Q1. Find predicted output of given program which is used to add and delete elements using Stack.

```
class mcq
{
        public static void main(string args[])
                {
                Stack obj = new Stack();
                obj.push(new Integer(3));
                obj.push(new Integer(2));
                obj.pop();
                obj.push(new Integer(5));
                System.out.println(obj);
        }
}
```

a) **[3, 5]**
b) [3, 2]
c) [3, 2, 5]
d) [3, 5, 2]

Q2. What is the output of the program?

```
interface calculate {
   int VAR = 0;
   void cal(int item);
}
class display implements calculate {
   int x;
   public void cal(int item) {
      if (item < 2)
         x = VAR;
      else
         x = item * item;
   }
}
class mcq {
   public static void main(String[] args) {
      display[] arr = new display[3];
      for (int i = 0; i < 3; i++) arr[i] = new display();
      arr[0].cal(0);
      arr[1].cal(1);
      arr[2].cal(2);
      System.out.print(arr[0].x + " " + arr[1].x + " " + arr[2].x);
   }
}
```

a) 0 1 2
b) 0 2 4
c) **0 0 4**
d) 0 1 4

- Q3. What are generic methods?

- A. Generic methods are methods that take void parameters
- B. Generic methods are the methods defined in a generic class
- C. Generic methods are the methods that extend generic class methods
- D. Generic methods are methods that introduce their own type parameters

Q4. What will be the output of the following program?

```
public class mcq
{
        public static void main(String[] args) {
                char ch = 65;
                System.out.println("character = " + ch);
        }
}
```

a) character =65
b) Compilation Error
c) character =A
d) character =”65”

Q5. What will be the output of the following program?

```
public class mcq {
        public static void main(String[] args) {
                Super s = new Sub();
                s.foo();
        }
}
class Super {
        void foo() {
                System.out.println("Class Super");
        }
}
class Sub extends Super {
        static void foo() {
                System.out.println("Class Sub");
        }
}
```

a) Class Super
b) Class Sub
c) foo() in Sub cannot override foo() in Super
d) The statement “Super s = new Sub();” is illegal.

Q6. Find predicted output of given program which is used to add and delete elements using Stack.

```
class mcq
{
        public static void main(string args[])
                {
                        Stack obj = new Stack();
                        obj.push(new Integer(3));
                        obj.push(new Integer(2));
                        obj.pop();
                        obj.push(new Integer(5));
                        System.out.println(obj);
            }
}
```
a) **[3, 5]**
b) [3, 2]
c) [3, 2, 5]
d) [3, 5, 2]

Q7. What is the output of the program?
```
interface calculate {
    int VAR = 0;
    void cal(int item);
}
class display implements calculate {
    int x;
    public void cal(int item) {
        if (item < 2)
            x = VAR;
        else
            x = item * item;
    }
}
class mcq {
    public static void main(String[] args) {
        display[] arr = new display[3];
        for (int i = 0; i < 3; i++) arr[i] = new display();
        arr[0].cal(0);
        arr[1].cal(1);
        arr[2].cal(2);
        System.out.print(arr[0].x + " " + arr[1].x + " " + arr[2].x);
    }
}
```
a) 0 1 2
b) 0 2 4
c) **0 0 4**
d) 0 1 4

- Q8. 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.println(gs.pop());
}
}
```

- A. H
- B. Hello
- C. Runtime Error
- D. Compilation Error

Q9. What will be the output of the following program?

```
public class mcq
{
        public static void main(String[] args) {
                char ch = 65;
                System.out.println("character = " + ch);
        }
}
```

e) character =65
f) Compilation Error
g) character =A
h) character =”65”

Q10. What will be the output of the following program?

```
public class mcq {
        public static void main(String[] args) {
                Super s = new Sub();
                s.foo();
        }
}
class Super {
        void foo() {
                System.out.println("Class Super");
        }
}
class Sub extends Super {
        static void foo() {
                System.out.println("Class Sub");
        }
}
```

e) Class Super
f) Class Sub
g) foo() in Sub cannot override foo() in Super
h) The statement “Super s = new Sub();” is illegal.

Q11. What will be the output of the following code?

```
import java.util.*;
public class Main
{
  public static void main(String[] args)
  {
        List<String> names = new LinkedList<>();
        names.add("Raman");
        names.add("Shyam");
        names.add("Rahul");
        names.add("Rohit");
        names.add("Ramit");
        ListIterator<String> it = names.listIterator(3);
        while(it.hasPrevious()){
        System.out.print(it.previous()+" ");
        }
  }
}
```

a) Raman Shyam Rahul
b) Shyam Rahul Rohit
c) Rahul Rohit Ramit
d) Rahul Shyam Raman

Q12. What will be the output of the following java code?

```
public class Main
{
        public static void main(String[ ] args)
        {
          StringBuffer[ ] stringBuffers = new StringBuffer[10];
          for (int i = 0; i < stringBuffers.length; i ++)
                        stringBuffers [i].append("index " + i);
          System.out.println(stringBuffers);
        }
}
```

a) Compile Time Error
b) Null Pointer Exception
c) index0index
d) 0123456789

Q13. What will be the output of the following java code?

```
class A {
        public void aMethod() {
                System.out.println("a Method from A");
        }
}
class B  extends A {
        public void aMethod() {
                System.out.println("a Method from B");
        }
}
public class Main {
        public static void main(String ar[]) {
                A a = new B();
                a.aMethod();
        }
}
```

a) a Method from A
b) a Method from B
c) Compilation Error
d) Runtime Exception

Q14. What will be the output of the following code?

```
import java.util.*;
   class genericstack <E>
   {
      Stack <E> stk = new Stack <E>();
         public void push(E obj)
   {
      stk.push(obj);
         }
         public E pop()
   {
      E obj = stk.pop();
            return obj;
         }   }
   public class Main
   {
      public static void main(String args[])
      {
         genericstack <String> gs = new genericstack<String>();
         gs.push("Hello");
         gs.push(25);
```

```
            System.out.println(gs.pop());
        }
    }
```
a) Compile Time Error: incompatible types
b) Compile Time Error: Stack class does not exist
c) Compile Time Error: E is not defined
d) No Compile Time Error

Q15. Which IO stream in Java is used to read or write primitive data types?
a) BufferedDataInputStream and BufferedDataOutputStream
b) DataInputStream and DataOutputStream
c) BufferedInputStream and BufferedOutputStream
d) FilteredDataInputStream and FilteredDataOutputStream

Q16. Find predicted output of given program which is used to add and delete elements using Stack.
```
class mcq
{
            public static void main(string args[])
                        {
                        Stack obj = new Stack();
                        obj.push(new Integer(4);
                        obj.push(new Integer(2));
                        obj.pop();
                        obj.push(new Integer(5));
                        System.out.println(obj);
            }
}
```
a) **[4, 5]**
b) [3, 2]
c) [3, 2, 5]
d) [3, 5, 2]

Q17. What is the output of the program?
```
interface calculate {
    int VAR = 0;
    void cal(int item);
}
class display implements calculate {
    int x;
    public void cal(int item) {
        if (item < 2)
            x = VAR;
        else
            x = item * item;
    }
}
class mcq {
    public static void main(String[] args) {
        display[] arr = new display[3];
        for (int i = 0; i < 3; i++) arr[i] = new display();
        arr[0].cal(0);
        arr[1].cal(1);
        arr[2].cal(2);
        System.out.print(arr[0].x + " " + arr[1].x + " " + arr[2].x);
    }
}
```

a) 0 1 2
b) 0 2 4
c) **0 0 4**
d) 0 1 4

Q18 Which of the following is incorrect statement regarding the use of generics and parameterized types in Java?
A. Generics provide type safety by shifting more type checking responsibilities to the compiler

B. Generics and parameterized types eliminate the need for down casts when using Java Collections

C. When designing your own collections class (say, a linked list), generics and parameterized types allow you to achieve type safety with just a single class definition as opposed to defining multiple classes

D. All of the mentioned

Q19. Which of the following allows us to call generic methods as a normal method?
A. Interface

B. Inner class

C. Type Interface

D. All of the mentioned

Q20. What will be the output of the following program?

```
public class mcq {
        public static void main(String[] args) {
                Super s = new Sub();
                s.foo();
        }
}
class Super {
        void foo() {
                System.out.println("Class Super");
        }
}
class Sub extends Super {
        static void foo() {
                System.out.println("Class Sub");
        }
}
```
a) Class Super
b) Class Sub
c) foo() in Sub cannot override foo() in Super
d) The statement “Super s = new Sub();” is illegal.

Q21. Find predicted output of given program which is used to add and delete elements using Stack.

```
class mcq
{
        public static void main(string args[])
                {
                Stack obj = new Stack();
                obj.push(new Integer(5));
                obj.push(new Integer(2));
                obj.pop();
                obj.push(new Integer(5));
                System.out.println(obj);
        }
}
```
a) **[5, 5]**
b) [3, 2]
c) [3, 2, 5]
d) [3, 5, 2]

Q22. What is the output of the program?

```
interface calculate {
   int VAR = 0;
   void cal(int item);
}
class display implements calculate {
   int x;
   public void cal(int item) {
      if (item < 2)
         x = VAR;
      else
         x = item * item;
   }
}
class mcq {
   public static void main(String[] args) {
      display[] arr = new display[3];
      for (int i = 0; i < 3; i++) arr[i] = new display();
      arr[0].cal(0);
      arr[1].cal(1);
      arr[2].cal(2);
      System.out.print(arr[0].x + " " + arr[1].x + " " + arr[2].x);
```

```
    }
}
```

a) 0 1 2
b) 0 2 4
c) **0 0 4**
d) 0 1 4

-
-

Q23. 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.println(gs.pop());
}
}
```

A. H

B. Hello

C. Runtime Error

D. Compilation Error

Q24. What will be the output of the following program?

```
public class mcq
{
        public static void main(String[] args) {
                char ch = 65;
                System.out.println("character = " + ch);
        }
}
```

a) character =65
b) Compilation Error
c) character =A
d) character =”65”

Q25. What will be the output of the following program?

```
public class mcq {
          public static void main(String[] args) {
                    Super s = new Sub();
                    s.foo();
          }
}
class Super {
          void foo() {
                    System.out.println("Class Super");
          }
}
class Sub extends Super {
          static void foo() {
                    System.out.println("Class Sub");
          }
}
```

e) Class Super
f) Class Sub
g) foo() in Sub cannot override foo() in Super
h) The statement “Super s = new Sub();” is illegal.

Q26. What will be the output of the following code?

```
import java.util.*;
public class Main
{
  public static void main(String[] args)
  {
          List<String> names = new LinkedList<>();
          names.add("Raman");
          names.add("Shyam");
          names.add("Rahul");
          names.add("Rohit");
          names.add("Ramit");
          ListIterator<String> it = names.listIterator(3);
          while(it.hasPrevious()){
          System.out.print(it.previous()+" ");
          }
  }
}
```

a) Raman Shyam Rahul
b) Shyam Rahul Rohit
c) Rahul Rohit Ramit
d) Rahul Shyam Raman

Q27. What will be the output of the following java code?

```
public class Main
{
        public static void main(String[ ] args)
        {
          StringBuffer[ ] stringBuffers = new StringBuffer[10];
          for (int i = 0; i < stringBuffers.length; i ++)
                        stringBuffers [i].append("index " + i);
          System.out.println(stringBuffers);
        }
}
```

a) Compile Time Error
b) Null Pointer Exception
c) index0index
d) 0123456789

Q28. What will be the output of the following java code?

```
class A {
        public void aMethod() {
                System.out.println("a Method from A");
        }
}
class B  extends A {
        public void aMethod() {
                System.out.println("a Method from B");
        }
}
public class Main {
        public static void main(String ar[]) {
                A a = new B();
                a.aMethod();
        }
}
```

a) a Method from A
b) a Method from B
c) Compilation Error
d) Runtime Exception

Q29. What will be the output of the following code?

```
import java.util.*;
   class genericstack <E>
   {
      Stack <E> stk = new Stack <E>();
```

```
        public void push(E obj)
    {
      stk.push(obj);
        }
        public E pop()
    {
      E obj = stk.pop();
          return obj;
        }   }
   public class Main
   {
      public static void main(String args[])
      {
          genericstack <String> gs = new genericstack<String>();
          gs.push("Hello");
          gs.push(25);
          System.out.println(gs.pop());
      }
   }
```

a) Compile Time Error: incompatible types
b) Compile Time Error: Stack class does not exist
c) Compile Time Error: E is not defined
d) No Compile Time Error

Q30. Which IO stream in Java is used to read or write primitive data types?
a) BufferedDataInputStream and BufferedDataOutputStream
b) DataInputStream and DataOutputStream
c) BufferedInputStream and BufferedOutputStream
d) FilteredDataInputStream and FilteredDataOutputStream

Q31 The InputStream class defines methods for performing input functions such as
i) reading bytes ii) closing streams
iii) skipping ahead in a stream iv) flushing streams

A) ii, iii and iv only
B) i, ii and iii only
C) i, iii and iv only
D) All i, ii, iii and iv

Q32 . Which of the following method(s) not included in InputStream class.
A) available( )
B) reset( )
C) flush( )
D) close( )

Q33. Which exception is thrown by the read( ) method of InputStream class.
A) Exception
B) IOException

C) ReadException
D) File Not Found Exception

Q34. Which of these packages contains all the collection classes?

a) java.awt
b) java.net
c) java.util
d) java.lang

Q35. What is Collection in Java?

a. A group of objects
b. A group of interfaces
c. A group of classes
d. None of the above

Q36. Which of these collection classes has the ability to scale dynamically?

a) Array
b) Arrays
c) ArrayList
d) All the answers are true

Q37. The interface Comparable contains the method ___________

a) toCompare
b) compare
c) compareTo
d) compareWith

Q38. Which of the following Sets maintains the insertion order?

a) HashSet
b) TreeSet
c) LinkedHashSet
d) All the answers are true

Q39. Default capacity of an ArrayList is ___

a) 12
b) 10
c) 8
d) 16

Q40. How many threads can a process contain?

a) 1
b) 2
c) Multiple

d) None

Q41.
```
int values[ ] = {1,2,3,4,5,6,7,8,9,10};
    for(int i=0;i< Y; ++i)
    System.out.println(values[i]);
```

Find the value of value[i]?

a. 10

b. 11

c. 15

d. None of the above

Q42

```
public class Hello {
        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.println( queue.remove());
        }
}
```

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

Q43
```
public class Hello {
        public static void main(String[] args) {
                TreeSet<String> treeSet = new TreeSet<>();
                treeSet.add("Sky");
                treeSet.add("Is");
                treeSet.add("Blue");
                treeSet.add("SkyIsBlue");
                for (String temp : treeSet)
                System.out.print(temp + " ");
                System.out.println("\n");
                }
        }
```

A. Blue Is Sky SkyIsBlue
B. Is Sky Blue SkyISblue
C. Blue Sky Is
D. SkyIsBlue

Q44. What will be the output of the following code?

```
public class hashSet {
    public static void main(String[] args)
    {
        HashSet<String> hashSet = new HashSet<>();
        hashSet.add("Hello");
        hashSet.add("For");
        hashSet.add("Hello");
        hashSet.add("Good");

        System.out.println(hashSet);
    }
}
```

a) Hello For Good

b) Hello For Hello Good

c) Good Hello For Hello

d) Hello Good Hello For

Q45.

```
import java.util.ArrayList;
class Demo {
public void show()
  {
    ArrayList<Integer> list = new ArrayList<Integer>();
    list.add(4);
    list.add(7);
    list.add(1);
    for (int number : list) {
      System.out.print(number + " ");
    }
    }
} public class Main {
public static void main(String[] args)
    {
        Demo demo = new Demo();
        demo.show();
    }
}
```

A. Compilation Error
B. 4 7 1
C. 1 4 7
D. None

Q46 The OutputStreams includes methods that are designed to perform the following tasks.

i) closing streams ii) flushing streams
iii) reading bytes iv) writing bytes

A) ii, iii and iv only
B) i, ii and iii only
C) i, ii and iv only
D) All i, ii, iii and iv

Q47 . Which of the following methods not included in OutputStream class.

A) write( )
B) skip( )
C) close( )
D) flush( )

Q48. The ……………………… method of the BufferedReader class is used for reading lines of text from the console, the file or other input streams.

A) read( )
B) read(byte[]b)
C) readLine( )
D) readByte( )

Q49. Which of these classes is not part of the Collection framework in Java?

a) Queue
b) Stack
c) ArrayList
d) Map

Q50. Which interface does not allow duplicates elements?

a) Set
b) List
c) Map
d) All the answers are true

Q51. HashMap allows _____________

a) null values
b) null key

c) All the answers are true
d) None of the above

Q52. Which of the following interfaces maintains the order in which the elements are inserted?
a) Set
b) List
c) Map
d) All the answers are true

Q53. Which class stores items as a key-value pair?
a) Arraylist
b) LinkedHashSet
c) TreeMap
d) TreeSet

Q54. Threads are
a) light weight process
b) heavyweight process
c) both
d) none

Q55. In character stream I/O, a single read/write operation performs _____.
a) Two bytes read/write at a time.
b) Eight bytes read/write at a time.
c) One byte read/write at a time.
d) Five bytes read/ write at a time.

MCQ (2 Marks)

Q56. What is the output of this program?

```
import java.util.LinkedList;

class Demo {

public void show()

 {

    LinkedList<String> list = new LinkedList<String>();

    list.add("Element1"); // line 6

    list.add("Element2");

    System.out.print(list.getFirst()); // line 8
    }
} public class Main {
public static void main(String[] args)
    {
        Demo demo = new Demo();
        demo.show();
    }
}
```

A. Element1

B. Compilation Error at line 8

C. Runtime Error

D. Element2

Q57. import java.util.ArrayList;

```
class Demo {
public void show()
    {
        ArrayList<String> list = new ArrayList<String>();
        list.add("GeeksForGeeks_one"); // line 6
        list.add("GeeksForGeeks_two");
        System.out.print(list.getFirst()); // line 8
    }
} public class Main {
public static void main(String[] args)
    {
        Demo demo = new Demo();
        demo.show();
    }
}
```

A. GeeksForGeeks_one

B. Compilation Error

C. Runtime Error

Q58

```
import java.util.*;
class Demo {
public void show()
   {
      List<Integer> list = new LinkedList<Integer>();
      list.add(1);
      list.add(4);
      list.add(7);
      list.add(5);
      Collections.sort(list); // line 8
      System.out.println(list);
   }
} public class Main {
public static void main(String[] args)
   {
      Demo demo = new Demo();
      demo.show();
   }
}
```

A. Compilation Error at line 9
B. [1, 4, 5, 7]
C. [1, 4, 7, 5]
D. Runtime Error

Q59. What will be the output of the following code?

```
import java.util.*;

public class linkedList {
   public static void main(String[] args)
   {
      List<String> list1 = new LinkedList<>();
      list1.add("Hello");
      list1.add("For");
      list1.add("Hello ");
      list1.add("ABC");
      list1.add("HelloForHello ");

      List<String> list2 = new LinkedList<>();
      list2.add("Hello");

      list1.removeAll(list2);
```

```
            for (String temp : list1)
                System.out.print(temp + " ");

            System.out.println();
        }
    }
```

a) For ABC HelloForHello
b) Hello ABC Hello
c) ABC Hello ABC
d) Hello Helo

Q60.

```
import java.util.*;

public class Hello {
public static void main(String[] args) {
LinkedHashSet<Integer> set = new LinkedHashSet<>();
  set.add(1);
  set.add(2);
  set.add(2);
  set.add(4);

  for (int temp : set)
  System.out.print(temp + " ");

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

   }
}
```

a) 1 2 4

b) 1 2 2 4

c) 4 2 2 1

d) Compile-time error

Q61. Which of the following is true about Generics in Java?
a) Generics is used to create objects of a class.
b) Generics helps in writing type-safe code.
c) Generics is used to convert data types.
d) Generics can only be used with primitive data types.

Q62. Which of the following is an example of Generics in Java?

a) ArrayList
b) Integer
c) Character
d) Float

Q63.Which of the following is a syntax for defining a generic method in Java?
a) <T> void myGenericMethod(T item) {}
b) void myGenericMethod<T>(T item) {}
c) void myGenericMethod(T item) {}
d) <T: E> void myGenericMethod(T item) {}

Q64.Which of the following is a syntax for defining a Generic Interface in Java?
a) interface MyGenericInterface<T> {}
b) interface MyGenericInterface<T, E> {}
c) interface MyGenericInterface(T) {}
d) interface MyGenericInterface<T: E> {}

Q65. What is an Iterator in Java?
a) An interface that represents a group of objects
b) An interface that represents a single object
c) An interface that represents a primitive data type
d) An interface that allows iterating over a collection

Q66 .Which of the following is a syntax for implementing the Comparable interface in Java?
a) class MyClass<T extends Comparable<T>> implements Comparable<MyClass<T>> {}
b) class MyClass<T> implements Comparable<MyClass<T>> {}
c) class MyClass<T extends Comparator<T>> implements Comparable<MyClass<T>> {}
d) class MyClass<T extends Comparable> implements Comparable<MyClass<T>> {}

Q67. What is Comparable in Java?
a) An interface that allows sorting of objects based on their natural order
b) An interface that allows sorting of objects based on a custom order
c) A class that allows sorting of objects based on their natural order
d) A class that allows sorting of objects based on a custom o

Q68 What is an IO Stream in Java?
a) A continuous flow of data
b) A physical layer that links to a stream
c) A class that reads data from a source
d) A class that writes data to a destination

Q69. Which of the following is a syntax for creating a Stack in Java?
a) Stack myStack = new Stack();
b) Stack<String> myStack = new Stack<String>();
c) Stack myStack = new Stack<String>();
d) Stack<String> myStack = new Stack();

Q70. What is the purpose of JDBC in Java?
a) To provide a standard abstraction for Java applications to communicate with various databases
b) To provide a standard abstraction for Java applications to communicate with various internet protocols
c) To provide a standard abstraction for Java applications to communicate with various file systems
d) To provide a standard abstraction for Java applications to communicate with various operating systems

Q71. What is the output of the following code snippet?

```
List<Integer> myList = new ArrayList<>();
```

```
myList.add(1);
myList.add(2);
myList.add(3);
myList.remove(2);
System.out.println(myList);
```

a) [1, 2, 3]
b) [1, 2]
c) [1, 3]
d) [2, 3]

Q72. What is the output of the following code snippet?

```
Set<Integer> mySet = new HashSet<>();
mySet.add(1);
mySet.add(2);
mySet.add(3);
mySet.add(1);
System.out.println(mySet.size());
```

a) 1
b) 2
c) 3
d) 4

Q73.What will be the output of the following code snippet?

```
Map<String, Integer> map = new HashMap<>();
map.put("John", 25);
map.put("Mary", 30);
map.put("Peter", 35);
System.out.println(map.get("Mary"));
```

a) 25
b) 30
c) 35
d) null

Q74. What will be the output of the following Java code?

```
class GenericsExample {
   public static void main(String[] args) {
      GenericClass<Integer> gc1 = new GenericClass<>(10);
      GenericClass<String> gc2 = new GenericClass<>("Hello");

      System.out.println(gc1.getData());
      System.out.println(gc2.getData());
   }
}

class GenericClass<T> {
   private T data;

   public GenericClass(T data) {
      this.data = data;
   }

   public T getData() {
      return data;
   }
}
```

a) 10 Hello
b) Hello 10
c) Error
d) None of the above

Q75.What will be the output of the following Java code?

```
import java.util.*;

public class CollectionsExample {
   public static void main(String[] args) {
      List<Integer> list1 = new ArrayList<>(Arrays.asList(1, 2, 3));
      List<Integer> list2 = new ArrayList<>(Arrays.asList(4, 5, 6));

      Collections.copy(list1, list2);

      System.out.println(list1);
   }
}
```

a) [1, 2, 3]
b) [4, 5, 6]
c) [4, 5, 6, null, null, null]

d) Error

Q76.What are generics in Java?
a) A type of data structure
b) A type of programming language
c) A type of algorithm
d) A way to create classes that work with different data types

Q77.What is a generic class in Java?
a) A class that can work with different types of objects
b) A class that can only work with one type of object
c) A class that can only work with primitive data types
d) A class that can only work with Strings

Q78.What is the Java Collections Framework?
a) A set of classes and interfaces implementing complex collection data structures
b) A set of classes and interfaces implementing simple collection data structures
c) A set of classes and interfaces implementing complex algorithms
d) A set of classes and interfaces implementing simple algorithms

Q79.Which interface represents a collection that does not allow duplicate elements?
a) Set
b) Map
c) List
d) Queue

Q80. What is an Iterator in Java?
a) An interface that represents a group of objects
b) An interface that represents a single object
c) An interface that represents a primitive data type
d) An interface that allows iterating over a collection

Q81 .What is a Map in Java?
a) An ordered collection of elements that can contain duplicates

b) An unordered collection of elements that cannot contain duplicates
c) A collection that orders its elements in a first-in, first-out (FIFO) manner
d) A collection that maps keys to values

Q82. What is Comparable in Java?
a) An interface that allows sorting of objects based on their natural order
b) An interface that allows sorting of objects based on a custom order
c) A class that allows sorting of objects based on their natural order
d) A class that allows sorting of objects based on a custom o

Q83 What is an IO Stream in Java?
a) A continuous flow of data
b) A physical layer that links to a stream
c) A class that reads data from a source
d) A class that writes data to a destination

Q84. What is JDBC in Java?
a) A Java API to connect and execute queries with the database
b) A Java API to connect and execute queries with the internet
c) A Java API to connect and execute queries with the file system
d) A Java API to connect and execute queries with the operating system

Q85. What is the purpose of JDBC in Java?
a) To provide a standard abstraction for Java applications to communicate with various databases
b) To provide a standard abstraction for Java applications to communicate with various internet protocols
c) To provide a standard abstraction for Java applications to communicate with various file systems
d) To provide a standard abstraction for Java applications to communicate with various operating systems

Q86. What will be the output of the following program?

```
List<Integer> list = new ArrayList<>();
list.add(10);
list.add(20);
list.add(30);
list.add(40);
System.out.println(list.get(2));
```

a) 10

b) 20

c) 30

d) 40

Q87. What will be the output of the following program?

```
Set<String> set = new HashSet<>();
set.add("apple");
set.add("banana");
set.add("orange");
set.add("apple");
System.out.println(set.size());
```

a) 1

b) 2

c) 3

d) 4

Q88.What will be the output of the following code snippet?

```
Map<String, Integer> map = new HashMap<>();
map.put("John", 25);
map.put("Mary", 30);
map.put("Peter", 35);
System.out.println(map.get("Mary"));
```

a) 25

b) 30

c) 35

d) null

Q89 What will be the output of the following Java code?

```
class GenericsExample {
    public static void main(String[] args) {
        GenericClass<Integer> gc1 = new GenericClass<>(10);
        GenericClass<String> gc2 = new GenericClass<>("Hello");

        System.out.println(gc1.getData());
        System.out.println(gc2.getData());
    }
}

class GenericClass<T> {
    private T data;

    public GenericClass(T data) {
        this.data = data;
    }

    public T getData() {
        return data;
    }
}
```
a) 10 Hello
b) Hello 10
c) Error
d) None of the above

Q90.What will be the output of the following Java code?

```
import java.util.*;

public class CollectionsExample {
    public static void main(String[] args) {
        List<Integer> list1 = new ArrayList<>(Arrays.asList(1, 2, 3));
        List<Integer> list2 = new ArrayList<>(Arrays.asList(4, 5, 6));

        Collections.copy(list1, list2);

        System.out.println(list1);
    }
}
```

a) [1, 2, 3]
b) [4, 5, 6]

c) [4, 5, 6, null, null, null]
d) Error