Q1
Which of the following JVM components converts the instructions in byte code to machine code?

Class loader

Byte code verifier

Interpreter

Javac

Correct Answer: Interpreter

The **Interpreter** is a component of the JVM that converts bytecode instructions into machine code at runtime. It executes Java bytecode line by line, making Java platform-independent.

Explanation of Other Options:
1. **Class Loader** → Loads `.class` files into memory for execution.
2. **Byte Code Verifier** → Checks bytecode for security and correctness before execution.
3. **Javac** → The Java compiler that converts Java source code (`.java`) into bytecode (`.class`).

Q2
[1 marks]
Which of the following tools is required to compile the Java programs?

JDK

JVM

JRE

Interpreter

Ans= JDK (Java Development Kit)

JDK (Java Development Kit):** The JDK contains the tools necessary to develop and run Java applications. This includes the Java compiler (`javac`), which is essential for compiling Java source code (`.java` files) into bytecode (`.class` files).

Let's look at why the other options are incorrect:

* JVM (Java Virtual Machine):** The JVM is responsible for executing the bytecode generated by the compiler. It's the runtime environment for Java applications, but it does not perform compilation.
* JRE (Java Runtime Environment):** The JRE provides the necessary libraries and the JVM to run Java applications. It doesn't include the compiler.
* Interpreter:** While Java can be considered an interpreted language in the sense that the JVM interprets bytecode, the `javac` compiler is the specific tool that performs the translation from source code to bytecode. "Interpreter" by itself, is too general.

In summary, the JDK is the development kit that provides the compiler (`javac`) needed to compile Java programs.

Q3

```
class Example {
    public static void main(String args[]) {
        Employee emp = new Employee();
        Employee empOne = new Employee();
        Employee empTwo = emp;
        Employee empThree = emp;
        emp = empOne;
        Employee empFour = new Employee();
    }
}
Ans=  Objects: 3
      References: 5

-------------------------------------------------------------------------
Q4

class Book {
    int stock;
    String name;

    Book(int stock, String name) {
        this.stock = stock;
        this.name = name;
    }

    public static void main(String[] args) {
        Book science = new Book(70, "Origin of the Species");
        Book maths = null;
        Book physics = science;
        Book social = science;
        Book english = new Book(50, "Wren&Martin");
    }
}

Ans=
Objects: 2
References: 5
-------------------------------------------------------------------------
Q5

[1 marks]
Which of the following statements is FALSE?

final instance variables can be initialized in the constructor of the class

There is no need to import classes which are present in the same package

A parent reference cannot hold a child object

ClassA in package1 can call method1 defined in a public ClassB present in
package2 only if ClassA imports ClassB

Ans=3

-------------------------------------------------------------------------
Q6

float expAnswer = num1 * num2 + num3 % (num2 - num1) - num2;
Therefore, the correct order of evaluation is:

i. num2 - num1
ii. num3 % (num2 - num1)
iii. num1 * num2
vi. num1 * num2 + num3 % (num2 - num1)
```

vii. num1 * num2 + num3 % (num2 - num1) - num2

Ans=The correct answer is: i, ii, iii, vi, vii

Q7

Which of the following statement(s) is/are INCORRECT?
a. Stacks follow LIFO
b. Queues can be constructed with the help of arrays or linked lists
c. Queue follows LIFO
d. Stacks can be constructed with the help of arrays or linked lists

Only a

Only b and c

Only b, c and d

Only c

Ans = Only C

Q8

What is the use of final keyword in Java?

When a class is made final, a sublcass of it cannot be created.

When a method is final, it cannot be overridden.

When a variable is final, it can be assigned value only once.

All the given options
Ans=The answer is: All the given options

Q9

[1 marks]

Consider the line given below which results in compilation error.

byte num=200;
How can the compilation error be resolved?
Choose THREE CORRECT statements from the below options.
Change the datatype from byte to int
Change the value from 200 to any positive value less than 128
Type cast the value 200 to byte
Replace 200 with 200.0

Ans1,2,3

Q10

```
import java.util.*;
public class Tester {
    public static void main(String[] args) {
        Set<Integer> set = new TreeSet<>();
        set.add(5);
        set.add(2);
        set.add(8);
        set.add(1);
        set.add(5);
        System.out.println(set);
    }
}
```

Ans=[1, 2, 5, 8]

Q11

[1 marks]

Consider the following queue of integers inQueue with capacity 7.

inQueue(front->rear): 23, 45, 6, 17, 15

What will be the contents of inQueue when the following operations are done?

```
inQueue.enqueue(43);
inQueue.dequeue();
inQueue.enqueue(12);
inQueue.dequeue();
```

inQueue(front->rear): 23, 45, 6, 17, 15

inQueue(front->rear): 6, 17, 15, 43, 12

inQueue(front->rear): 12, 43, 23, 45, 6

inQueue(front->rear): 23, 45, 6, 43, 12

Ans=6, 17, 15, 43, 12

Q12

[1 marks]

Consider the following statements:

```
int val1 = 25;
Float val2 = new Float(12.50);
boolean value3 = true;
Boolean value4= false;
```

Identify the number of object(s) of wrapper type.

Ans= 2

Q13

[1 marks]

Steven has written the following code but he is getting a compilation error.

```
class AdditionQuiz{
      public void add(int num1, int num2){
            System.out.println(num1+num2);
      }
      public void add(int num2, int num1){
            System.out.println(num1+num2);
      }
      public void add(float num1, float num2){
            System.out.println(num1+num2);
      }
      public void add(double num1, float num2){
            System.out.println(num1+num2);
      }
}}
```

Why is Steven getting the error?

There are two add methods which have the same number of parameters and the parameters order and datatypes are also same

All the add methods have the same parameter names and have same number of parameters

Addition of float with double is not possible without typecasting

All the given options

Option 1 is correct → "There are two add methods which have the same number of parameters and the parameters order and datatypes are also same."

Q14

[1 marks]
Consider the following code that depicts BINARY SEARCH algorithm for the list of elements sorted in ASCENDING ORDER

```
public static int search(int arrayOfElements[],int low,int high,int
elementToBeSearched) {
      if (low <= high) {
            int mid = (low + high) / 2;
            if (arrayOfElements[mid] == elementToBeSearched)
                  return mid;
            if (arrayOfElements[mid] < elementToBeSearched)
                  return search(arrayOfElements, mid + 1, high,
elementToBeSearched);
            return search(arrayOfElements, low, mid - 1, elementToBeSearched);
      }
      return -1;
}
```

Consider the arrayOfElements having 8 elements with low as 0 and high as 7. The elements of the array
12345678
How many iterations will be required to search if the elementToBeSearched is 6?
3
1
4
2
Ans=2

Q15

Number 14 needs to be seardhed using BINARY SEARCH 1,3, 7,9, 14, in the following sorted aray of numbers 19, 45
How many comparisons will be required to condude that the number 14 is found in the 5th position= Note: The index starts with 0 (zero)
3
2
4
1

Ans=3

Q16

[1 marks]

Consider the following code that depicts BINARY SEARCH algorithm for the array of elements sorted in ASCENDING ORDER.

```
public static int search(int arrayOfElements[],int low,int high,int
elementToBeSearched){
    if (low <= high) {
        int mid = (low + high) / 2;
        if (arrayOfElements[mid] == elementToBeSearched)
            return mid;
        if (arrayOfElements[mid] < elementToBeSearched)
            return search(arrayOfElements, mid + 1, high, elementToBeSearched);
        return search(arrayOfElements, low, mid - 1, elementToBeSearched);
    }
    return -1;
}
```

Consider the arrayOfElements having 10 elements with low as 0 and high as 9. The elements of the array are as follows.
116, 124, 135, 150, 167, 184, 212, 250, 280, 290

Identify the elements that will be compared when using the Binary Search technique if the elementToBeSearched is 116.

Ans=167, 124, 116

Q17

[1 marks]

Identify the PMD rules that are violated in the below code?

```
import JAVA.UTIL.*;
class Demo {
    public static void main(String args[]){
        new Demo().display();
    }
    void display(){
        String inStr1="Welcome";
        System.out.print("welcome");
    }
}
```

a. Unused import & print statement
b. Unused variable & package naming convention
c. Unused variable & method naming conventions
d. Unused import & Class naming convention

Ans=B

Q18

```
class Customer {

    private int custId;
    private String custName;

    public Customer(int custId) {
        this.custId = custId;
    }

    public void ValidateCustomerId() {
        if (this.custId > 9999) {
            System.out.println("Invalid customer id");
        } else {
            System.out.println("Valid customer id");
        }
    }
}
```

Which of the following programming flaws can be identified by the PMD tool?

Unused variable 'custName'

Invalid method maming convention

Use of System.out.println()

Ans=All the given options

Q19

[1 marks]

Examine the following code. What is the term for converting the wrapper class back to a primitive?

```
class Demo
{
    public static void main(String args[]){
        Integer price=new Integer(34);
        int finalPrice=price;
    }
}
```

Autoboxing

Unboxing

Type casting

Q20

[1 marks]

Identify the correct option which can be placed at Line1 in the below code to get the output as true?

Note: Line numbers are for reference only.

```
class Demo

{

    public static void main(String args[]){

        Long value1 = new Long("42");

        Long value2 = new Long(42);

        Short value3 = new Short("42");

        System.out.println(_________);//Line1

    }

}
```

value1.equals(value2)

value1==value2

value1.equals(value3)

value1==value3

Ans=1

Q21

[1 marks]

Consider the below problem statement

Problem Statement: Assume that you are given with two $20 currency notes, three $15 currency notes, thirty $1 currency notes. You are required to make $30 change with a minimum number of currency notes.

If greedy technique is used to solve the above problem then identify the combination of currency notes that the technique would produce.

Two $15

One $20 and Ten $1

Thirty $1

One $15 and Fifteen $1

Ans = 2

Q22

Problem Statement: Given a weighing machine and a bag of even numbered coins, identify the counterfeit coin. Bag contains ONLY one counterfeit coin. Counterfeit coins are ones which are lighter when compared to other coins.

Proposed Solution: Place half the coins on one side of the weighing machine and the rest of the coins on the other side. The side with the counterfeit coin will be lighter. Take only the coins on the lighter side and repeat the operation until the counterfeit coin is identified.

Identify the algorithmic design technique used in the proposed solution given above.

○ Greedy Technique

○ Dynamic Programming

○ Divide and Conquer

○ Brute Force Technique

Ans=Divide and Conquer

Q23

16. Mr. Peter is Planning to throw a huge backyard bash for 1000 people to celebrate a special occasion. He decides to break down the party into manageable sections like food, decorations, games, etc and assigns a person for planning and execution.

Identify the design technique for the scenario provided, which seems appropriate

Answer: Divide and Conquer

Q24
17. Consider a game in which a player has to reach the end of a maze by opening various doors. Player is given location of the first door. After opening the first door he would get a hint about the location of next door and so on till he reaches the last door.

Answer: LinkedList

Q25

Problem Statement: Consider a knapsack with a capacity 'c' consisting of 'n' items. Each of the items is associated with a weight and profit. The knapsack is to be packed such that :
· the sum of the weights of packed items should not exceed the capacity 'c' and
· the profit obtained must be maximum
Proposed Solution: Select the object with maximum profit that fits into the knapsack. Continue the process on the remaining objects until the conditions are satisfied.
Identify the algorithmic design technique used in the proposed solution given above.
Greedy Technique
Divide and Conquer
Dynamic Programming
Brute Force

✅ **Greedy Technique**

The proposed solution follows a **Greedy Approach** because it selects the item with the **maximum profit that fits** at each step without considering future choices. This locally optimal choice may or may not lead to the global optimal solution.

Q26

[1 marks]
A popular university, ReadWell, is developing an application for tracking the students enrolled for the various courses it offers. In case a student enrolls for more than one course then the application is expected NOT to store the duplicate records of a student.
Considering the scenario mentioned, identify the most suitable data structure for storing the student records.
LinkedList
HashSet
ArrayDeque
ArrayList
Ans = HashSet

Q27

[1 marks]

GetWellSoon hospital wants to keep track of its patients' blood report details. Each patient is assigned with a unique ID when they register the hospital. A patient may consult the hospital multiple times however the registration needs to be done only once. The blood report details must association with the ID and only the latest blood report needs to be stored.

Considering the scenario mentioned, identify the most suitable data structure for storing the blood report details.

○ HashMap

○ HashSet

○ LinkedList

○ ArrayDeque

Ans=1

Q28

[1 marks]
FlyHigh, an airline company, through an application wants to track the flight details like routes taken, passengers travelled, fight crewe expected to perform large number of retrieval operations for its smooth functioning.
Considering the scenario mentioned, identify the most suitable data structure for storing the flight details?
ArrayList
LinkedList
ArrayDeque
Set
Ans=ArrayList:
It provides fast random access to elements using their index (e.g., get(index)). This makes it efficient for retrieval operations.
It stores elements in contiguous memory locations, which contributes to faster access times.
Let's look at why the other options are less suitable:

LinkedList:
While it's good for insertions and deletions, it has slow random access because you need to traverse the list sequentially to reach a specific element.
ArrayDeque:
It's primarily used for implementing queues and stacks, optimized for operations at the ends (adding/removing). It's not the best choice for general retrieval.
Set:
Sets are designed for storing unique elements and are not optimized for ordered retrieval based on an index.

Q29

[1 marks]

Identify the INCORRECT statement from the given statements.

a. All classes in Java inherits the Object class

b. Wrapper classes override equals() method of Object class to che

c. The hashCode() method of Object class must return a float value

d. Two unequal objects can have the same hash code

Only c

Only d

a and c

Only c and d

Ans 1

Q30

[1 marks]

Identify the activities among the following that are carried out during the coding phase for a software development.

(Choose exactly two appropriate options)

Compilation of source code

Documenting user manual

Arriving at SRS

Finding user needs

Self-testing of implemented code

Ans=The two activities carried out during the coding phase are:

[1]Compilation of source code

[5]Self-testing of implemented code

Q31

[1 marks]

Consider the code given below which is written in a file 'Demo.java'.

```
class Book{
    //Class definition
}

class Demo{
    public static void main(String[] args){
        //Logic
    }
}
```

How many class files will be generated for the above code and which class ou

2, Demo

2, Book

1, Demo

Ans 1

Q32

Consider the problem size as 'n'. What is the worst case complexity of the algorithm given below?

```
if num > 100 then
for(counter1=0;counter1<n;counter1=counter1+1)
print("Inside if")
end-for

else

for(counter2=1;counter2<n;counter2=counter2*2)
print("Inside else")
end-for

end-if
```

O(nlogn)
O(n)
O(logn)
O(n^2)

Ans=2

Q33

[1 marks]

What is the time complexity of the code given below?

```
public void display(int num){
    for(int counter1=0; counter1<num; counter1++){
        for(int counter2=0; counter2<num; counter2++){
            System.out.println("success");
        }
    }
}
```

O(n2)

Q34
[1 marks]

Consider the code given below:

```
class Pet {
    public String preferredFood;
    public Pet(String preferredFood){
        this.preferredFood=preferredFood;
    }
    public void eatsFood(){ //Line 1
        System.out.println("Pet eats the desired food: "+this.preferredFood);
    }
    public void eatsFood(String food){ //Line 2
        System.out.println("Pet eats the desired food: "+food);
    }
    public static void main(String[] args) {
        Pet cat= new Pet("Fish");
    }
}
```

Which of the following method calls will call the method defined at Line 2?

a. cat.eatsFood();

b. cat.eatsFood("Milk");

c. cat.eatsFood("Curd");

○ Only a

○ Only b and c

○ Only c

○ All a, b and c

Q35

```
class Demo {
      public static void main(String[] args) {
            String mob = "1234512345";
            //Line1
            if (mob.matches(re)) {
                  System.out.println("valid");
            } else {
                  System.out.println("invalid");
            }
      }
}
```

Note: Line numbers are for reference only.

□ String re="\d{10}";

□ String re="[0-9]{10}";

□ String re="\\d{10}";

□ String re="\D{10}";

Ans=1,2

Q36

[1 marks]

John is trying to write a regular expression for searching a string which should consist of only lowercase and uppercase letters and mus teammates of John suggest few expressions.

Identify the suggestion(s) which would produce the desired output.

1. Peter suggested to write it as ^([(?i)a-z]*S)

2. Smith suggested to write it as ^((?i)[A-Z]*s)

Only Peter

Only Smith

Both Peter and Smith

None of the suggestions

Ans=4

Q37 of 20

Which of the asymptotic notations is used to represent the best-case analysis of an algorithm?

O Big Theta

O Big Omega

O Big Oh

O There is no asymptotic notation for representing best-case analysis

Ans=c

Q38 of 20

What is the time complexity of the code given below?

```
int number1 = 0, counter = 10;
while (counter > 0) {
    number1 += counter;
    counter /= 2;
}
```

O(n)

O(sqrt(n))

O(n/2)

O(log n)

Ans D

---**

Q39 of 20

What is the best-suited condition for using linear search algorithm?

When the array is sorted

When the array contains huge number of elements

When the array contains only Integer elements

When the array contains only few elements

Ans= When the array contains only few elements

---**

Q40 of 20

Consider an array, arr ={12,16,17,19,23,35,40}.

How many iterations are required to search 23 using binary search algorithm?

0

2

3

4

Ans= 3

---**
Q41 of 20

What is the time complexity of bubble sort algorithm?

O(n*n)

O(n*log n)

O(n)

O(1)

Ans=1

---**
Q42 of 20

How many minimum numbers of stacks are needed to implement a queue?

1

2

3

It is not possible to implement a queue using stack

Ans=2

Q43 of 20

Which of the given algorithmic approach tries to achieve a localized optimum solution?

Dynamic programming

Divide and conquer

Brute force

Greedy approach

Ans =4

---**
Q44

[1 marks]

Consider the code given below:

```
class AdditionQuiz{
    int score;
    int num1;
    int num2;

    public AdditionQuiz(int num1, int num2){
        this.score=0;
        this.num1=num1;
        this.num2=num2;
    }

    public String evaluateQuiz(int answer){
        if((this.num1+this.num2)==answer){
            this.score+=1;
            return "Good job";
        }
        return "Try again";
    }
}

class MultiplicationQuiz extends AdditionQuiz {

    MultiplicationQuiz(int num1, int num2) {
        super(num1, num2);
    }

    @Override
    public String evaluateQuiz(int answer) {
        if ((this.num1 * this.num2) == answer) {
            this.score += 2;
            return "Well done";
        }
        return "Can do better";
    }

    public static void main(String[] args) {
        AdditionQuiz quiz1 = new AdditionQuiz(2, 3); // Line 1
        System.out.println(quiz1.evaluateQuiz(5)); // Line 2
    }
}
```

Which of the following modifications can be done on Line 1 so that Line 2 will call the overridden method in

Note: Line numbers are for reference only

a. AdditionQuiz quiz1=new MultiplicationQuiz(2,3);

b. MultiplicationQuiz quiz1=new MultiplicationQuiz(2,3);

c. MultiplicationQuiz quiz1=new AdditionQuiz(2,3);

Only b

Only a and b

Only b and c

All a, b and c

Ans= Only a and b

==
============

Q45
2 Mark]

```
class LoopTester{
public static void main(String[] args) {
            // TODO Auto-generated method stub
            int endNum;
        for (endNum = 1; endNum <= 10; endNum++) {
            if (endNum > 5) {
                continue;
            }
        }
        for (int startNum = 0; startNum < endNum; startNum++) {
            endNum--;
            System.out.print(startNum + " " + endNum + " ");
            if (endNum < 8) {
                break;
            }
        }
      }
}
```

10 1 9 2 8
10 1 9 2 8 3 7
05
11 1 10 2 9 3 8

Ans-0 10 1 9 2 8 3 7

Q46

[2 marks]

What is the output of the code given below?

```
class Demo{
    public static void main(String[] args){
        int counter=1;
        int sum = 0;
        for(;counter<=6; counter++){
            sum+=counter;
            if(sum==10){
                continue;
            }
            if(sum > counter*2){
                break;
            }
        }
        do {
            sum-=counter;
        }while(sum>counter);
        System.out.println("Sum:"+sum);
    }
}
```

Ans=5

Q47

2 marks]

What is the output of the code given below?

```
class Demo {
      public static void main(String[] args) {
            int num1 = 1;
            int num2 = 1;
            int sum = 0;
            int outputNum = 0;
            for (; num1 < 10; num1 = num1 + 2) {
                  sum += num1;
                  for (; num2 < 5; num2++) {
                        sum -= num2;
                  }
            }
            switch (sum) {
            case 15:
                  outputNum = 15;
            case 30:
                  outputNum = 10;
                  break;
            case 35:
                  outputNum = 20;
                  break;
            default:
                  outputNum = -1;
            }
            System.out.println(outPutnum);
      }
}
```

20
10
-1
15
Ans=10

Q48

```
class Student {
    protected int rollNo;
    protected String studentName;

    public Student() {
        this.rollNo = 1001;
        this.studentName = "Roger";
    }

    public Student(int rollNo, String studentName) {
        this.rollNo = rollNo;
        this.studentName = studentName;
    }

    protected void display() {
        System.out.println(this.rollNo + " " + this.studentName);
    }
}

class Hostelite extends Student {
    private int hostelId;
    private String blockName;

    public Hostelite( int hostelId, String blockName) {

        this.hostelId = hostelId;
        this.blockName = blockName;
    }

    public void display() {
        super.display();
        System.out.println(this.hostelId + " " + this.blockName);
    }

    public static void main(String[] args) {
        Hostelite ben = new Hostelite( 2001, "Arthur");
        ben.display();
    }
}
```

Compilation Error: The constructor of Student class is not invoked

```
B} 1001 Roger
   2001 Arthur
```

```
0 null
2001 Arthur
```

```
2001 Arthur
1001 Roger
```

Ans= B

Q49
[1 marks]

Which of the following should be replaced in Line1 to get the output as 26?

```
class ClassA{
    private int num;
    public ClassA(int num){
        this.num=num;
    }
    public void setNum(int num){
        this.num=num;
    }
    public int method1(int var){
        return var+this.num;
    }
}

class ClassB{
    private ClassA refA;
    private int num1;
    public ClassB(ClassA refA){
        this.refA=refA;
        this.num1=10;
    }

public int method3(int val){
    this.refA.setNum(this.num1+val);
    return      //Line1
}

class Tester{
    public static void main(String[] args){
        ClassA objA=new ClassA(5);
        ClassB objB=new ClassB(objA);
        System.out.println(objB.method3(3));
    }
}
}
```

Note: Line numbers are for reference only.

```
this.refA.method1(16)+this.num1;
```

```
this.refA.method1(3)+this.num1; //23+3
```

```
this.refA.method1(11)+this.num1;
```

this.refA.method1(6)+this.num1;

Ans =[B]this.refA.method1(3)+this.num1;

Q50

[2 marks]
What is the output of the code given below?

```
abstract class PrintRequest {
      int noOfPages;
      private double price;

      public PrintRequest(int noOfPages) {
            this.noOfPages = noOfPages;
      }

      public void setPrice(double price) {
            this.price = price;
      }

      public double getPrice() {
            return this.price;
      }

      public abstract void calculatePrice();
}

class InvitationPrintRequest extends PrintRequest {
      int noOfCopies;

      public InvitationPrintRequest(int noOfPages, int noOfCopies) {
            super(noOfPages);
            this.noOfCopies = noOfCopies;
      }

      public void calculatePrice() {
            double basicPrice = this.noOfPages * 2;
            double finalPrice = basicPrice * 50;
            double discount = this.calculateDiscount(finalPrice);
            this.setPrice(finalPrice - discount);
      }

      public double calculateDiscount(double finalPrice) {
            double baseDiscount;
            if (this.noOfCopies >= 25 && this.noOfPages > 15) {
                  baseDiscount = finalPrice * 0.20;
            } else {
                  baseDiscount = finalPrice * 0.10;
            }
            return baseDiscount;
      }

      public static void main(String[] args) {
            InvitationPrintRequest invitationObj = new
InvitationPrintRequest(20, 20);
            invitationObj.calculatePrice();
            System.out.println(invitationObj.getPrice());
      }
}
```

options
Compilation Error: abstract class PrintRequest cannot have a concrete method
1800.0

1600.0
2000.0
Ans=1800.0

Q51

[1 marks]
What should be replaced at Line1 to obtain output as below:
G002 is available for use
false, true
Note: Line numbers are for reference only.

```
class ClassRoom {
      public String classRoomNo;
      private boolean available;

      public ClassRoom(String classRoomNo) {
            this.classRoomNo = classRoomNo;
            this.available = true;
      }

      public boolean getAvailable() {
            //**
            return this.available;
      }

      public void setAvailable(boolean available) {
            this.available = available;
      }
}

class Educator {
      public void checkClassRoomAvailability(ClassRoom classRoom) {
            if (classRoom.getAvailable()) {
                  System.out.println(classRoom.classRoomNo + " is available for
use");
                  classRoom.setAvailable(false);
            } else {
                  System.out.println(classRoom.classRoomNo + " is unavailable");
            }
      }
}

class Tester {
      public static void main(String[] args) {
            ClassRoom classRoomRef1 = new ClassRoom("G002");
            ClassRoom classRoomRef2 = new ClassRoom("L1002");
            new Educator().checkClassRoomAvailability(classRoomRef1);
            System.out.println(); // Line1
      }
}
```

```
classRoomRef1.getAvailable()+", "+classRoomRef2.getAvailable()

classRoomRef1.getAvailable() +", " + classRoomRef1.getAvailable()

classRoomRef2:getAvailable()+", "+classRoomRef1.getAvailable()

classRoomRef2.getAvailable()+", "+classRoomRef2.getAvailable()
```

Ans=1

Q52

[2 marks]
What is the output of the code given below?

```
class Student {
      public String stdName;
      public int stdld;
      public String stdBranch;

      public Student() {
      }

      public Student(String stdName, int studentld, String stdBranch) {
            this.stdName = stdName;
            stdld = studentld;
            stdBranch = stdBranch;
      }
}

class Demo {
      public static void main(String args[]) {
            Student std1 = new Student();
            System.out.println("stdName:" + std1.stdName + ", " + "stdBranch:" +
std1.stdBranch + ", " + "stdld:" + std1.stdld);
            Student std2 = new Student("John", 1234, "CSE");
            System.out.println("stdName:" + std2.stdName + ", " + "stdBranch:" +
std2.stdBranch + ", " + "stdld:" + std2.stdld);
      }
}
```

```
stdName: null, stdBranch: null, stdld: 0

stdName: John, stdBranch: null, stdld: 1234

stdName: null, stdBranch: null, stdld: 0
stdName: John, stdBranch: null, stdld: 0

stdName: null, stdBranch: null, stdld: 0
stdName: null, stdBranch: null, stdld: 0

stdName: null, stdBranch: null, stdld: 0
stdName: John, stdBranch: CSE, stdld:1234
```

Ans=1

Q53

[1 marks]
What is the output of the code given below when run with the default Junit runner ?

```
class Computation {
      public int add(int num1, int num2) {
            return num1 + num2;
      }

      public int divide(int num1, int num2) {
            return num1 / num2;
      }
}

public class TestComputation {
      Computation comput = new Computation();
@Test
public void testAdd1()
```

```
      {
            int expected = 5;
            int actual = comput.add(2, 3);
            Assert.assertEquals(expected, actual);
      }

      @Test
      public void testAdd2() {
            int expected = 7;
            int actual = comput.add(2, 5);
            Assert.assertEquals(expected, actual);
      }
}
```

Both testAdd1 and testAdd2 fails

testAdd1 fails and testAdd2 passes

Both testAdd1 and testAdd2 passes

testAdd1 passes and testAdd2 fails

Ans 3

Q54

[2 marks]

What is the output of the code given below?

```
abstract class Demo
{
    public int num;
    Demo(){
        num = 10;
    }
    abstract public void setNum();
    abstract final public void getNum();
}

class Test extends Demo
{
    public void setNum(int num){
        this.num = num;
    }
    final public void getNum(){
        System.out.println("num = " + num);
    }
    public static void main(String[] args){
        Test obj = new Test();
      obj.setNum(30);
      obj.getNum();
}
}
```

10,

20,

30,

final method cant be overridden

Ans =D

Q55

[2 marks]

What is the output of the code given below?

```
public class Tester {
    public static void main(String args[]){
        String tempStr = "ItsAVerySimpleProblem";
        tempStr = tempStr.toLowerCase();
        String tempOne = tempStr.substring(3,tempStr.length()-2);
        String finalValue = "";
        int index1 = 3;
        if(tempOne.substring(tempOne.length()-1).equals("e")){
            for(int counter1 =0;counter1 <tempOne.length()-5;counter1++){
                finalValue += tempOne.substring(counter1-6);
            }
        }
        else{
            finalValue += tempOne.substring(index1-2);
        }
        System.out.println(finalValue);
    }
}
```

Avery
Verysimple
verysimpleprobl
Simple

Ans=verysimpleprobl

Q56

[2 marks]

What is the output of the code given below?

```
class Demo {

      private static int var1 = 40;

      final int var3 = 40;

      private int var2 = 40;

      public Demo(int var2) {

            this.var2 = var2;

            this.var2 = ++this.var2 - var3;

            Demo.var1 = Demo.var1--;

            Demo.var1--;

      }

      public int method10() {

            return Demo.var1 + this.var2;

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

            Demo.var1--;

            Demo obj1 = new Demo(40);

-
            Demo obj2 = new Demo(30);

            System.out.println(obj1.method10()+obj2.method10());

      }
}
```

66,
70,
64,
76
Ans=66

Q57

What is the output of the code given below?

```
class Toy{
    protected int toyId;
    protected float price;
    public Toy(){
        this.toyId=3001;
        this.price=100.0f;
    }
    protected void displayDetails(){
        System.out.println("Toy Id:"+this.toyId+" Price:"+this.price);
    }
}

public class SoftToy extends Toy{
    private int discount;
    SoftToy(int discount, float price){
        this.toyId++;
        this.price=price;
        this.discount=discount;
    }

public void calculateTotalPrice(){
    this.price=this.price-(this.price*(float)this.discount)/100;
}

public static void main(String args[]){
    Toy car=new Toy();
    car.displayDetails();
    SoftToy teddyBear=new SoftToy(10.300);
    teddyBear.calculateTotalPrice();
    teddyBear.displayDetails();
}
}
```

```
Toy Id:3001 Price:100.0
Toy Id:3002 Price:270.0
```

```
Toy Id:3001 Price:100.0
Toy Id:3001 Price:270.0
```

~~~
Toy Id:3002 Price:100.0
Toy Id:3002 Price:270.0

Toy Id:3001 Price:300.0
Toy Id:3002 Price:270.0
```
~~~

Ans=1

Q58

~~~java
class ClassX {
      private int val;

      public ClassX(int val) {
            this.val = val;
      }

      public int method1() {
            return this.val / 2;
      }
}

class ClassA {
      protected static int counter = 0;
      private int num1;

      public ClassA(int val1){
            ClassA.counter += 1;
            this.num1 = val1;
      }

      public int method2() {
            this.num1 *= 3;
            return this.num1;
      }

      public int getNum1() {
            return this.num1;
      }

}

class ClassB extends ClassA {
      private static int variable = 100;
      private int num2;
      private ClassX objX;

      public ClassB(int val1, int val2, ClassX obj){
            super(val1);
            ClassA.counter -= 1;
            this.num2 = val2 + ClassB.variable;
            this.objX = obj;
      }

      public int method1() {
            return this.objX.method1() + this.getNum1() + this.num2;
      }
}

      class Tester {
            public static void main(String args[]){
~~~

```
                  ClassX obj1 = new ClassX(20);
                  ClassB obj2 = new ClassB(15, 20, obj1);
                  System.out.println(obj2.method1() + obj2.method2() +
ClassA.counter);
            }
      }
```

Ans=190

Q59

[2 marks]

What is the output of the code given below?

```
class Demo {
      private static int var1 = 40;
      final int var3 = 40;
      private int var2 = 40;

      public Demo(int var2) {
            this.var2 = var2;
            this.var2 = ++this.var2 - var3;
            Demo.var1 = Demo.var1--;
            Demo.var1--;
      }

      public int method1() {
            return Demo.var1 + this.var2;
      }

      public static void main(String args[]) {
            Demo.var1--;
            Demo obj1 = new Demo(40);
            Demo obj2 = new Demo(30);
            System.out.println(obj1.method1() + obj2.method1());
      }
}
```

66
70
64
76
Ans=66

Q60

[2 marks]

What is the output of the code given below?

```
class Patient {
      private int patientId;

      public Patient(int patientId) {
            this.patientId = patientId;
      }

      public int identifyAmt(double[] amountArr, int[] patientIdArr) {
            double amt = 0.0;
            int counter = 0;
            for (int index = 0; index < patientIdArr.length; index++) {
                  if (patientIdArr[index] == this.patientId) {
                        amt = amountArr[index];
                        break;
                  }
            }
            for (int index = 0; index < patientIdArr.length; index++) {
                  if (amountArr[index] > amt) {
                        counter++;
                        break;                           // if there is no break
then ans is 2
                  }
            }
            return counter;
      }
}

class Demo {
      public static void main(String[] args){
            double[] amountArr = {12,13,16,16,19,20,10,9};
            int[] patientIdArr = {1002,1004,1001,1003,1005,1009,1011,1008};
            Patient patientObj = new Patient(1001);
            System.out.println(patientObj.identifyAmt(amountArr, patientIdArr));
      }
}
```

1

3

2

Runtime Exception: Array index out of bound

Ans=1

Q61

[2 marks]

What is the output of the code given below?

```
class DailyWager {
      public String wagerName;
      public String wagerAddress;

      public DailyWager() {
            this();             // if this remove then ans is 1
            System.out.println("Parameterless constructor called");
      }

      public DailyWager(String wagerName, String wagerAddress) {
            this.wagerName = wagerName;
            this.wagerAddress = wagerAddress;
      }

      public void displayWagerName() {
            System.out.println("Wager Name: " + this.wagerName);
      }



      public void displayDetails() {
            this.displayWagerName();
            System.out.println("Address: " + this.wagerAddress);
      }
}

public class Demo {
      public static void main(String[] args) {
            DailyWager wager1 = new DailyWager("Hales", "Canada");
            wager1.displayDetails();
      }
}
```

```
Wager Name: Hales
Address: Canada
```

```
Parameterless constructor calle
Wager Name: Hales
Address: Canada
```

```
Wager Name: null
Address: null
```

```
Compilation Error: Recursive constructor invocation
```

ans=D

Q62

[2 marks]

What is the output of the code given below?

```
abstract class AbstractClass {
      private String str1;
      String value1;

      AbstractClass(String label, int num) {
            str1 = "" + num;
            value1 = label;
      }

      String getLabel() {
            return str1;
      }

      abstract String getNumber();
}

class ConcreteClass extends AbstractClass {

      ConcreteClass(String label, int num) {
            super(label, num);
      }

      String getNumber() {
            return value1;
      }
}

class Tester {
      public static void main(String args[]) {
            ConcreteClass obj = new ConcreteClass("AJ", 1);
            System.out.println(obj.getNumber());
            System.out.println(obj.getLabel());
      }
}
```

AJ
1

Q63

[2 marks]

What is the output of the code given below?

```
class Student {
      private int studentId;
      private int yearOfEnrollment;
      private static int counter1 = 2014101;
      private static int counter2 = 101;

      public Student(int yearOfEnrollment) {
            this.yearOfEnrollment = yearOfEnrollment;
            if (this.yearOfEnrollment == 2014) {
                  this.studentId = Student.counter1++;
            } else {
                  this.studentId = Student.counter2++;
            }
      }

      public static int totalNumberOfStudents() {
            return (Student.counter2 - 101);
      }

      public static int totalNumberOfStudents(int year) {
            return (Student.counter1 - 2014100);
      }

}

public class Demo {
      public static void main(String[] args) {
            Student studentObj1 = new Student(2014);
            System.out.println(Student.totalNumberOfStudents(2014));
      }
}
```

Compilation Error: Static variables cannot be accessed inside constructor

0

2

1

Ans=2

Q64
[2 marks]
What is the output of the code given below?

```
public class Demo {

      public static void main(String[] args) {
            String stringOne="Sachin ";
            String stringTwo="Tendulkar";
            String stringThree=stringOne.concat(stringTwo);
            String stringFour=new String("Sachin Tendulkar");
            if(stringThree.equalsIgnoreCase(stringFour)){
                  System.out.println("Equal");
            }
            else if(stringThree == stringFour){
                  System.out.println("Equal");
            }
            else{
                  System.out.println("Not Equal");
            }
            System.out.println(stringThree);
      }

}
```

```
Equal
Sachin Tendulkar
```

```
Equal
```

```
Equal
SachinTendulkar
```

```
Not Equal
Sachin Tendulkar
```

Ans=1

Q65

[2 marks]
Consider the code given below.
Identify the code that needs to be filled in Line 1, 2 and 3 respectively such that:
the studentld is auto-generated starting from 501 in steps of 1
· the method "getNoOfStudent" returns the total number of students enrolled at any given point.

```
class Student {
      private int studentld;
      private String studentName;
      private int yearOfEnrollment;
      public static int counter;
      static {
//Line 1
      }

      public Student(String name, int yearOfEnrollment) {
            this.studentName = name;
            this.yearOfEnrollment = yearOfEnrollment;
// Line 2
      }

      public static int getNoOfStudent() {
            // Line 3
      }
}
```

Note: Line numbers are for reference only.

```
Line 1: Student.counter=501;
Line 2: this.studentld=Student.counter++;
Line 3: return (Student.counter-500);
```

```
Line 1: Student.counter=501;
Line 2: this.studentld=++Student.counter;
Line 3. return (Student counter-501).
```

```
Line 1: Student.counter=500;
Line 2: this.studentld=Student.counter++; I
Line 3: return (Student.counter-500);
```

```
Line 1: Student.counter=500;
Line 2: this.studentld=++Student.counter;
Line 3: return (Student.counter-500);
```

Ans=D

Q66

```
class ExceptionExample {

      public void checkForExceptions(int num1, int num2) {
            int intArr[] = { 1, 2, 3 };
            String str = null;
            try {
                  str.charAt(0);
                  System.out.println(num1 / num2);
                  System.out.println("Enjoy no exception");
            } catch (ArithmeticException e) { // Line 1
                  System.out.println("ArithmeticException handler!");
            } catch (NullPointerException e) { // Line 2
                  System.out.println("NullPointerException handler!");
            } catch (Exception e) { // Line 3
                  System.out.println("Default exception handler!");
            }
      }
}

class Tester {
      public static void main(String[] args) {
            ExceptionExample exceptionExample = new ExceptionExample();
            exceptionExample.checkForExceptions(2, 0);
      }
}
```

Which of the below catch block(s) will get executed?
a. catch block placed at Line1
b. catch block placed at Line2
c. catch block placed at Line3
Note: Line numbers are for reference only
Only a
Only b
Only c
Only b &c
Ans=B

Q67
[3 marks]

What is the output of the code given below?

```
class Demo{
    public static int specialAdd(int num1){
        if (num1!=0)
            return (num1+2)+specialAdd(num1-1);
        else
            return 3;
    }
    public static int extraordinaryAdd(int num2){
        if (num2!=0)
            return specialAdd(num2)+extraordinaryAdd(num2-1);
        else
            return 0;
    }
    public static void main(String[] args){
        System.out.println(extraordinaryAdd(5));
    }
}
```

80
52
70
25
Ans=80

Q68

[3 marks]

What is the output of the code given below?

```
class Demo{
    public static void main(String args[]){
        Demo obj=new Demo();
        System.out.println(obj.test(4, "Sam"));
    }
    public String test(int num, String name){
        if(num == 0){
            return name;
        }
        else if(name.equals(name+"43")){
            num = num + 1;
            return test(num, name);
        }
        else{
            name = name + "" + num;
            num = num - 1;
            return test(num, name);
        }
    }
}
```

Sam4321
Sam434321
Sam
4321Sam

Ans=Sam4321

Q69

[3 marks]

What is the output of the code given below?

```
public class Demo {
      public static int count(int chocolate, int wrap) {
            if (chocolate < wrap)
                  return 0;
            int newChocolates = chocolate / wrap;
            return newChocolates + count(newChocolates + chocolate % wrap,
wrap);
      }
      public static int findMaxChocolates(int money, int price, int wrap) {
            int chocolate = (money / price);
            return chocolate + count(chocolate, wrap);
      }

      public static void main(String[] args) {
            int money = 45;
            int price = 5;
            int wrap = 3;
            System.out.println(findMaxChocolates(money, price, wrap));
      }
}
```

14
13
9
15
Ans=13

##############PartOne And Part2#############

Java MCQ

1)What will be the output for the below code?

```
import java.util.*;
public class demo{
    public static void main(String args[]){
        List<Integer> intList = new ArrayList<Integer>();
        intList.add(35);
        intList.add(40);
        intList.add(45);
        intList.add(50);
        intList.add(55);
        Set<Integer> inSet=new TreeSet<Integer>();
        inSet.add(30);
        inSet.add(45);
        inSet.add(65);
        inSet.add(45);
        inSet.add(55);
        for(int val=0;val<intList.size();val++) {
            if(inSet.add(intList.get(val)))
                  System.out.println(intList.get(val)+",");
        }
        System.out.println(""+inSet);
    }
}
```

Answer:- 35,40,50
[30, 35, 40, 45, 50, 55, 65]

2. What will be the output for the below code?

```
class Demo
{
      int count = 5;
      public void methodOne()
      {
            this.count++;
      }
      public void methodTwo(int num)
      {
            count = count + num ;
      }
      public int getCount()
      {
            return count;
      }
}
public class Tester{
      public static void main(String args[])
      {
            Demo demo1 = new Demo();
            Demo demo2 = new Demo();
            demo1.methodOne();
            demo2.methodTwo(demo1.getCount());
            System.out.println(demo2.getCount());
      }
}
```

Output: 11

3. Consider the following sequence of operations performed on an empty Queue:

```
enqueue(7)
enqueue(1)
enqueue(5)
dequeue()
enqueue(2)
dequeue()
enqueue(3)
enqueue(1)
dequeue()
enqueue(4)
```

After performing the above questions in the order shown, what will be the elements in the queue?
Note: Elements in options are given from front to rear.

Answer: 2, 3, 1, 4

4. Consider this skeleton code given below

```
interface interfaceOne
{
 void methodOne;
}
abstract class AbstractClass implements interfaceOne {
      public abstract void methodTwo();
      public void methodThree()
      {
            System.out.println("method 3");
      }
}
class AbsChild extends AbstractClass
{

}
```

This will lead to a compilation error. which among the given options will prevent the compilation error inside the AbsChild
(select 2 incorrect options)
a)implement methodOne() and methodTwo() inside AbstractClass
b)Override the methodTwo() inside the AbsChild class
c)Override the methodTwo(), methofThree() inside the AbsChild class
d)Declare the class AbsChild as abstract class

Answer: B,D

4. What will be the output for the below code?

```
public static void main(String[] args) {
            int total = 1;
            for(int i=1;i<10;i++)
            {
                  if(i/4==0.75 || i % 4==2)
                  {
                        continue;
                  }
                  total = total + i ;
            }
            System.out.println(total);
      }
```

Answer: 38

5. What will be the output for the below code?

```
public static void main(String[] args) {
            double n1=7;
            double n2=5;
            if((n1/n2 >=1) &&(n2/n1 < 0.5))
            {
                  System.out.println("One");
            }
            else if((n1+n2 )>=10 &&(n2/n1 >= 0.5))
            {
                  System.out.println("Two");
            }
            else
            {
                  System.out.println("three");
            }
      }
```

Answer: Two

6.

7. What will be the output for the below code?

```
class Cale
{
      public int div(int num1,int num2)
      {
            try
            {
                  return num1/num2;
            }
            catch(ArithmeticException exp)
            {
                  System.out.println("AE");
            }
            return -1;
      }
}

public class Tester{
      public static void main(String args[])
      {
            Cale cale = new Cale();
            int arr[] = {20,10,5,0};
            System.out.println(cale.div(arr[1],arr[3]));
      }
}
```

Answer:- AE
-1

8. What is the time complexity of the logic

```
            int num=10;
            for(int index=0;index<=num;index++)
            {
                  for(int index2=index;index2<=num;index++)
                  {
                        System.out.println("index"+index+"index2"+index2);
                  }
            }
```

Answer: O(n^2)

9. What is the output for the below code

```
public static void main(String args[])
      {
            int num=0;
            for(int index=1;index<=5;index++)
            {
                  for(int index2=index;index2<=5;index2++)
                  {
                        if(index % index2 ==0)
                        {
                              continue;
                        }
                        num++;
                  }
            }
            System.out.println("num"+num);
      }
```

Answer: 10

10. Consider the list of numbers given below 15,14,21,6,2 what will be the state of the list at the end of the 2 pass using bubble sort algorithm to sort the given list in ascending order?

Answer: 14,6,2,15,21

11.

Answer: (21,0,21,20)

12. class Swap

```
{
      private int num1=10;
      private int num2=20;
      public void swap(Swap s2)
      {
            int temp = this.num1;
            this.num1=s2.num2;
            s2.num2=temp;
      }
      public void disp()
      {
            System.out.println(num1+" "+num2);
      }
}
public class Tester{

      public static void main(String args[])
      {
            Swap swap1 = new Swap();
            Swap swap2 = new Swap();
            swap1.disp();
            swap2.disp();
      }
}
```

Answer:

```
10 20
10 20
```

```
13. class Shape
{
      public void draw()
      {
            System.out.println("drawing a shape");
      }
}
class Rectange extends Shape
{
      public void draw()
      {
            System.out.println("drawing a rect");
      }
}


public class Tester{

      public static void main(String args[])
      {
            Shape s1=new Shape();
            s1.draw();
      }
}
```

Answer: drawing a shape

```
14. public class Tester{

      public static void main(String args[])
      {
            LinkedList linkedList = new LinkedList();
            linkedList.addAtEnd("1");
            linkedList.addAtEnd("2");
            linkedList.addAtEnd("3");
            linkedList.addAtEnd("4");
            linkedList.addAtEnd("5");
            updateList(linkedList.getHead());
            linkedList.display();
      }
      public static void updateList(Node head)
      {
            Node temp=head;
            while(temp.getNext().getNext()!=null)
            {
                  int val=Integer.parseInt(temp.getNext().getData());
                  temp=temp.getNext();
                  val+=Integer.parseInt(temp.getData());
```

```
                temp.setData(""+val);
            }
            head.setData(temp.getData());
        }
}
```

Answer: 8,4,6,8,5

```
15. public class Tester{

        public static void main(String args[])
        {
            double arrNum[] = {10.5,5.0,15.0,3.5,2.5};
            for(int i=0; i <arrNum.length;i=i+2)
            {
                arrNum[i]=arrNum[i]+i;
            }
            for(int i=0; i <arrNum.length;i++)
            {
                System.out.println(arrNum[i]+" ");;
            }
        }
}
```

Answer: 10.5 5.0 17.0 3.5 6.5

16. Mr. Peter is Planning to throw a huge backyard bash for 1000 people to celebrate a special occasion. He decides to break down the party into manageable sections like food, decorations, games, etc and assigns a person for planning and execution.
Identify the design technique for the scenario provided, which seems appropriate

Answer: Divide and Conquer

17. Consider a game in which a player has to reach the end of a maze by opening various doors. Player is given location of the first door. After opening the first door he would get a hint about the location of next door and so on till he reaches the last door.

Answer: LinkedList

```
18. class Parent
{
        private int num1=5;
        public int modifyNum()
        {
            return ++num1;
        }
}
class Child extends Parent
{
        private int num2=10;
        public int modifyNum()
        {
            this.num2=this.num2+super.modifyNum();
            return num2;
        }
}

public class Tester{

        public static void main(String args[])
        {
            Parent parent = new Child();
            System.out.println(parent.modifyNum());
      }
}
```

Answer: 16

19.

20. Which of the following words will match with the given regular expression |r(a|e|i|o|u)]+n

Answer: ran

```
21. public class Tester{

      public static void main(String args[])
      {
            int num1=9;
            double num2=4.5;
            float num3=2F;
            int res=num1+(int) (num2/num3);
            System.out.println(res);
      }
}
```

Answer: 11

```
22.public class Tester{

      public static void main(String args[])
      {
            char ch='*';
            switch(ch)
            {
            case '#':System.out.println("#");
            case '*':System.out.println("*");
            case '$':System.out.println("$");
            break;
            default: System.out.println("no match");
            }
      }
}
```

Answer: * $

23.

```
24. public class Tester{
      private static final int MAX_VAL;
      static
      {
            MAX_VAL=100;
            System.out.println("Static");
      }
      public static void main(String args[])
      {

            System.out.println("main");
            System.out.println(MAX_VAL+"");
            System.out.println("end");
      }
}
```

Answer:
Static
main
100
end

25.

1. What will be the output of the following Code Snippet?

```
public class Demo {
      public static void main(String[] args)
      {
            Stack stack1 = new Stack(5);
            stack1.push(44);
            stack1.push(40);
            stack1.push(24);

            Stack stack2 = new Stack(5);
            stack2.push(23);
            stack2.push(24);
            stack2.push(52);

            Queue outputQueue = returnQueue(stack1,stack2);
            outputQueue.display();
      }
      public static Queue returnQueue(Stack stack1, Stack stack2)
      {
            Queue outputQueue = new Queue(10);
            while(!stack1.isEmpty())
            {
                  Stack tempStack = new Stack(10);
                  int element1 = stack1.pop();
                  while(!stack2.isEmpty())
                  {
                        int element2 = stack2.pop();
                        if(element1%10 == element2%10)
                        {
                              outputQueue.enqueue(element1);
                        }
                        tempStack.push(element2);
                  }
                  stack2 = tempStack;
            }
            return outputQueue;
      }
}
```

Assumption: All references to necessary files are available.

Answer: 24,44

2. What will be the output of the following Code Snippet?

```
public class Demo1 {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            List<String> strList = new ArrayList<String>();
            strList.add("Welcome");
            strList.add("to");
            strList.set(1,"in");
            strList.add("Java");
            Iterator<String> iterator = strList.iterator();
            while(iterator.hasNext())
            {
                  System.out.println(iterator.next());
                  if(strList.contains("to"))
                  {
                        strList.set(1, "Programming");
                  }
                  else
                  {
                        strList.add("World");
                  }
                  System.err.println(iterator.next());
            }
      }

}
```

Answer: Welcome
followed by Runtime Exception: Concurrent modification on Set

3. What will be the output of the following Code Snippet?

```
public class Tester {

      public static void main(String[] args) {
            // TODO Auto-generated method stub
            Tester obj = new Tester();
            LinkedList linkedList = new LinkedList();
            linkedList.addAtEnd(1);
            linkedList.addAtEnd(4);
            linkedList.addAtEnd(6);
            linkedList.addAtEnd(7);
            linkedList.addAtEnd(9);
            obj.arrangeElements(linkedList.getHead(),5);
            linkedList.display();
      }
      public void arrangeElements(Node head, int num)
      {
            Node temp = head;
            int index,temp1,val;
            if(head==null)
            {
                  return;
            }
            else
            {
                  index = 1;
                  while(index < num)
                  {
                        val = temp.getData();
                        temp = temp.getNext();
                        head.setData(val);
                        index +=1;
                  }
                  temp1 = head.getData();
                  temp.setData(temp1);
            }
      }
}
```

Answer: 7 4 6 7 7

4. Consider a garage which is only one car wide but it is long enough to park many cars. A red car is first parked in the garage. After 20mins a blue car is

parked followed by a white car. In order to remove red car and blue car need to be removed in the respective order. Which data structures can be best illustrated using this scenario?

Answer: Stack

5. Consider the following code Snippet

```
class Computation {
      public int add(int num1,int num2)
      {
            return num1+num2;
      }
      public int divide(int num1,int num2)
      {
            return num1/num2;
      }
}
public class TestComputation
{
      Computation comput = new Computation();
      @Test
      public void testAdd()
      {
            int expected = 5;
            int actual = comput.add(2, 3);
            //Line 1
      }
      @Test
      public voiid testDivide1()
      {
            int expected = 7;
            int actual = comput.divide(2, 5);
            //line 2
      }
}
```

What should be written at Line 1 and Line 2 such that both the test cases pass when the code is executed with the default Junit runner?

Answer:

```
Assert.assertEquals(expected,actual)
Assert.assertNotEquals(expected,actual)
```

6. What will be the output of the following code Snippet

```
public class Tester {

      public static void main(String[] args)
      {
            int numOne = 4;
            int numTwo= -4;
            int result =0;
            for(int index=0; index <numOne; index++)
            {
                  if(index % 2 !=0)
                  {
                        result +=index+numOne;
                  }
                  else
                  {
                        result+=index+numTwo;
                  }
            }
            System.err.println(result);
      }
}
```

Answer: 6

7. Consider the following code

```
public class Demo2 {
      public static void main(String args[])
      {
            String ten="10";
            Integer value1 = Integer.parseInt(ten);
            Integer value2 = new Integer(ten);
            Integer value3;
            if(value1)
            {
                  System.out.println(value2==value3);
            }
      }
}
```

Answer: The code will result in compilation error

8. Consider the below code

```
class Movie {
    private String[] availMoviesArr = { "Tarzan", "Transformers", "Avengers" };
    private double[] moviePriceArr = { 300.0, 250.0, 220.0 };

      public void calculateTotalPrice(String[] reqMovieArr) {
        try {
            double price = 0.0;
            for (int index1 = 0; index1 <= reqMovieArr.length; index1++) {
                for (int index2 = 0; index2 <= availMoviesArr.length; index2++)
{
                    price += moviePriceArr[index2];
                    break;
                }
            }
        } catch (ArithmeticException e) {
            System.out.println("Arithmetic Exception occurred");
        } catch (ArrayIndexOutOfBoundsException e) {
            System.out.println("Array index out of bounds occurred");
        } catch (NullPointerException e) {
            System.out.println("Null pointer exception occurred");
        } catch (Exception e) {
            System.out.println("Some error occurred");
        } finally {
            System.out.println("Exiting calculateTotalPrice method.");
        }
    }
}

public class tester2 {
    public static void main(String[] args) {
        String[] reqMovieArr = { "Avengers", "Tarzan" };
        Movie movies = new Movie();
        movies.calculateTotalPrice(reqMovieArr);

    }
}
```

Answer: Array index out of bounds occurred.

9. Consider the below code

```
public class Tester1 {

      public static void main(String[] args)
      {
            int num1=4;
            int sum=0;
            for(int index=0;index<=num1;index++)
            {
                  sum=sum+num1;
            }
            System.out.println(index);
            System.out.println(sum);
      }
}
```

Answer: Compilation error

10. How many reference variables and objects will exist after Line#1 in the given code snippet

```
class Client
{
      String clientName;
      //getters and setters
}
public class Tester1 {

      public static void main(String[] args)
      {
            Client clientOne;
            clientOne = new Client();
            Client clienttwo = new Client();
            new Client();
            clientOne=null;//line #1
      }
}
```

Answer: Reference Variables: 2
Objects: 1

11.What will be the output of the following code snippet

```
abstract class OnlinePaymentService
{
      private String itemName;
      private double amount;
      public OnlinePaymentService(String itemName, double amount)
      {
            this.itemName=itemName;
            this.amount=amount;
      }
      String getItemName()
      {
            return this.itemName;
      }
      public double getAmount()
      {
            return this.amount;
      }
      public void display()
      {
            System.out.println("Online Payment Service class");
      }
      abstract double computeBill();
}
```

```
class PayPal extends OnlinePaymentService
{
      public int discount;
      public PayPal(String itemName, double amount,int discount) {
            super(itemName, amount);
            this.discount=discount;
            // TODO Auto-generated constructor stub
      }
      @Override
      public double computeBill()
      {
            double initialAmount = this.getAmount();
            double finalAmount = initialAmount+(initialAmount*discount/100.0);
            return finalAmount;
      }
}
class GooglePay extends OnlinePaymentService
{
      private int cashBackAmounnt;
      public GooglePay(String itemName,double amount,int cashBackAmount)
      {
            super(itemName, amount);
            this.cashBackAmounnt=cashBackAmount;
      }
}
public class Tester1 {

      public static void main(String[] args)
      {
            PayPal objPal = new PayPal("chocolate", 75, 5);
            objPal.display();
            System.out.println(objPal.getItemName()+","+objPal.computeBill());
      }
}
```

Answer:
Online Payment Service class
chocolate,78.75

12. Select the correct sequence of operations performed by JVM from the following options.
Answer: Loads ByteCode -> verifies bytecode -> converts bytecode to machine code ->Executes the code

13. Consider the following array of elements
5,7,8,3,1,9,4
How many merge operations are required to sort the given elements in ascending order using merge sort algorithm?

Answer: 6

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

```
public class Test {
      public double mtdDivision(float num1,int num2)
      {
            System.out.println("Inside float division");
            return(num1/num2);
      }
      public int mtdDivision(char num1,int num2)
      {
            System.out.println("Inside Integer division");
            return(num1/num2);
      }
      public static void main(String args[])
```

```
	{
		double num1=10.5f;
		int num2=4;
		Test objOne= new Test();
		double result = objOne.mtdDivision((float)num1, num2);
		System.out.println("Result"+result);
	}
}
```

Answer:
Inside float division
Result 2.625

15. What will be the output

```
public class Demo {
	public static void main(String args[])
	{
		Demo obj=new Demo();
		System.out.println(obj.test(4,"Sam"));
	}
	public String test(int num,String name)
	{
		if(num==0)
		{
			return name;
		}
		else if(name.equals(name+"43"))
		{
			num=num+1;
			return test(num,name);
		}
		else
		{
			name = name +""+num;
			num=num-1;
			return test(num,name);
		}
	}
}
```

Answer: Sam4321

16.What will be the output of the below code

```
class Resort
{
	public static String resortId;
	static {
		System.out.println("Initializing resort Id");
		resortId="Cr1900";
	}
}
class ClubResort extends Resort
{
	public static int noOfRooms;
	public static String rating;
	static
	{
		System.out.println("Initializing no of rooms");
		noOfRooms=10;
	}
	public String identifyResortRating()
	{
		if(Resort.resortId.substring(0, 2).equalsIgnoreCase("CR"))
		{
```

```
                  ClubResort.rating="A+";
            }
            else
            {
                  ClubResort.rating="B+";
            }
            return(ClubResort.rating);
      }
}
public class Tester3 {
      static
      {
            System.out.println("Welcome to resort");
      }
      public static void main(String[] args)
      {
            ClubResort club = new ClubResort();


System.out.println(ClubResort.resortId+","+ClubResort.noOfRooms+","+club.identif
yResortRating());
      }

}
```

Answer:-

```
Welcome to resort
Initializing resort Id
Initializing no of rooms
Cr1900,10,A+
```

17. What will be the output of the following code snippets?

```
class Base
{
      private static int courseId=0;
      protected static int studentId=0;
      public Base()
      {
            courseId++;
      }
      protected void display()
      {
            System.out.println(courseId+" ");
      }
}
class subClass extends Base
{
      private int allocationId=0;
      public subClass()
      {
            studentId++;
            allocationId++;
      }
      public void display()
      {
            System.out.println(allocationId);
      }
}
public class Demo5 {
      public static void main(String args[])
      {
            Base childOne = new subClass();
            subClass childTwo = new subClass();
            childOne.display();
      }
}
```

Answer: 1

18. Ron have been asked to create an online shopping application where customers can purchase items and keep them in their shopping cart. To implement this scenario, he has created Customer,Item and ShoppingCart classes. Which of the following relationships is most suitable for the scenario?
a)Customer uses ShoppingCart and ShoppingCart uses Item
b)Customer uses ShoppingCart and ShoppingCart has Item
c)Customer has ShoppingCart and ShoppingCart has Item
d)Customer has ShoppingCart and ShoppingCart uses Item

Answer: Customer has ShoppingCart and ShoppingCart has Item

19.Consider the below Code

```
public class Tester4 {
      public static void main(String args[])
      {
            Queue inQueue1 = new Queue(3);
            Queue inQueue2 = new Queue(3);
            for(int counter1=0; counter1 <=2;counter1++)
            {
                  inQueue1.enqueue(counter1*3);
            }
            for(int counter2=0; counter2 <=2;counter2++)
            {
                  if(counter2==2)
                  {
                        break;
                  }
                  inQueue2.enqueue(inQueue1.dequeue());
            }
            inQueue2.enqueue(12);
            System.out.println("inQueue1:");
            inQueue1.display();
            System.err.println("inQueue2:");
                        inQueue2.display();
      }
}
```

Assumption: All the references to the necessary files are available
What will the values of inQueue1 and inQueue2 after the given code is executed?

Answer: inQueue1: 6 and inQueue2:0,3,12

20. Identify the incorrect statements from the following. choose all that apply.
a)Protected data members can't be accessed by a non-child class in a different package
b)Default data members can be accessed by child classes in the same package
c)Protected data members can't be accessed by non-child class in the same package
d)Default data members can be accessed by child classes in the different package.

Answer: C,D

===

DSA ONLY

Q70

[2 marks]

Consider a queue of integers inQueue and a stack of integers inStack both implemented using the inbuilt ArrayDeque.
following elements:

inQueue (front -> rear): [2, 14, 3]

inStack (top -> bottom): [2, 2, 7, 5]

What will be the content of outStack (top -> bottom) if inQueue and inStack are passed as input parameters to the b

```
public static ArrayDeque<Integer> findNumbers(ArrayDeque<Integer> inQueue,
ArrayDeque<Integer> inStack){
    ArrayDeque<Integer> outStack = new ArrayDeque<Integer>();
    for(Integer num: inQueue){
        Integer sum = 0;
        while (inStack.peek()!=null){
            sum += inStack.pop();
            if (sum == num){
                outStack.push(num);
                break;
            }
        }
        if (sum < num){
            outStack.pop();
        }
    }
    return outStack;
}
```

outStack(top ->bottom): [14, 2]

outStack(top ->bottom): [2]

outStack(top ->bottom): [2, 14]

outStack(top ->bottom): [14]

Ans=outStack(top ->bottom): [2]

Q71]

[2 marks]

Consider the following ArrayList inArrayList consisting of Integers:

inArrayList: [10, 20, 30, 40, 50, 60, 70, 80]

What will be the content of outList if inArrayList is passed as a parameter to the below method?

```
public static ArrayList<Integer> getModifiedList(ArrayList<Integer> inArrayList)
{
    ArrayList<Integer> outList = new ArrayList<>();
    ArrayList<Integer> tempList = new ArrayList<>();
    for(int index=0;index<inArrayList.size();index++){
        if(inArrayList.get(index)%(index+1)==0){
            int temp = inArrayList.remove(index);
            tempList.add(temp);
        }
    }
    ListIterator<Integer> iter = tempList.listIterator();
    while(iter.hasNext()){
        outList.add(iter.next());
        if(iter.nextIndex()%3==0){
            iter.remove();
        }
        iter.next();
    }
      outList.addAll(tempList);
      return outlist;
}
```

outList: [10, 30, 60, 80]

outList: [10, 60]

outList: [10, 60, 10, 30, 80]

Runtime Exception: ConcurrentModificationException

Ans=outList: [10, 60, 10, 30, 80]

Q72]

Q24
[2 marks]
Given the below inIntLinkedList:
inIntLinkedList (Head to Tail) : 10->40->17->6->24->38
Consider the function processLinkedList given below:

```
public void processLinkedList(LinkedList inIntLinkedLst){
      Node head=inIntLinkedLst.getHead();
      Node tail=inIntLinkedLst.getTail();
      int tempSum=0;
      while(head.getNext().getNext() != null){
            if(head.getData()%5 == 0){
                  tempSum+=head.getData();
            }
            else{
                  tempSum -= head.getNext().getData();
            }
            head=head.getNext();
      }
      inIntLinkedLst.delete(tail.getData());
      inlntLinkedLst.addAtBeginning(tail.getData());
      System.out.println("sum:" +tempSum);
}
```

What will be the output of the function processLinkedList when inIntLinkedList is passed as the input parameter?
Assumption: LinkedList class, with the necessary methods, is available

sum: 20
inIntLinkedList (Head to Tail): 24 10 40 17 6 38

sum: 20
inIntLinkedList (Head to Tail): 38 10 40 17 6 24

sum: - 27

```
inIntLinkedList (Head to Tail): 38 10 40 17 6 24

sum: 6
inIntLinkedList (Head to Tail): 10 38 40 17 6 24
```

Ans=2

Q73]

Q29

```
import java.util.ArrayDeque;

public class QueueConverter {

    public static ArrayDeque<Integer> convertStructure(ArrayDeque<String>
inQueue) {
        ArrayDeque<Integer> outArrayDeque = new ArrayDeque<>();
        while (!inQueue.isEmpty()) {
            String str = inQueue.remove();
            outArrayDeque.push(str.length());
            if (str.length() > inQueue.peek().length()) { // Corrected: Using
peek() to check next element without removing
                outArrayDeque.offer(inQueue.remove().length());
            }
        }
        if (outArrayDeque.peek() > 5) {
            outArrayDeque.remove();
        }
        return outArrayDeque;
    }

    public static void main(String[] args) {
        ArrayDeque<String> inQueue = new ArrayDeque<>();
        inQueue.offer("George");
        inQueue.offer("Larry");
        inQueue.offer("Martin");
        inQueue.offer("Melissa");
        inQueue.offer("Karen");
        inQueue.offer("Caley");

        ArrayDeque<Integer> outQueue = convertStructure(inQueue);
        System.out.println(outQueue);
    }
}
```

Ans=The program throws a NullPointerException at runtime

Q74]

Q30
[2 marks]
Consider a queue of strings inQueue implemented using the inbuilt ArrayDeque class. inQueue contains the following elements:
inQueue(front->rear): ["George", "Larry", "Martin", "Melissa", "Karen", "Caley"]
What will be the content of outArrayDeque(front -> rear) if inQueue is passed as an input parameter to the below method?

```
public static ArrayDeque < Integer> convertStructure(ArrayDeque <String>
inQueue) {
      ArrayDeque<Integer> outArrayDeque = new ArrayDeque<>();
      while(!inQueue.isEmpty() {
            String str = inQueue.remove();
            outArrayDeque.push(str.length());
            if(str.length() != inQueue.remove().length()) {
                  outArrayDeque.offer(inQueue.remove().length());
            }
      }
      if(outArrayDeque.peek()>5){
            outArrayDeque.remove();
      }
      return outArrayDeque;
}
```

outArrayDeque(front -> rear): [7, 6, 6]
outArrayDeque(front -> rear): [7, 6, 6, 5]
outArrayDeque(front -> rear): [6, 6, 5]
outArrayDeque(front -> rear): [6, 6, 7]

Ans=C , outArrayDeque(front -> rear): [6, 6, 5]

Q75]

Q31
[3 marks]

Consider an ArrayDeque inArrayDeque and a Queue inQueue containing the following elements:

inArrayDeque (front->rear): [1, 2, 3, 4, 5]

inQueue (front->rear): ["Apple", "Bat", "Cotton", "Desert", "English"]

What will be the content of the outStack (top->bottom) if inArrayDeque and inQueue are passed as parameters to the below method?

```
public Stack convertToStack(ArrayDeque<Integer> inArrayDeque, Queue inQueue) {
      List<Integer> interList = new ArrayList<>();
      Stack outStack = new Stack(5);
      do {
            interList.add(inArrayDeque.remove() + inArrayDeque.size());
      } while (inArrayDeque.poll() != null);
      for (int num : interList) {
            if (!inQueue.isEmpty()) {
                  String str = inQueue.dequeue();
                  outStack.push(num + str.length());
            } else {
                  outStack.push(num);
            }
      }
      outStack.display();
      return outStack;
}
```

Note: 'Stack' and 'Queue' refer to the user defined classes discussed as a part of the course.
outStack (top->bottom): [11, 8, 10]
Exception thrown: java.util.NoSuchElementException
outStack (top->bottom): [10, 8, 11]
outStack (top->bottom): [7, 8, 14]

Ans=outStack (top->bottom): [11, 8, 10]

Q76]

Q32
[2 marks]
Consider a HashMap inHashMap which has string elements as key and integer elements as value and an array of strings inArray consisting of the following values:
inHashMap: {Cut=3, Roman=5, Car=3, Yard=6}
inArray: {"Car", "Lower", "Yard", "Cat", "Milo"}

What will be the content of outHashMap if inHashMap and inArray are passed as input parameters to the below method?

```
public static HashMap<Integer,String> formHashMap(HashMap<String,Integer>
inHashMap, String[] inArray){
      HashMap<Integer,String> outHashMap= new HashMap<Integer,String>();
      for (String word : inArray){
            boolean flag =false;
            for (Map.Entry<String,Integer> entry: inHashMap.entrySet(){
                  if (word.equals(entry.getKey())){

                        outHashMap.put(entry.getValue(),entry.getKey());
                        flag=true;
                        break;
                  }
            }if(!flag)

            {
                  outHashMap.put(word.length(), word);
            }
      }return outHashMap;
      }
```

outHashMap: {3=Cat, 5= Roman, 6=Yard}
outHashMap: {3=Cat, 4=Milo, 5= Lower, 6=Yard}
outHashMap: {3=Car, 4=Yard, 5= Lower}

Ans=2

Q77]

[2 marks]

Consider the following ArrayList inArray consisting of Strings:

inArray: ["25", "14", "5", "62", "79"]

What would be the content of outArray if the inArray and numStr='2' are passed as parameters to the below method?

```
public static ArrayList<String> parseArray(ArrayList<String> inArray, char
numStr){
    ArrayList<String> outArray = new ArrayList<String>();
    for(String elem: inArray){
        Boolean flag = true;
        for (int index=0; index<elem.length(); index++){
            if (elem.charAt(index)==numStr){
                outArray.add(0, elem + ":" + index);
                flag=false;
                break;
            }
        }
        if (flag){
            outArray.add(elem);
        }
    }
return OutArray;
}
```

outArray: [25:0, 62:1, 14, 5, 79]

outArray: [79, 5, 14, 25:0, 62:1]

```
outArray: [62:1, 25:0, 14, 5, 79]

outArray: [79, 5, 14, 62:1, 25:0]

Ans= ["62:1", "25:0", "14", "5", "79"]  //not 100%
-------------------------------------------------------------------------
```

Q78]

Consider the below code:

```
class Tester

{

        public static void main(String args[]){                  //Solve
        Queue inIntQueue1 = new Queue(3);
        Queue inIntQueue2 = new Queue(3);
        for(int counter1 =0;counter1<=2;counter1++){
                inIntQueue1.enqueue(counter1*3);
        }
        for(int counter2=0;counter2<=2;counter2++) {
                if(counter2 == 2){
                        break;
                }
                inIntQueue2.enqueue(inIntQueue1.dequeue());
        }
        inIntQueue2.enqueue(12);
System.out.println("inIntQueue2 elements are:");

inIntQueue2.display();
```

Assumption: All the references to the necessary files are available

What is the status of inIntQueue1 and inIntQueue2 after the execution of the above code?

In the options, consider the elements of the queue from Front to Rear.

* inIntQueue1: 0, 3, 6 and inIntQueue2: 0, 3, 12
* inIntQueue1: 6 and inIntQueue2: 0, 3
* inIntQueue1: 6 and inIntQueue2: 0, 3, 12

Ans=B

==
Q79]
[2 marks]

Consider the following function reverseLinkedList, which is supposed to reverse a linked list containing unique elements:

```
public LinkedList reverseLinkedList(LinkedList originalList){
    LinkedList reverseList=new LinkedList();
    Node temp1=originalList.getHead();
    Node temp2=originalList.getTail();
    int count=0;
    while(temp1!=null){
        count+=1;
        temp1=temp1.getNext();
    }
    while(count>0){
        temp2=originalList.getTail();
        ________________________//Line1
        ________________________//Line2
        count-=1;
    }
    originalList=reverseList;
    return originalList;
}
```

Which of the following options must be added at Line1 and Line2 so that the code performs correctly?
Assumption: LinkedList class, with the necessary methods, is available

Notes:

* Line numbers are only for reference
* "LinkedList" refers to the user defined class discussed as a part of the course

```
Line 1: reverseList.addAtEnd(temp2.getData());
Line 2: originalList.delete(temp2.getData());
```

```
Line 1: originalList.addAtEnd(temp2.getData());
Line 2: reverseList.delete(temp2.getData());
```

```
Line 1: reverseList.addAtEnd(temp2);
Line 2: originalList.delete(temp2);
```

```
Line 1: originalList.addAtEnd(temp2);
Line 2: reverseList.delete(temp2);
```

Ans=1

===========SELF ASSESSMENT QUESTION====================

```
class Tester{

public static void main(String args[]){
    char arr[]=new char[4];
    arr[0]='A';
    arr[1]='S';
    arr[2]='D';
    arr[3]='F';

    ArrayTest.insert(arr, 4, 'J');

    for(int index=0; index<arr.length; index++)
        System.out.println(arr[index]);
}
}
```

ArrayTest class is given below.

```
class ArrayTest {

public static void insert(char[] ar, int pos, char val){

    for(int index=ar.length-1; index>=pos; index--) {
        ar[index]=ar[index-1];
    }
    ar[pos-1]=val;
}
}
```

ANSWER:ASDJ

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------------
```

Q60

Q2 of 20

What will be the output of the code given below?

Assumption: LinkedList class is already implemented with all the required methods.

```
public class Tester {

public static void main(String args[]) {
    LinkedList list = new LinkedList();
    list.addAtEnd("11");
    list.addAtEnd("13");
    list.addAtEnd("18");
    list.addAtEnd("34");
    list.addAtEnd("46");
    operate(list);
    list.display();
}

public static void operate(LinkedList list) {
    Node temp = list.getHead();
    while (temp.getNext().getNext() != null) {
        temp.setData(temp.getNext().getData());
        temp = temp.getNext();
    }
}
}
```

ANSWER:11,13,18,34,46

```
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------------
```

Q61

Q3 of 20

What will be the output of the code given below?

Assumption: Stack class is already implemented with all the required methods.

```
public class Tester {
public static void main(String args[]) {
Stack stack = new Stack(10);
stack.push(18);
stack.push(10);
stack.push(24);
stack.push(56);
stack.push(27);
operate(stack);
stack.display();
}

public static void operate(Stack stack) {
    for (int i = 0; i <= 2; i++) {
        if (stack.pop() % 3 == 0) {
            int temp = stack.pop();
            stack.push(++temp);
            stack.push(++temp);
        }
    }
}
}
```

ANSWER:26,25,10,18

Q4 of 20

What will be the output of the code given below?

Assumption: Stack class is already implemented with all the required methods.

```
public class Tester {
public static void main(String args[]) {
Stack stack = new Stack(10);
stack.push(11);
stack.push(19);
stack.push(18);
stack.push(20);
stack.push(15);
stack.push(13);
stack.push(17);
System.out.println(operate(stack));
}

public static int operate(Stack stack) {
    int value = 0;
    while (!stack.isEmpty()) {
        if (stack.peek() % 2 != 0) {
            value += stack.pop();
            stack.pop();
        } else {
            stack.pop();
        }
    }
    return value;
}
}
```

ANSWER: 51

Q5 of 20

What will be the output of the code given below?

Assumption: Queue class is already implemented with all the required methods.

```
public class Tester {
public static void main(String args[]) {
Queue queue = new Queue(10);
operate(queue);
queue.display();
}

public static void operate(Queue queue) {
    int[] numbers = {12, 18, 17, 16, 28, 34, 36};
    int count = 6;
    for (int number : numbers) {
        if (count == 0) {
            break;
        }
        if (number % count == 0) {
            queue.enqueue(number);
        }
        --count;
    }
}
}
```

Q9 of 20

Choose the correct option based on the execution of the code given below.

```
public class Tester {

public static void main(String args[]) {
    List<Integer> elements = new LinkedList<Integer>();
    elements.add(10);
    elements.add(12);
    elements.add(33);
    elements.add(44);
    elements.add(75);
    elements.add(67);
    int temp = 0;
    int sum = 0;
    for (int element : elements) {
        temp = element;
        while (temp != 0) {
            sum += temp % 10;
            temp = temp / 10;
        }
        if (sum % 2 == 0) {
            System.out.println("Infosys");
        }
    }
}
}
```

Q11 of 20

What will be the output of the code given below?

```
public class Tester {
    public static void main(String[] args) {
        Set<String> treeSet = new TreeSet<String>();
        treeSet.add(new String("A"));
        treeSet.add(new String("B"));
        treeSet.add(new String("C"));
        treeSet.add(new String("C"));
        treeSet.add(new String("E"));
        treeSet.add(new String("D"));
        treeSet.add(new String("a"));
        treeSet.add(new String("F"));
        Object[] elements = treeSet.toArray();
        for (Object element : elements)
            System.out.print(element + " ");
    }
}
```

Ans=ABCDEFa

Q12 of 20

Choose the correct option based on the execution of the code given below.

```
public class Tester {

    public static void main(String args[]) {
        Map<String, Integer> studentDetails = new HashMap<String, Integer>();
        studentDetails.put("Max", 337);
        studentDetails.put("Stocks", 480);
        studentDetails.put("Malinda", 570);
        studentDetails.put("Mathew", 640);
        studentDetails.put("Max", 340);
        if (studentDetails.replace("stocks", 480, 650)) {
            studentDetails.remove("Max");
        } else {
            studentDetails.put("Sam", 490);
        }
        System.out.println(studentDetails);
    }
}
```

```
{Max=340, Stocks=480, Malinda=570, Mathew=640, Sam=490}
```


Q13 of 20

What will be the output of the code given below?

```
public class Tester {

    public static void main(String args[]) {
        Map<Integer, Integer> hashMap = new HashMap<Integer, Integer>();
        for (int counter1 = 0; counter1 <= 5; counter1++) {
            for (int counter2 = 5; counter2 >= 1; counter2--) {
                hashMap.put(counter1, counter2);
            }
        }
        System.out.println(hashMap);
    }
}
```

```
{0=1, 1=1, 2=1, 3=1, 4=1, 5=1}
```


```
---------------
```

Q14 of 20

Choose the correct option based on the execution of the code given below.

```
public class Tester {

    public static void main(String[] args) {
        Deque<String> brands = new ArrayDeque<String>();
        brands.add("Apple");
        brands.add("Samsung");
        brands.add("One Plus");
        brands.add("Nokia");
        brands.add("Blueberry");
        brands.poll();
        brands.add("Microsoft");
        brands.element();
        brands.peek();
        brands.remove();
        for (String brand : brands) {
            System.out.println(brand);
        }
    }
}
One Plus
Nokia
Blueberry
Microsoft
--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------------
```

Q10 of 20

What will be the output of the code given below?

```
public class Tester {

    public static void main(String[] args) {
        Set<String> linkedHashSet = new LinkedHashSet<String>();
        linkedHashSet.add(new String("A"));
        linkedHashSet.add(new String("B"));
        linkedHashSet.add(new String("C"));
        linkedHashSet.add(new String("C"));
        linkedHashSet.add(new String("E"));
        linkedHashSet.add(new String("D"));
        linkedHashSet.add(new String("E"));
        linkedHashSet.add(null);
        linkedHashSet.add(new String("E"));
        Object[] elements = linkedHashSet.toArray();
        for (Object element : elements)
            System.out.print(element + " ");
    }
}

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------------
--------------------------------------------------------------------------------
-------------------------
```

Q7 of 20

What will be the output of the code given below?

```
public class Tester {

    public static void main(String[] args) {
        List<String> employees = new ArrayList<String>();
        employees.add("Alex");
        employees.add("Tom");
        employees.add("Sam");
        employees.add("john");
        employees.add("Jack");
        updateEmployee(employees);
        for (String employee : employees) {
            System.out.print(employee + " ");
        }
    }

    public static void updateEmployee(List<String> employees) {
        String[] newEmployees = {"John", "Jack", "Robert", "Steve"};
        for (int counter = 0; counter <= newEmployees.length - 1; counter++) {
            if (!employees.contains(newEmployees[counter])) {
                employees.add(counter + 1, newEmployees[counter]);
            }
        }
    }
}
```


Q8 of 20

What will be the output of the code given below?

```
public class Tester {

    public static void main(String args[]) {
        List<Integer> elements = new LinkedList<Integer>();
        elements.add(1);
        elements.add(2);
        elements.add(3);
        elements.add(4);
        elements.add(5);
        elements.add(6);
        elements.remove(1);
        elements.add(3, 34);
        elements.set(5, 15);
        System.out.println(elements);
    }
}
```


Q9 of 20

Choose the correct option based on the execution of the code given below.

```
public class Tester {

    public static void main(String args[]) {
        List<Integer> elements = new LinkedList<Integer>();
        elements.add(10);
        elements.add(12);
        elements.add(33);
        elements.add(44);
        elements.add(75);
        elements.add(67);
        int temp = 0;
        int sum = 0;
        for (int element : elements) {
            temp = element;
            while (temp != 0) {
                sum += temp % 10;
                temp = temp / 10;
            }
            if (sum % 2 == 0) {
                System.out.println("Infosys");
            }
        }
    }
}

--------------------------------------------------------------------------------
--------------------------------------------------------------------------------
---------------
```