**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 types cannot be used to initiate a generic type?

a) Float class

b) Integer class

c) Collections

**d) Primitive Types**

2) Which of these instance cannot be created?

a) Collection instances

b) Integer instance

**c) Generic type instance**

d) Generic class instance

3)List, Set and Queue \_\_\_\_\_\_\_\_\_\_ Collection.

**a) extends**

b) implements

c) both of the above

d) none of the above

4) What is Collection in Java?

**a) A group of objects**

b) A group of interfaces

c) A group of data types

d) A group of classes

5) Which of these methods deletes all the elements from invoking collection?

**a) clear()**

b) reset()

c) delete()

d) refresh()

6) Which of these is an incorrect form of using method max() to obtain a maximum element?

a) max(List c)

b) max(Collection c)

**c) max(Comparator comp)**

d) max(Collection c, Comparator comp)

7) Which of the following method is used to perform DML statements in JDBC?

a) executeResult()

b) executeQuery()

**c) executeUpdate()**

d) execute()

8) How many transaction isolation levels provide the JDBC through the Connection interface?

a) 3

**b) 4**

c) 7

d) 2

9) Which of these is specified by a File object?

a) a file in disk

b) directory path

**c) directory in disk**

d) none of the mentioned

10) Which of these is method for testing whether the specified element is a file or a directory?

a) IsFile()

**b) isFile()**

c) Isfile()

d) isfile()

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

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

11) What is the output of the following program?

import java.util.\*;

public class priorityQueue {

public static void main(String[] args)

{

PriorityQueue<Integer> queue

= new PriorityQueue<>();

queue.add(11);

queue.add(10);

queue.add(22);

queue.add(5);

queue.add(12);

queue.add(2);

while (queue.isEmpty() == false)

System.out.printf("%d ", queue.remove());

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

}

}

a) 11 10 22 5 12 2

b) 2 12 5 22 10 11

**c) 2 5 10 11 12 22**

d) 22 12 11 10 5 2

12) At what line will there be a compilation error?

public class Generics {

public static void main(String[] args) {

B b = new B(); /\* LINE A \*/

C c = b.process(new C()); /\* LINE B \*/

B<C> b2 = new B<C>(); /\* LINE C \*/

C c2 = b2.process(new C()); /\* LINE D \*/

}

}

interface A {

int count();

void show();

}

class B<T extends A> {

T process(T t) {

t.count();

t.show();

return t;

}

}

class C implements A {

public int count() { return 25; }

public void show() { System.out.print("Class C"); }

}

a) LINE A

**b) LINE B**

c) LINE C

d) LINE D

13) What will be the output of the following Java program? (Note: file is made in c drive.)

import java.io.\*;

class files

{

public static void main(String args[])

{

File obj = new File("/java/system");

System.out.print(obj.getParent());

System.out.print(" " + obj.isFile());

}

}

a) java true

b) java false

**c) java false**

d) java true

14) What is the output of the following program?

import java.util.\*;

public class Treeset {

public static void main(String[] args)

{

TreeSet<String> treeSet = new TreeSet<>();

treeSet.add("a");

treeSet.add("b");

treeSet.add("a");

treeSet.add("aba");

for (String temp : treeSet)

System.out.printf(temp + " ");

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

}

}

a) a b a aba

b) a b aba

**c) a aba b**

d) b aba a

15) What will be the output of the following program?

import java.util.\*;

public class SampleDemo {

public static void main(String args[]) {

Sample <Integer> obj = new Sample<Integer>();

obj.push(36);

System.out.println(obj.pop());

}

}

class Sample<E>

{

Stack <E> ob = new Stack <E>();

public void push(E obj) {

ob.push(obj);

}

public E pop() {

E obj = ob.pop();

return obj;

}

}

**a) 36**

b) Some Other Output

c) Compilation Error

d) Runtime Error

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

Q16) Write Java program to merge contents of all the files in a directory

**Input :**

file 1= hello

file 2= world

file 3= java

**Output:**

hello

world

java

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | file 1= java is easy  file 2= I use windows  file 3= pwd is 90 | file 1= birds  file 2= animals  file 3= cities | file 1= hello 23984  file 2= world 8349  file 3= java 839 |
| **Output** | java is easy  I use windows  pwd is 90 | birds  animals  cities | hello 23984  world 8349  java 839 |

Solution :

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

**class JavaProgram {**

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

**{**

**// create instance of directory**

**File dir = new File("F:\\abc");**

**// create object of PrintWriter for output file**

**PrintWriter pw = new PrintWriter("output.txt");**

**// Get list of all the files in form of String Array**

**String[] fileNames = dir.list();**

**// loop for reading the contents of all the files**

**// in the directory GeeksForGeeks**

**for (String fileName : fileNames) {**

**System.out.println("Reading from " + fileName);**

**// create instance of file from Name of**

**// the file stored in string Array**

**File f = new File(dir, fileName);**

**// create object of BufferedReader**

**BufferedReader br = new BufferedReader(new FileReader(f));**

**pw.println("Contents of file " + fileName);**

**// Read from current file**

**String line = br.readLine();**

**while (line != null) {**

**// write to the output file**

**pw.println(line);**

**line = br.readLine();**

**}**

**pw.flush();**

**}**

**System.out.println("Reading from all files" +**

**" in directory " + dir.getName() + " Completed");**

**}**

**}**

Q17) Given a string with repeated characters, we have to insert a star i.e.” \* “ between pair of adjacent identical characters using recursion.

**Input** : aabb

**Output** : a\*ab\*b

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | Pizza | kidding | moon |
| **Output** | piz\*za | kid\*ding | mo\*on |

Solution**:**

**class JavaProgram**

**{**

**static String output="";**

**// Function to insert \* at desired position**

**static void pairStar(String input,**

**int i)**

**{**

**// Append current character**

**output = output + input.charAt(i);**

**// If we reached last character**

**if (i == input.length() - 1)**

**return;**

**// If next character is same, append '\*'**

**if (input.charAt(i) == input.charAt(i+1))**

**output = output + '\*';**

**pairStar(input, i+1);**

**}**

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

**{**

**String input = "pizza";**

**pairStar(input,0);**

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

**}**

**}**

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

Q18) Given a stack with push(), pop(), and empty() operations, The task is to delete the middle element of it without using any additional data structure.

**Input** : Stack[] = [1, 2, 3, 4, 5]

**Output** : Stack[] = [1, 2, 4, 5]

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | Stack[] = [1, 2, 3, 4, 5, 6] | Stack[] = [1, 2, 3] | Stack[] = [11,22,33,44,55] |
| **Output** | Stack[] = [1, 2, 4, 5, 6] | Stack[] = [1, 3] | Stack[] = [11,22,44,55] |

Solution :

**import java.util.Stack;**

**import java.util.Vector;**

**public class JavaProgram {**

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

**Stack<Character> st = new Stack<Character>();//create stack**

**st.push('1');**

**st.push('2');**

**st.push('3');**

**st.push('4');**

**st.push('5');**

**st.push('6');**

**st.push('7');**

**Vector<Character> v = new Vector<Character>();**

**while (!st.empty()) {**

**v.add(st.pop()); //add stack elements to vector**

**}**

**int n = v.size();**

**//push all stack elements expect middle**

**if (n % 2 == 0) {**

**int target = (n / 2);**

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

**if (i == target) continue;**

**st.push(v.get(i));**

**}**

**} else {**

**int target = (int) Math.ceil(n / 2);**

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

**if (i == target) continue;**

**st.push(v.get(i));**

**}**

**}**

**System.out.print("Printing stack after deletion of middle: ");**

**while (!st.empty()) {**

**char p = st.pop();**

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

**}**

**}**

**}**