**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) In Java, declaring a class abstract is useful

a) To prevent developers from further extending the class.

**b) When it doesn't make sense to have objects of that class.**

c) When default implementations of some methods are not desirable.

d) To force developers to extend the class not to use its capabilities.

2) Which of the following is the correct way of importing an entire package ‘pkg’?

a) import pkg.

b) Import pkg.

**c) import pkg.\***

d) Import pkg.\*

3) Which of the following is an incorrect statement about packages?

a) A package can contain other package within it

b) Java uses file system directories to store packages

c) Package defines a namespace in which classes are stored

**d) A package can be renamed without renaming the directory in which the classes are stored**

4) Which keyword is used to specify the exception thrown by method?

a) catch

**b) throws**

c) finally

d) throw

5) Which of the following blocks execute compulsorily whether exception is caught or not.

**a) finally**

b) catch

c) throws

d) throw

6) Abstract class can have constructors and static methods?

**a) TRUE**

b) FALSE

c) Abstract class can have constructors but can not have static methods.

d) Abstract class can not have constructors but can have static methods.

7) Package names and directory structure are closely related.

**a) True**

b) False

8) Which of these keywords are used to implement synchronization?

a) synchronize

b) syn

c) synch

**d) synchronized**

9) Java String object cannot be changed after creation as it is marked \_\_\_\_\_\_\_\_\_\_

a) Constant

b) transient

**c) final**

d) volatile

10) Which of the following package stores all the standard java classes?

**a) Java**

b) Util

c) Lang

d) Java.packages

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

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

11) What is the output for the below code ?

interface A{

public void printValue();

}

public class Test{

public static void main (String[] args){

A a1 = new A(){

public void printValue(){

System.out.println("A");

}

};

a1.printValue();

}

}

a) Compilation fails due to an error on line 3

**b) A**

c) Compilation fails due to an error on line 8

d) null

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 will be the output of below program?

public class Test {

public static void main(String[] args) {

String x = "abc";

String y = "abc";

x.concat(y);

System.out.print(x);

}

}

**A. abc**

B. abcabc

C. null

14) What is the output of the Java program with PROTECTED access modifer?

package package1;

public class Milk

{

protected int volume = 1000; //Liters

}

package package2;

import package1.Milk;

public class RoseMilk extends Milk

{

public static void main(String[] args)

{

Milk object = new Milk();

System.out.println(object.volume);

}

}

a) 0

b) 1000

c) null

**d) Compiler error**

15) What will be the output of below program?

public class Test {

public static void main(String[] args) {

String s1 = "abc";

String s2 = "abc";

System.out.println("s1 == s2 is:" + s1 == s2);

}

}

**a) false**

b) s1 == s2 is:true

c) s1 == s2 is:false

d) true

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

Q16) Write a java program to throw exception when weight of the product is less than 100

**Input :** 60

**Output:** Caught the exception

Product Invalid

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 90 | 110 | 100 |
| **Output** | Caught the exception  Product Invalid |  |  |

Solution :

**class InvalidProductException extends Exception**

**{**

**public InvalidProductException(String s)**

**{**

**// Call constructor of parent Exception**

**super(s);**

**}**

**}**

**public class JavaProgram**

**{**

**void productCheck(int weight) throws InvalidProductException{**

**if(weight<100){**

**throw new InvalidProductException("Product Invalid");**

**}**

**}**

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

**{**

**JavaProgram obj = new JavaProgram();**

**try**

**{**

**obj.productCheck(60);**

**}**

**catch (InvalidProductException ex)**

**{**

**System.out.println("Caught the exception");**

**System.out.println(ex.getMessage());**

**}**

**}**

**}**

Q17) Write a JAVA program which will generate the threads:

- To display n terms of Fibonacci series.

- To display 1 to n in reverse order.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| Input | Enter the limit for fibonacci; 5  Enter the limit for reverse: 3 | Enter the limit for fibonacci; 2  Enter the limit for reverse: 7 | Enter the limit for fibonacci; 7  Enter the limit for reverse: 2 |
| **Output** | Fibonacci series: 0 1 1 2 3  Reverse is: 3 2 1 | Fibonacci series: 0 1  Reverse is: 7 6 5 4 3 2 1 | Fibonacci series: 0 1 1 2 3 5 8  Reverse is: 2 1 |

Solution :

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

**//thread1**

**class Fibonacci extends Thread**

**{**

**public void run()**

**{**

**try**

**{**

**int a=0, b=1, c=0;**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**System.out.print("Enter the Limit for fabonacci: ");**

**int n = Integer.parseInt(br.readLine());**

**System.out.println("\n=================================");**

**System.out.println("Fibonacci series:");**

**while (n>0)**

**{**

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

**a=b;**

**b=c;**

**c=a+b;**

**n=n-1;**

**}**

**}**

**catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**}**

**}**

**//thread 2**

**class Reverse extends Thread**

**{**

**public void run()**

**{**

**try**

**{**

**BufferedReader br=new BufferedReader(new InputStreamReader(System.in));**

**System.out.print("Enter the Limit for reverse: ");**

**int n = Integer.parseInt(br.readLine());**

**System.out.println("\n=================================");**

**System.out.println("\nReverse is: ");**

**System.out.println("=================================");**

**for (int i=n; i >= 1 ;i-- )**

**{**

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

**}**

**System.out.println("\n=================================\n\n");**

**}**

**catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**}**

**}**

**public class MyClass**

**{**

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

**{**

**try**

**{**

**Fibonacci fib = new Fibonacci();**

**fib.start();**

**fib.sleep(4000);**

**Reverse rev = new Reverse();**

**rev.start();**

**}**

**catch (Exception ex)**

**{**

**ex.printStackTrace();**

**}**

**}**

**}**

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

Q18) Given three strings A, B and C. Write a method that checks whether C is an interleaving of A and B. C is said to be interleaving A and B, if it contains all and only characters of A and B and order of all characters in individual strings is preserved.

Example:

**Input**: strings: "XXXXZY", "XXY", "XXZ"

**Output**: XXXXZY is interleaved of XXY and XXZ

The string XXXXZY can be made by

interleaving XXY and XXZ

String: XXXXZY

String 1: XX Y

String 2: XX Z

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | strings: "XY", "X", "XXY" | strings: "XY", "WZ", "WZXY" | strings: "YX", "X", "XXY" |
| **Output** | XXY is interleaved of XY and X | WZXY is interleaved of XY and WZ | XXY is not interleaved of YX and X |

Solution :

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

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

**class JavaProgram{**

**// The main function that returns true if C is an interleaving of A and B, otherwise false.**

**static boolean isInterleaved(String A, String B,String C)**

**{**

**// Find lengths of the two strings**

**int M = A.length(), N = B.length();**

**// Let us create a 2D table to stores olutions of subproblems. C[i][j] will be true if C[0..i+j-1] is an**

**// interleaving of A[0..i-1] and B[0..j-1].**

**boolean IL[][] = new boolean[M + 1][N + 1];**

**// IL is default initialised by false**

**// C can be an interleaving of A and B only if the sum of lengths of A and B is equal to length of C**

**if ((M + N) != C.length())**

**return false;**

**// Process all characters of A and B**

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

**{**

**for(int j = 0; j <= N; j++)**

**{**

**// Two empty strings have an empty strings as interleaving**

**if (i == 0 && j == 0)**

**IL[i][j] = true;**

**// A is empty**

**else if (i == 0)**

**{**

**if (B.charAt(j - 1) ==**

**C.charAt(j - 1))**

**IL[i][j] = IL[i][j - 1];**

**}**

**// B is empty**

**else if (j == 0)**

**{**

**if (A.charAt(i - 1) ==**

**C.charAt(i - 1))**

**IL[i][j] = IL[i - 1][j];**

**}**

**// Current character of C matches with current character of A, but doesn't match with current character if B**

**else if (A.charAt(i - 1) ==**

**C.charAt(i + j - 1) &&**

**B.charAt(j - 1) !=**

**C.charAt(i + j - 1))**

**IL[i][j] = IL[i - 1][j];**

**// Current character of C matches with current character of B, but doesn't match with current character if A**

**else if (A.charAt(i - 1) !=**

**C.charAt(i + j - 1) &&**

**B.charAt(j - 1) ==**

**C.charAt(i + j - 1))**

**IL[i][j] = IL[i][j - 1];**

**// Current character of C matches with that of both A and B**

**else if (A.charAt(i - 1) ==**

**C.charAt(i + j - 1) &&**

**B.charAt(j - 1) ==**

**C.charAt(i + j - 1))**

**IL[i][j] = (IL[i - 1][j] ||**

**IL[i][j - 1]);**

**}**

**}**

**return IL[M][N];**

**}**

**// Function to run test cases**

**static void test(String A, String B, String C)**

**{**

**if (isInterleaved(A, B, C))**

**System.out.println(C + " is interleaved of " + A + " and " + B);**

**else**

**System.out.println(C + " is not interleaved of " + A + " and " + B);**

**}**

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

**{**

**test("XXY", "XXZ", "XXZXXXY");**

**test("XY", "WZ", "WZXY");**

**test("XY", "X", "XXY");**

**test("YX", "X", "XXY");**

**test("XXY", "XXZ", "XXXXZY");**

**}**

**}**