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

a) Concrete class

b) Virtual Class

**c) Interface**

d) All the above

2) Which of these can be used to fully abstract a class from its implementation?

a) Objects

b) Packages

**c) Interfaces**

d) None of the Mentioned

3) Which of these access specifiers can be used for an interface?

**a) Public**

b) Protected

c) private

d) All of the mentioned

4) Which of these keywords is used by a class to use an interface defined previously?

a) import

b) Import

**c) implements**

d) Implements

5) Which among the following best describes the protected specifier?

a) Members are most secure and can’t be used outside class

b) Members are secure but can be used outside the class

**c) Members are secure as private, but can be inherited**

d) Members are secure like private, but can’t be inherited

6) What will be the output of below statements?

String s = "Java"+1+2+"Quiz"+""+(3+4);

System.out.println(s);

a) Java3Quiz7

**b) Java12Quiz7**

c) Java12Quiz34

d) Java3Quiz34

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

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

a) 0

b) 1

**c) 2**

d) 3

8) Which of the given methods are of Object class?

a) notify(), wait( long msecs ), and synchronized()

b) wait( long msecs ), interrupt(), and notifyAll()

**c) notify(), notifyAll(), and wait()**

d) sleep( long msecs ), wait(), and notify()

9) What is the default relation b/w Thread & Runnable?

a) Thread extends Runnable

**b) Thread implements Runnable**

c) They are not related

d) None of these

10) What is the default priority of the main thread?

a) 1

**b) 5**

c) 10

d) Random value within 1 to 10

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

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

11) 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

12) What is the output of this program?

Note : Output.class file is not in directory pkg.

package pkg;

class output

{

public static void main(String args[])

{

StringBuffer s1 = new StringBuffer("Hello World");

s1.insert(6 , "Good ");

System.out.println(s1);

}

}

**a)Exception in thread "main" java.lang.Error: Unresolved compilation problem**

b)Runtime Exception

c)infinity

d)Not recognize "main"

13) What will be the output of below statements?

String s1 = null;

System.out.print(s1); // line 2

System.out.print(s1.toString()); // line 3

a) nullnull

**b) null followed by NullPointerException**

c) NullPointerException

14) 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.**

15) What will be the output of the following piece of code:

class Person{

public void talk() {}

}

public class Test{

public static void main(String args[]){

Person p = null;

try{

p.talk();

}

catch(NullPointerException e){

System.out.print("There is a NullPointerException. ");

}

catch(Exception e){

System.out.print("There is an Exception. ");

}

System.out.print("Everything went fine. ");

}

}

**a) There is a NullPointerException. Everything went fine.**

b) There is a NullPointerException.

c) There is a NullPointerException. There is an Exception.

d) This code will not compile, because in Java there are no pointers.

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

Q16)Write Java Program to create shapes and find its area using concept of interfaces.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | Length= 10 width=5  Radius=5 | Length= 7 width=3  Radius=8 | Length= 21 width=13  Radius=6 |
| **Output** | Area of rectangle: 50  Area of circle: 78.5 | Area of rectangle: 21  Area of circle: 200.96 | Area of rectangle: 273  Area of circle: 113.04 |

Solution :

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

**// Interface**

**interface ShapeCls {**

**// Abstract method**

**void draw();**

**double area();**

**}**

**class Rect implements ShapeCls {**

**int length, width;**

**// constructor**

**Rect(int length, int width)**

**{**

**this.length = length;**

**this.width = width;**

**}**

**@Override public void draw()**

**{**

**System.out.println("Rectangle has been drawn ");**

**}**

**@Override public double area()**

**{**

**return (double)(length \* width);**

**}**

**}**

**class Cir implements ShapeCls {**

**double pi = 3.14;**

**int radius;**

**// constructor**

**Cir(int radius) { this.radius = radius; }**

**@Override public void draw()**

**{**

**System.out.println("Circle has been drawn ");**

**}**

**@Override public double area()**

**{**

**return (double)((pi \* radius \* radius));**

**}**

**}**

**class JavaProgram {**

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

**{**

**// Creating the Object of Rectangle class**

**// and using shape interface reference.**

**ShapeCls rect = new Rect(2, 3);**

**System.out.println("Area of rectangle: "**

**+ rect.area());**

**// Creating the Objects of circle class**

**ShapeCls cir = new Cir(2);**

**System.out.println("Area of circle: "**

**+ cir.area());**

**}**

**}**

Q17) Given a string S consisting of N characters, the task is to find the minimum number of pairs of characters that are required to be swapped such that no two adjacent characters are the same. If it is not possible to do so, then print “-1”.

**Input:** S = “ABAACD”

**Output**: 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | S = “AABA” | S = “BBAACC” | S = “BCDBC” |
| **Output** | -1 | 2 | 0 |

Solution :

**class JavaProgram**

**{**

**static int ansSwaps ;**

**// Function to check if S contains any pair of adjacent characters that are same**

**static boolean check(char[] S)**

**{**

**// Traverse the String S**

**for (int i = 1; i < S.length; i++) {**

**// If current pair of adjacent**

**// characters are the same**

**if (S[i - 1] == S[i]) {**

**return false;**

**}**

**}**

**// Return true**

**return true;**

**}**

**// Utility function to find the minimum number**

**// of swaps of pair of characters required**

**// to make all pairs of adjacent characters different**

**static void minimumSwaps(char[] S,**

**int swaps, int idx)**

**{**

**// Check if the required String**

**// is formed already**

**if (check(S)) {**

**ansSwaps = Math.min(ansSwaps, swaps);**

**}**

**// Traverse the String S**

**for (int i = idx;**

**i < S.length; i++) {**

**for (int j = i + 1;**

**j < S.length; j++) {**

**// Swap the characters at i**

**// and j position**

**swap(S,i,j);**

**minimumSwaps(S,**

**swaps + 1, i + 1);**

**// Swap for Backtracking**

**// Step**

**S= swap(S,i,j);**

**}**

**}**

**}**

**static char[] swap(char []arr, int i, int j){**

**char temp= arr[i];**

**arr[i]=arr[j];**

**arr[j]=temp;**

**return arr;**

**}**

**// Function to find the minimum number of swaps of pair of characters required**

**// to make all pairs of adjacent characters different**

**static void findMinimumSwaps(char[] S)**

**{**

**// Stores the resultant minimum**

**// number of swaps required**

**ansSwaps = Integer.MAX\_VALUE;**

**// Function call to find the**

**// minimum swaps required**

**minimumSwaps(S, 0,0);**

**// Print the result**

**if (ansSwaps == Integer.MAX\_VALUE)**

**System.out.print("-1");**

**else**

**System.out.print(ansSwaps);**

**}**

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

**{**

**String S = "BCDBC";**

**findMinimumSwaps(S.toCharArray());**

**}**

**}**

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

Q18) Given an array of lowercase strings A[] of size N, determine if the strings can be chained together to form a circle. A string X can be chained together with another string Y if the last character of X is same as first character of Y. If every string of the array can be chained, it will form a circle.

**Input**: N = 4

A[] = { "ab" , "bc", "cd", "da" }

**Output**: 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | N = 3  A[] = { "abc", "bcd", "cdf" } | N = 3  A[] = { "trust", "tea", "art"} | N = 2  A[] = { "alpha", "beta"} |
| **Output** | 0 | 1 | 0 |

Solution :

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

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

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

**public class MyClass**

**{**

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

**{**

**String A[] ={"abc", "bcd", "cdf"};**

**int N =3;**

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

**System.out.println(ob.isCircle(N, A));**

**}**

**}**

**class Solution**

**{**

**static int isCircle(int n, String a[])**

**{**

**ArrayList<ArrayList<Integer>> adj=new ArrayList<ArrayList<Integer>>();**

**for(int i=0;i<26;i++)**

**{**

**adj.add(new ArrayList<Integer>());**

**}**

**int in[]=new int[26];**

**int out[]=new int[26];**

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

**{**

**String temp=a[i];**

**adj.get(temp.charAt(0)-'a').add(temp.charAt(temp.length()-1)-'a');**

**out[temp.charAt(0)-'a']++;**

**in[temp.charAt(temp.length()-1)-'a']++;**

**}**

**for(int i=0;i<26;i++)**

**{**

**if(in[i]!=out[i])**

**return 0;**

**}**

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

**dfs(adj,a[0].charAt(0)-'a',visited);**

**for(int i=0;i<26;i++)**

**{**

**if(visited[i]==false && (in[i]!=0 || out[i]!=0))**

**return 0;**

**}**

**return 1;**

**}**

**static void dfs(ArrayList<ArrayList<Integer>>adj,int ind,boolean visited[])**

**{**

**visited[ind]=true;**

**for(int i:adj.get(ind))**

**{**

**if(visited[i]==false)**

**{**

**dfs(adj,i,visited);**

**}**

**}**

**}**

**}**