**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) Identify the isolation level that prevents the dirty in the JDBC Connection class?

a) TRANSACTION\_READABLE\_READ

**b) TRANSACTION\_READ\_COMMITTED**

c) TRANSACTION\_READ\_UNCOMMITTED

d) TRANSACTION\_NONE

2) What does setAutoCommit(false) do?

a) It will not commit transactions automatically after each query.

**b) It explicitly commits the transaction.**

c) It never commits the transactions.

d) It does not commit transaction automatically after each query.

3) Which of these is a method to clear all the data present in output buffers?

a) clear()

**b) flush()**

c) fflush()

d) close()

4) Which of these stream contains the classes which can work on character stream?

a) InputStream

b) OutputStream

**c) Character Stream**

d) All of the mentioned

5) Generics does not work with?

a) Set

b) List

c) Tree

**d) Array**

6) Which of these instances cannot be created?

a) Integer Instance

b) Generic Class Instance

**c) Generic Type Instance**

d) Collection Instances

7) Which of the following is legacy class in java?

a) Arraylist

b) Vector

c) Hashtable

**d) Only B & C**

8) Which among the following Sets maintains insertion order?

a) HashSet

b) TreeSet

**c) LinkedHashSet**

d) Both B & C

9) Which of these methods can randomize all elements in a list?

a) rand()

**b) shuffle()**

c) randomize()

d) ambiguous()

10) Which of these class is used to read characters in a file?

**a) FileReader**

b) FileWriter

c) FileInputStream

d) InputStreamReader

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

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

11) Which result set generally does not show changes to the underlying database that are made while it is open. The membership, order, and column values of rows are typically fixed when the result set is created?

a) TYPE\_FORWARD\_ONLY

**b) TYPE\_SCROLL\_INSENSITIVE**

c) TYPE\_SCROLL\_SENSITIVE

d) ALL MENTIONED ABOVE

12) Which of the given choices is a possible output?

import java.util.\*;

public class hashSet {

public static void main(String[] args)

{

HashSet<String> hashSet = new HashSet<>();

hashSet.add("apple");

hashSet.add("pineapple");

hashSet.add("mango");

hashSet.add("grapes");

hashSet.add("pineapple");

System.out.println(hashSet);

}

}

**a) [apple, pineapple, mango, grapes]**

b) [apple,mango, grapes]

c) []

13) Given:

Object [] myObjects = {

new Integer(12),

new String("foo"),

new Integer(5),

new Boolean(true)

};

Arrays.sort(myObjects);

for(int i=0; i<myObjects.length; i++) {

System.out.print(myObjects[i].toString());

System.out.print(" ");

}

What is the result?

a) Compilation fails due to an error in line 23.

b) Compilation fails due to an error in line 29.

**c) A ClassCastException occurs in line 29.**

d) A ClassCastException occurs in line 31.

14) Which is the correct absolute path of a file in Java?

**a) C:\Program Files\Java\jdk1.8.0\_131\bin\file\_name.txt**

b) C:\Program Files\Java\file\_name.txt

c) C:\Program Files\Java\jdk1.8.0\_131\file\_name.txt

d) C:\Program Files\Java\jdk1.8.0\_131\bin\File Handling\file\_name.txt

15) What will be the output of following Java code?

import java.util.Hashtable;

public class HashTableClass {

int hashcode;

HashTableClass(int hashcode) {

this.hashcode = hashcode;

}

public int hashCode() {

return hashcode;

}

public String toString() {

return hashcode + " ";

}

public static void main(String[] args) {

Hashtable ht = new Hashtable();

ht.put(new HashTableClass(10), "Java");

ht.put(new HashTableClass(3), "C");

ht.put(new HashTableClass(4), "C++");

ht.put(new HashTableClass(5), "Ruby");

ht.put(new HashTableClass(6), "null");

System.out.println(ht);

}

}

a) {10 =Java, 3 =C, 4 =C++, 6 =null, 5 =Ruby}

**b) {10 =Java, 6 =null, 5 =Ruby, 4 =C++, 3 =C}**

c) {3 =C, 4 =C++, 5 =Ruby, 6 =null, 10 =Java}

d) None of these

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

Q16) Given a source stack, copy the contents of the source stack to destination stack maintaining the same order without using extra space.

**Input** : Source:-

|3|

|2|

|1|

**Output** : Destination:-

|3|

|2|

|1|

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | Source= 34 56 76 9 | Source= 3 4 6 8 | Source= 4 3 2 1 |
| **Output** | Destination= 34 56 76 9 | Destination= 3 4 6 8 | Destination= 4 3 2 1 |

Solution :

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

**// Define a class for Stack**

**class Stack {**

**private List<Integer> stack = new ArrayList<>();**

**public void push(int value) {**

**stack.add(value);**

**}**

**public int pop() {**

**int topVal = stack.get(stack.size() - 1);**

**stack.remove(stack.size() - 1);**

**return topVal;**

**}**

**public int length() {**

**return stack.size();**

**}**

**public void display() {**

**for (int i = stack.size() - 1; i >= 0; i--) {**

**System.out.println(stack.get(i));**

**}**

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

**}**

**}**

**public class JavaProgram {**

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

**Stack source = new Stack(); // Source Stack**

**Stack dest = new Stack(); // Destination Stack**

**source.push(1);**

**source.push(2);**

**source.push(3);**

**System.out.println("Source Stack:");**

**source.display();**

**int count = 0;**

**// Reverse the order of the values in source stack**

**while (count != source.length() - 1) {**

**int topVal = source.pop();**

**while (count != source.length()) {**

**dest.push(source.pop());**

**}**

**source.push(topVal);**

**while (dest.length() != 0) {**

**source.push(dest.pop());**

**}**

**count += 1;**

**}**

**// Pop the values from source and push into destination stack**

**while (source.length() != 0) {**

**dest.push(source.pop());**

**}**

**System.out.println("Destination Stack:");**

**dest.display();**

**}**

**}**

Q17) write a generic function print class of the passed object as parameter.

**Input**: 23

**Output**: java.lang.Integer = 23

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | “JAVAProgramming” | 1.0 | ‘V’ |
| **Output** | java.lang.String = JAVAProgramming | java.lang.Double = 1.0 | java.lang.Character=V |

Solution :

**class JavaProgram{**

**static <T> void genericDisplay(T element)**

**{**

**System.out.println(element.getClass().getName() + " = " + element);**

**}**

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

**{**

**// Calling generic method with Integer argument**

**genericDisplay(11);**

**// Calling generic method with String argument**

**genericDisplay("JavaProgram");**

**// Calling generic method with double argument**

**genericDisplay(1.0);**

**}**

**}**

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

Q18) Given an expression string x. Examine whether the pairs and the orders of {,},(,),[,] are correct in expression.

**Input**: [()]{}{[()()]()}

**Output**: Input string [()]{}{[()()]()} is balanced.

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Test Case 1** | **Test Case 2** | **Test Case 3** |
| **Input** | [()]{}{[]()} | [()]{}{[] | []{}() |
| **Output** | Input string [()]{}{[]()} is balanced. | Input string [()]{}{[] is not balanced. | Input string []{}() is balanced. |

Solution :

**import java.util.Stack;**

**import java.util.Scanner;**

**public class JavaProgram{**

**public static boolean balancedParenthesis(String inputStr) {**

**Stack stack = new Stack();**

**char[] charArray = inputStr.toCharArray();**

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

**char current = charArray[i];**

**if (current == '{' || current == '[' || current == '(') {**

**stack.push(current);**

**continue;**

**}**

**if (stack.isEmpty()) {**

**return false;**

**}**

**// use switch statement to pop element from stack and if it is '(', '[' or '{', return false**

**char popChar;**

**switch (current) {**

**case ')':**

**popChar = (char) stack.pop();**

**if (popChar == '{' || popChar == '[')**

**return false;**

**break;**

**case '}':**

**popChar = (char) stack.pop();**

**if (popChar == '(' || popChar == '[')**

**return false;**

**break;**

**case ']':**

**popChar = (char) stack.pop();**

**if (popChar == '(' || popChar == '{')**

**return false;**

**break;**

**}**

**}**

**return (stack.isEmpty());**

**}**

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

**String inputStr;**

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

**System.out.println("Enter input string to check:");**

**inputStr = sc.nextLine();**

**if (balancedParenthesis(inputStr))**

**System.out.println("Input string "+inputStr+" is balanced.");**

**else**

**System.out.println("Input string "+inputStr+" is not balanced.");**

**}**

**}**