**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 of these is static variable defined in Collections?

a) EMPTY\_SET

b) EMPTY\_LIST

c) EMPTY\_MAP

**d) All of the mentioned**

2) Which of these method of Array class is used sort an array or its subset?

a) binarysort()

b) bubblesort()

**c) sort()**

d) insert()

3) Stored procedure can be called by using the ????..?

a) CallableStatement

**b) Statement**

c) CalledStatement

d) PreparedStatement

4) What should be the correct order to close the database resource?What should be the correct order to close the database resource?

a) Connection, Statements, and then ResultSet

b) ResultSet, Connection, and then Statements

c) Statements, ResultSet, and then Connection

**d) ResultSet, Statements, and then Connection**

5) Which of these method of FileReader class is used to read characters from a file?

**a) read()**

b) scanf()

c) get()

d) getInteger()

6) Which of these class can be used to implement the input stream that uses a character array as the source?

a) BufferedReader

b) FileReader

**c) CharArrayReader**

d) FileArrayReader

7) Which of these collection class has the ability to grow dynamically?

a) Array

b) Arrays

**c) ArrayList**

d) (None of these)

8) The accuracy and efficiency of a HashMap can be guaranteed with:

a) override equals method

b) override hashCode method

c) (None of these)

**d) (All of these)**

9) Which of these classes can return more than one character to be returned to input stream?

a) BufferedReader

b) Bufferedwriter

**c) PushbachReader**

d) CharArrayReader

10) \_\_\_\_\_\_\_\_is responsible for establishing a connection.

a) Socket

b) ServerSocket

**c) ClientSocket**

d) None of the above

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

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

11) class A{

public void doA(){

B b = new B();

b.dobB();

System.out.print("doA");

}

}

class B{

public void dobB(){

C c = new C();

c.doC();

System.out.print("doB");

}

}

class C{

public void doC(){

if(true)

throw new NullPointerException();

System.out.print("doC");

}

}

public class Test{

public static void main(String args[]){

try{

A a = new A();

a.doA();

}catch(Exception ex){

System.out.print("error");

}

}

}

a) "doCdoBdoA" is printed

b) "doAdoBdoC" is printed

c) "doBdoAerror" is printed

**d) "error" is printed**

12) Determine output of the following code.

interface A { }

class C { }

class D extends C { }

class B extends D implements A { }

public class Test extends Thread{

public static void main(String[] args){

B b = new B();

if (b instanceof A)

System.out.println("b is an instance of A");

if (b instanceof C)

System.out.println("b is an instance of C");

}

}

a) Nothing.

b) b is an instance of A.

c) b is an instance of C.

**d) b is an instance of A followed by b is an instance of C.**

13) The output of the following fraction of code is

public class Test{

public static void main(String args[]){

String s1 = new String("Hello");

String s2 = new String("Hellow");

System.out.println(s1 = s2);

}

}

a) Hello

**b) Hellow**

c) Compilation error

d) Throws an exception

14) Given the following piece of code:

public class School{

public abstract double numberOfStudent();

}

which of the following statements is true?

a) The keywords public and abstract cannot be used together.

b) The method numberOfStudent() in class School must have a body.

c) You must add a return statement in method numberOfStudent().

**d) Class School must be defined abstract.**

15) What is the output of the following program code?

public class Test{

public static void main(String args[]){

try{

int i;

return;

}

catch(Exception e){

System.out.print("inCatchBlock");

}

finally{

System.out.println("inFinallyBlock");

}

}

}

a) inCatchBlock

b) inCatchBlock inFinallyBlock

**c) inFinallyBlock**

d) The program will return without printing anything

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

Q16) Given an array arr[][] consisting of N pairs such that each pair {L, R} represents that ith house can be painted in L number of days before the Rth day, the task is to find the maximum number of house that can be painted continuously.

**Input**: N = 4, paint[ ][ ] = [[1, 19], [2, 2], [4, 17], [1, 1]]

**Output**: 3

Explanation: Maximum of three houses can be painted and order is {4, 3, 1}

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | N = 3, paint[ ][ ] = [[2, 7], [2, 9], [4, 19]] | N = 2, paint[ ][ ] = [[3, 10], [5, 21]] | N = 3, paint[ ][ ] = [[3, 10], [5, 21],[4,6]] |
| **Output** | 3 | 2 | 3 |

Solution :

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

**public class JavaProgram {**

**// Pair class**

**static class Pair {**

**int x, y;**

**Pair(int x, int y)**

**{**

**this.x = x;**

**this.y = y;**

**}**

**}**

**// Comparator to sort by second element**

**static class SortBySecond implements Comparator<Pair> {**

**public int compare(Pair a, Pair b)**

**{**

**return a.y - b.y;**

**}**

**}**

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

**{**

**// Given Input**

**int n = 4;**

**int[][] paint={ { 1, 19 }, { 2, 2 }, { 4, 17 }, { 1, 1 } };**

**// Vector to store the pairs**

**ArrayList<Pair> V = new ArrayList<Pair>();**

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

**// If house can be painted**

**if (paint[i][0] <= paint[i][1]) {**

**V.add(new Pair(paint[i][0], paint[i][1]));**

**}**

**}**

**V.sort(new SortBySecond());**

**if (V.size() == 0) {**

**System.out.println(0);**

**return;**

**}**

**int t = V.get(0).x;**

**PriorityQueue<Pair> pq = new PriorityQueue<>(new SortBySecond());**

**pq.add(V.get(0));**

**// Traversing the vector**

**for (int i = 1; i < V.size(); i++) {**

**t += V.get(i).x;**

**pq.add(V.get(i));**

**if (t <= V.get(i).y) {**

**continue;**

**}**

**else {**

**// Pop the top element**

**Pair temp = pq.poll();**

**t = t - temp.x;**

**}**

**}**

**System.out.println(pq.size());**

**}**

**}**

Q17) Write a generic method to sort array of different data types.

**Input**: {8,54,7,98,43,2}

**Output**: {2,7,8,43,54,98}

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | {“riya”,”gauri”,”abhay”} | {‘k’,’m’,’p’,’e’} | {5.6,9.7,4.3} |
| **Output** | {“abhay”,”gauri”,”riya”} | {‘e’,’k’,’m’,’p’} | {4.3,5.6,9,7} |

Solution:

**public class JavaProgram{**

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

**{**

**Integer[] a = { 10, 6, 54, 41, 5, 1 };**

**Character[] c = { 'v', 'a', 'i', 'b', 'h', 'a', 'v','i' };**

**String[] s = { "pune", "delhi", "satara", "kokan","beed"};**

**System.out.print("Sorted Integer array : ");**

**sort\_generics(a);**

**System.out.print("Sorted Character array : ");**

**sort\_generics(c);**

**System.out.print("Sorted String array : ");**

**sort\_generics(s);**

**}**

**public static <T extends Comparable<T> > void sort\_generics(T[] a)**

**{**

**//Bubble Sort logic**

**for (int i = 0; i < a.length - 1; i++) {**

**for (int j = 0; j < a.length - i - 1; j++) {**

**if (a[j].compareTo(a[j + 1]) > 0) {**

**swap(j, j + 1, a);**

**}**

**}**

**}**

**// Printing the elements after sorted**

**for (T i : a)**

**{**

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

**}**

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

**}**

**public static <T> void swap(int i, int j, T[] a)**

**{**

**T t = a[i];**

**a[i] = a[j];**

**a[j] = t;**

**}**

**}**

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

Q18) Implement different operations on Hashmap. Different types of queries will be provided.

A query can be of four types:

1. a x y (adds an entry with key x and value y to the Hashmap)

2. b x (print value of x if present in the Hashmap else print -1. )

3. c (prints the size of the Hashmap)

4. d x (removes an entry with key x from the Hashmap)

**Input**:

5

a 1 2 a 66 3 b 66 d 1 c

**Output**:

3 1

Explanation :

There are five queries. Queries are performed in this order

1. a 1 2 ---> map has a key 1 with value 2

2. a 66 3 ---> map has a key 66 with value 3

3. b 66 ---> prints the value of key 66 if its present in the map ie 3.

4. d 1 ---> removes an entry from map with key 1

5. c ---> prints the size of the map i.e. 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 3  a 1 66 b 5 c | 2  a 3 98 b 3 | 3  a 3 98 a 4 76 c |
| **Output** | -1 1 | 98 | 2 |

Solution :

**import java.util.Scanner;**

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

**public class MyClass**

**{**

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

**{**

**Scanner sc = new Scanner(System.in);**

**//Declaring and Initializing a HashMap of Integer,Integer Pair**

**HashMap<Integer, Integer> hm = new HashMap<Integer, Integer>();**

**int q = sc.nextInt();**

**while(q>0)**

**{**

**Solution g=new Solution();**

**char c = sc.next().charAt(0);**

**if(c=='a')**

**{**

**int x = sc.nextInt();**

**int y = sc.nextInt();**

**g.add\_Value(hm,x,y);**

**}**

**if(c=='b')//at query type 'b' we find the value of y**

**{**

**int y = sc.nextInt();**

**System.out.print(g.find\_value(hm,y)+" ");**

**}**

**if(c=='c')//at query type 'c' we get size of the HashMap**

**System.out.print(g.getSize(hm)+" ");**

**if(c=='d')//at query type of 'd' we remove the key x**

**{**

**int x = sc.nextInt();**

**g.removeKey(hm,x);**

**}**

**}**

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

**}**

**}**

**class Solution**

**{**

**/\*Inserts an entry with key x and value y in map \*/**

**void add\_Value(HashMap<Integer,Integer> hm, int x, int y)**

**{**

**hm.put(x,y);**

**}**

**/\*Returns the value with key x from the map \*/**

**int find\_value(HashMap<Integer, Integer> hm, int x)**

**{**

**if(hm.containsKey(x))**

**return hm.get(x);**

**else**

**return -1;**

**}**

**/\*Returns the size of the map \*/**

**int getSize(HashMap<Integer, Integer> hm)**

**{**

**return hm.size();**

**}**

**/\*Removes the entry with key x from the map \*/**

**void removeKey(HashMap<Integer, Integer> hm, int x)**

**{**

**hm.remove(x);**

**}**

**}**