**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) Thin driver is also known as?

a) Type 3 Driver

b) Type-2 Driver

**c) Type-4 Driver**

d) Type-1 Driver

2) How many statement objects can be created using a Connection?

a) 2

b) 1

c) 3

**d) Multiple**

3) Which of these interface is not a member of java.io package?

a) DataInput

b) ObjectInput

**c) ObjectFilter**

d) FileFilter

4) Which of these class is not related to input and output stream in terms of functioning?

**a) File**

b) Writer

c) InputStream

d) Reader

5) Which of these type of parameters is used for a generic class to return and accept any type of object?

A) K

b) N

**c) T**

d) V

6) Which of the following types of reference cannot be generic?

a) Interface

b) Inner classes

**c) Inner class anonymous**

d) All the answers are true

7) Can elements of a Set be traversed without using Iterator?

**a) Yes**

b) No

8) Collection \_\_\_\_\_\_\_\_\_\_\_\_\_\_

a) inherits the Collections class

**b) inherits the Iterable interface**

c) implements the Serializable interface

d) implements the Traversable interface

9) Which implementation of Iterator can traverse a collection back and forth?

a) Iterator

**b) ListIterator**

c) SetIterator

d) MapIterator

10) The collection is a \_\_\_\_\_\_\_\_\_\_\_\_

**a) framework and interface**

b) framework and class

c) only interface

d) only class

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

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

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

import java)util.\*;

class Collection\_iterators

{

public static void main(String args[])

{

LinkedList list = new LinkedList();

list.add(new Integer(2));

list.add(new Integer(8));

list.add(new Integer(5));

list.add(new Integer(1));

Iterator i = list.iterator();

Collections.reverse(list);

while(i.hasNext())

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

}

}

a) 2 8 5 1

**b) 1 5 8 2**

c) 2

d) 2 1 8 5

12) What will be the output of the following Java program?

import java.util.\*;

public class genericstack

{

Stack stk = new Stack ();

public void push(E obj)

{

stk.push(obj);

}

public E pop()

{

E obj = stk.pop();

return obj;

}

}

class Output

{

public static void main(String args[])

{

genericstack gs = new genericstack();

gs.push("Hello");

System.out.print(gs.pop() + " ");

genericstack gs = new genericstack();

gs.push(36);

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

}

}

a) Error

b) Hello

c) 36

**d) Hello 36**

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.getAbsolutePath());

}

}

a) java

b) system

c) java/system

**d) javasystem**

14) 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.canWrite());

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

}

}

a) true false

b) false true

c) true true

**d) false false**

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

public class UseGenerics {

public static void main(String args[]){

MyGen<Integer> m = new MyGen<Integer>();

m.set("merit");

System.out.println(m.get());

}

}

class MyGen<T>

{

T var;

void set(T var)

{

this.var = var;

}

T get()

{

return var;

}

}

a) merit

**b) 0**

c) Compilation Error

d) Runtime Error

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

Q16) Write a generic java method for finding max of 3 comparable objects.

**Input :** 60 78 95

**Output:** 95

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | 4.5 6.8 1.4 | 110 120 845 | apple pear orange |
| **Output** | 6.8 | 845 | Pear |

Solution :

**public class JavaProgram {**

**public static <T extends Comparable<T>> T maximum(T x, T y, T z) {**

**T max = x; // assume x is initially the largest**

**if(y.compareTo(max) > 0) {**

**max = y; // y is the largest so far**

**}**

**if(z.compareTo(max) > 0) {**

**max = z; // z is the largest now**

**}**

**return max; // returns the largest object**

**}**

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

**System.out.printf("Max of %d, %d and %d is %d ",**

**3, 4, 5, maximum( 3, 4, 5 ));**

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

**System.out.printf("Max of %.1f,%.1f and %.1f is %.1f ",**

**6.6, 8.8, 7.7, maximum( 6.6, 8.8, 7.7 ));**

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

**System.out.printf("Max of %s, %s and %s is %s","pear",**

**"apple", "orange", maximum("pear", "apple", "orange"));**

**}**

**}**

Q17) Given a directed Graph G(V, E) with V vertices and E edges, the task is to check that for all vertices of the given graph, the incoming edges in a vertex is equal to the vertex itself or not.

Adjacency of vertices is given in the form of array of vector.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| Input | V=[(0,1),(1,2),(0,2),(0,3)  ,(1,3),(2,3)] | V=[(0,1),(1,2),(0,2),(0,3)  ,(1,3)] | V=[(0,1),(1,2),(0,2)] |
| **Output** | Yes | No | No |

Solution :

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

**class JavaProgram{**

**// A utility function to add an edge in an directed graph**

**static void add\_edge(Vector<Integer> adj[],**

**int x, int y)**

**{**

**adj[x].add(y);**

**}**

**// Function to check that given graph**

**// in-degree value equal to vertex value**

**static boolean Indegree(Vector<Integer> adj[],**

**int v)**

**{**

**// Create array indeg initialized to zero**

**int indeg[] = new int[v];**

**// Traversing across all vertex to compute in degree value**

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

**{**

**for(int j = 0; j < adj[i].size(); j++)**

**{**

**indeg[adj[i].get(j)]++;**

**}**

**}**

**// Check in degree value equal to vertex value**

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

**{**

**if (i == indeg[i])**

**continue;**

**else**

**return false;**

**}**

**return true;**

**}**

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

**{**

**int v = 4;**

**// To store adjacency list of graph**

**Vector<Integer> []adj = new Vector[v];**

**for(int i = 0; i < adj.length; i++)**

**adj[i] = new Vector<Integer>();**

**add\_edge(adj, 0, 1);**

**add\_edge(adj, 1, 2);**

**add\_edge(adj, 0, 2);**

**add\_edge(adj, 0, 3);**

**add\_edge(adj, 1, 3);**

**add\_edge(adj, 2, 3);**

**if (Indegree(adj, v))**

**System.out.print("Yes");**

**else**

**System.out.print("No");**

**}**

**}**

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

Q18) Write a java program to get primary key value (auto-generated keys) from inserted queries using JDBC.

**Input**: name= john , dept= sales, salary=83409

**Output**: Generated Emp id: 1

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | name= kevin , dept= management, salary=78900 | name= arya , dept= sales, salary=30000 | name= geeta , dept= maintenance, salary=30000 |
| **Output** | Generated Emp id: 2 | Generated Emp id: 3 | Generated Emp id: 4 |

Solution :

**import java.sql.Connection;**

**import java.sql.DriverManager;**

**import java.sql.PreparedStatement;**

**import java.sql.ResultSet;**

**import java.sql.SQLException;**

**import java.sql.Statement;**

**public class JavaProgram {**

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

**Connection con = null;**

**PreparedStatement pstmt = null;**

**ResultSet rs = null;**

**try {**

**// load driver**

**Class.forName("com.mysql.cj.jdbc.Driver");**

**con=DriverManager.getConnection("jdbc:mysql://localhost:3306/details?au" +**

**"toReconnect=true&useSSL=false","root","vaibhu");**

**// create query**

**String query = "insert into emps (name, dept, salary) values (?,?,?)";**

**// create prepare statement**

**pstmt = con.prepareStatement(query,Statement.RETURN\_GENERATED\_KEYS);**

**// set prepare statement values**

**pstmt.setString(1, "John");**

**pstmt.setString(2, "Acc Dept");**

**pstmt.setInt(3, 10000);**

**pstmt.executeUpdate();**

**// use getGeneratedKeys() method for fetching primary keys present at column 1**

**rs = pstmt.getGeneratedKeys();**

**if(rs != null && rs.next()){**

**System.out.println("Generated Emp Id: "+rs.getInt(1));**

**}**

**} catch (ClassNotFoundException e) {**

**e.printStackTrace();**

**} catch (SQLException e) {**

**e.printStackTrace();**

**} finally{**

**try{**

**if(rs != null) rs.close();**

**if(pstmt != null) pstmt.close();**

**if(con != null) con.close();**

**} catch(Exception ex){}**

**}**

**}**

**}**