined**

5) What happens when a constructor is defined for an interface?

**a) Compilation failure**

b) Runtime Exception

c) The interface compiles successfully

d) The implementing class will throw exception

6) What is mutex?

**a) a mutually exclusive synchronization object**

b) can be acquired by more than one thread at a time

c) helps in sharing of resource which can be used by one thread

d) all of the mentioned

7) What is Semaphore?

a) Grant more than one thread access to a shared resource at the same time

b) Useful when a collection of resources is being synchronized

c) Make use of a counter to control access to a shared resource

**d) All of the mentioned**

8) Which of these method of class String is used to remove leading and trailing whitespaces?

a) startsWith()

**b) trim()**

c) Trim()

d) doTrim()

9)Which of these class is used to create an object whose character sequence is mutable?

a) String()

**b) StringBuffer()**

c) Both of the mentioned

d) None of the mentioned

10) Which of these methods return description of an exception?

a) getException()

**b) getMessage()**

c) obtainDescription()

d) obtainException()

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

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

11) What will be the output of the following Java code?

class exception\_handling

{

public static void main(String args[])

{

try

{

throw new NullPointerException ("Hello");

}

catch(ArithmeticException e)

{

System.out.print("B");

}

}

}

a) A

b) B

c) Compilation Error

**d) Runtime Error**

12) What will be the output?

public interface InfA{

protected String getName();

}

public class Test implements InfA{

public String getName(){

return "test-name";

}

public static void main (String[] args){

Test t = new Test();

System.out.println(t.getName());

}

}

a) test-name

b) Compilation fails due to an error on lines 1

**c) Compilation fails due to an error on lines 2**

d) Compilation succeed but Runtime Exception

13) What is the result of the following program?

public static synchronized void main(String[] args) throws

InterruptedException {

Thread f = new Thread();

f.start();

System.out.print("A");

f.wait(1000);

System.out.print("B");

}

a) It prints A and B with a 1000 seconds delay between them

b) It only prints A and exits

c) It only prints B and exits

**d) A will be printed, and then an exception is thrown.**

14) String s=new String("TIH");

How many objects are created for the above statement in java?

a) 0

b) 1

**c) 2**

d) 3

15) What will be the output of the following Java code?

class Myexception extends Exception

{

int detail;

Myexception(int a)

{

detail = a;

}

public String toString()

{

return "detail";

}

}

class Output

{

static void compute (int a) throws Myexception

{

throw new Myexception(a);

}

public static void main(String args[])

{

try

{

compute(3);

}

catch(Myexception e)

{

System.out.print("Exception");

}

}

}

a) 3

**b) Exception**

c) Runtime Error

d) Compilation Error

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

Q16) Given a string of lowercase alphabets, count all possible substrings (not necessarily distinct) that has exactly k distinct characters.

**Input**: abc, k = 2

**Output**: 2

Possible substrings are {“ab”, “bc”}

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | aba, k = 2 | abcbaa, k=3 | Pizza, k=2 |
| **Output** | 3 | 8 | 5 |

Solution :

**import java.util.Arrays;**

**public class JavaProgram**

**{**

**// Function to count number of substrings with exactly k unique characters**

**int countkDist(String str, int k)**

**{**

**// Initialize result**

**int res = 0;**

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

**// To store seen characters from 'a' to 'z'**

**boolean seen[] = new boolean[26];**

**// Consider all substrings beginning with str[i]**

**for (int i = 0; i < n; i++)**

**{**

**int distCount = 0;**

**// mark all chars as unseen**

**Arrays.fill(seen, false);**

**// Consider all substrings between str[i..j]**

**for (int j=i; j<n; j++)**

**{**

**// If this is a new character for this**

**// substring, increment dist\_count.**

**if (!seen[str.charAt(j) - 'a'])**

**distCount++;**

**// mark current char as seen**

**seen[str.charAt(j) - 'a'] = true;**

**// If distinct character count becomes k,**

**// then increment result.**

**if (distCount == k)**

**res++;**

**}**

**}**

**return res;**

**}**

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

**{**

**JavaProgram ob = new JavaProgram();**

**String ch = "abcbaa";**

**int k = 3;**

**System.out.println("Total substrings with exactly " +**

**k + " distinct characters : "**

**+ ob.countkDist(ch, k));**

**}**

**}**

Q17) Given a string and a string dictionary, find the longest string in the dictionary that can be formed by deleting some characters of the given string. If there are more than one possible results, return the longest word with the smallest lexicographical order. If there is no possible result, return the empty string.

**Input**: d = {"ale", "apple", "monkey", "plea"}

S = "abpcplea"

**Output**: "apple"

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | d = {"a", "b", “c"}  S = "abpcplea" | d = {"sun", "moon", “earth"}  S = "masternoon" | d = { "kick", "veer", "sultan"}  S = " khirmock" |
| **Output** | a | moon | kick |

Solution :

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

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

**public class MyClass**

**{**

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

**{**

**List<String> d =Arrays.asList("ale", "apple", "monkey", "plea");**

**String S ="abpcplea";**

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

**System.out.println(ob.findLongestWord(S, d));**

**}**

**}**

**class Solution {**

**static String findLongestWord(String S, List<String> d) {**

**int max = Integer.MIN\_VALUE;**

**Collections.sort(d);**

**String output = "";**

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

**if(check(word,S)){**

**if(word.length()>max){**

**max = word.length();**

**output = word;**

**}**

**}**

**}**

**return output;**

**}**

**static boolean check(String word, String str){**

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

**int l = str.length();**

**int p = 0;**

**for(int i = 0; i < l; i++){**

**if(str.charAt(i)==word.charAt(p)) p++;**

**if(p==n) return true;**

**}**

**return false;**

**}**

**}**

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

Q18) Given a positive integer N, return its corresponding column title as it would appear in an Excel sheet.For N =1 we have column A, for 27 we have AA and so on.

Note: The alphabets are all in uppercase.

**Input**: N = 51

**Output**: AY

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 76 | 98 | 65 |
| **Output** | BX | CT | BM |

Solution :

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

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

**public class MyClass {**

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

**int n = 76;**

**Solution T = new Solution();**

**System.out.println(T.excelColumn(n));**

**}**

**}**

**class Solution {**

**public String excelColumn(int N){**

**if(N <= 26)**

**return ""+(char)('A'+N-1);**

**int quo = N/26;**

**int rem = N%26;**

**if(rem == 0) {**

**rem = 26;**

**quo--;**

**}**

**return excelColumn(quo)+(char)('A'+rem-1);**

**}**

**}**