1.

```java
import java.util.Scanner;


public class StudentGrading {
     public static void main(String[] args) {
        Scanner scan = new Scanner(System.in);
        System.out.print("Enter the 5 marks : ");
        int m1 = scan.nextInt();
        int m2 = scan.nextInt();
        int m3 = scan.nextInt();
        int m4 = scan.nextInt();
        int m5 = scan.nextInt();
        int tot = m1+m2+m3+m4+m5;
        int total = tot/5;
        if((total>=91)&&(total<=100)){System.out.println("Outstanding Grade 10");}
        else if ((total>=81)&&(total<=90)){System.out.println("Excellent Grade 9");}
        else if ((total>71)&&(total<=81)){System.out.println("Very Good Grade 8");}
        else if ((total>=61)&&(total<=70)){System.out.println("Good Grade 7");}
        else if ((total>=50)&&(total<=60)){System.out.println("Average Grade 6");}
        else if ((total<50)){System.out.println("RA Grade 0");}
        else{System.out.print("Invalud Entry");}
        scan.close();
    }
}
```

2

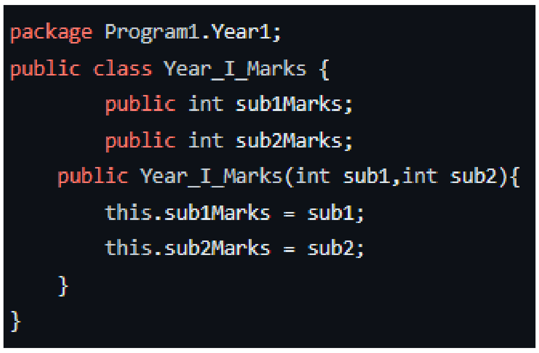
a)

```java
class complex{
    private double real;
    private double imaginary;

    public complex(double real,double imaginary){
        this.real =real;
        this.imaginary =imaginary;
    }

    public void add(complex other){
        this.real+=other.real;
        this.imaginary+=other.imaginary;
    }

    public void sub(complex other){
        this.real-=other.real;
        this.imaginary-=other.imaginary;
    }

    public void mul(complex other){
        double tempreal = this.real*other.real - this.imaginary*other.imaginary;
        double tempimaginary = this.real*other.imaginary + other.real*this.imaginary;
        this.real = tempreal;
        this.imaginary = tempimaginary;
    }
    public void div(complex other){
        double divisor = other.real*other.real + other.imaginary*other.imaginary;
        double tempreal = (this.real*other.real - this.imaginary*other.imaginary)/divisor;
        double tempimaginary = (this.imaginary*other.real + other.imaginary*this.real)/divisor;
        this.real = tempreal;
        this.imaginary = tempimaginary;
    }

    @Override
    public String toString(){
        if(imaginary>=0){
            return real + " + " + imaginary +"i";
        }
        else{
            return real + " - " + Math.abs(imaginary) +"i";
        }
    }

}

public class ComplexMain{
    public static void main(String[] args) {
        complex c1 = new complex(3,2 );
        complex c2 = new complex(3, -4);

        c1.add(c2);
        System.out.println("Addition: " + c1);

        c2 = new complex(3, -4);
        c1.sub(c2);
        System.out.println("Subtraction: " + c1);

        c1 = new complex(3, 2);
        c1.mul(c2);
        System.out.println("Multiplication: " + c1);

        c1 = new complex(3, 2);
        c1.div(c2);
        System.out.println("Division: " + c1);

    }
}
```

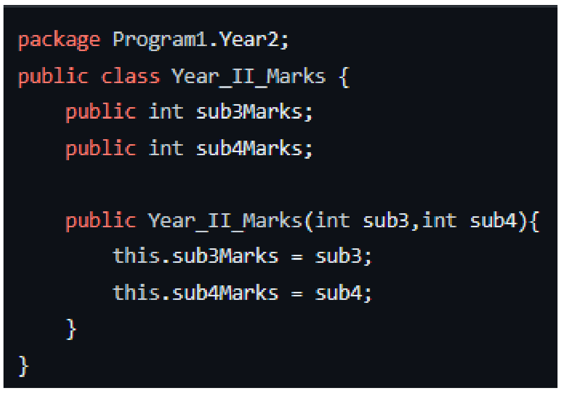
b)

```java
import java.util.Scanner;
public class DaysToMonths {
    public static void main(String[] args) {
        int days;
        Scanner obj = new Scanner(System.in);
        System.out.print("Enter the number of Days : ");
        days = obj.nextInt();
        int months,RDays;
        months = days / 30;
        System.out.print( months +" Months ");
        RDays = days - (months*30);
        System.out.println( RDays +" Days");
        obj.close();
    }

}
```

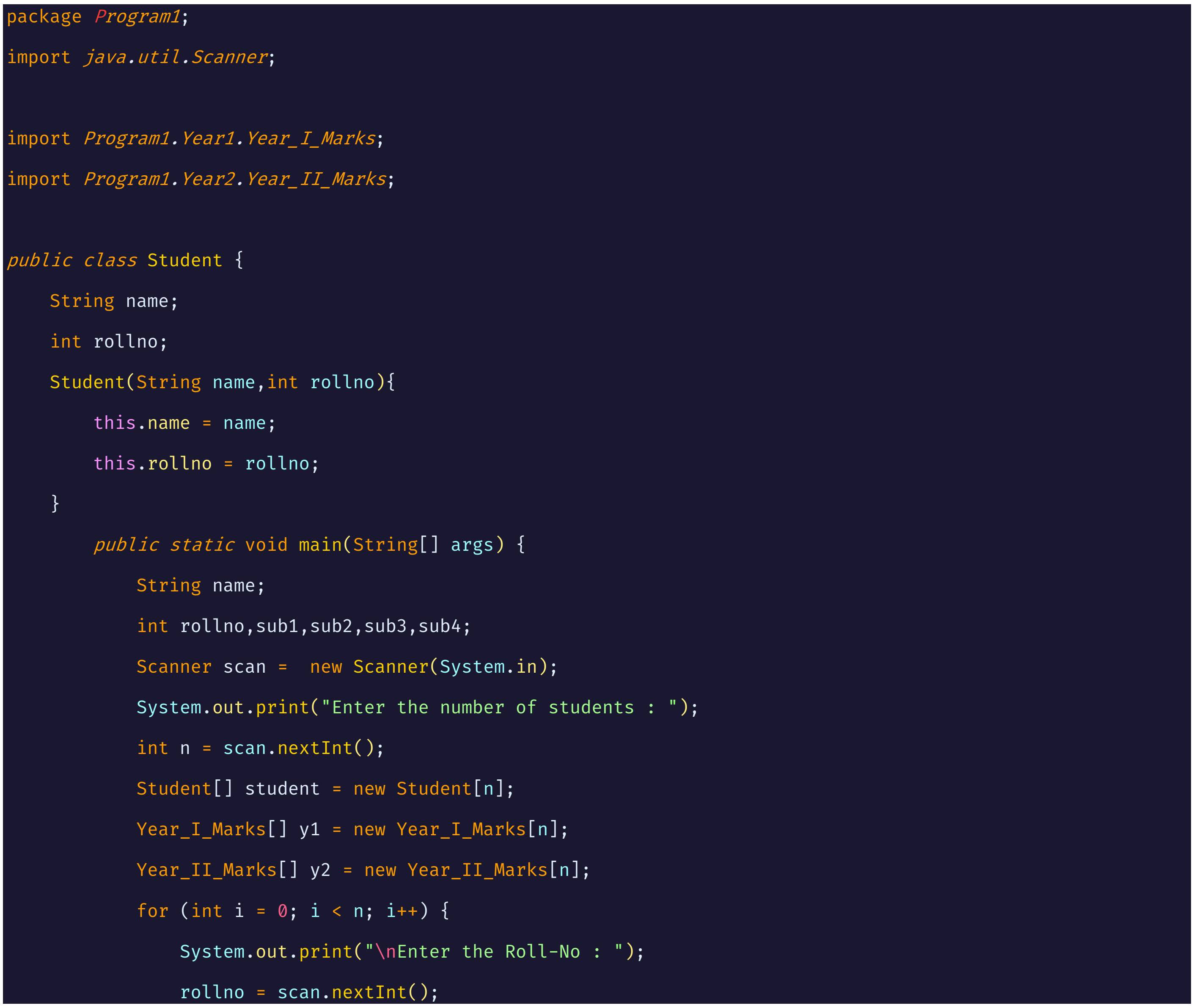
3)



```
package Program1.Year1;
public class Year_I_Marks {
        public int sub1Marks;
        public int sub2Marks;
    public Year_I_Marks(int sub1,int sub2){
        this.sub1Marks = sub1;
        this.sub2Marks = sub2;
    }
}
```



```
package Program1.Year2;
public class Year_II_Marks {
    public int sub3Marks;
    public int sub4Marks;

    public Year_II_Marks(int sub3,int sub4){
        this.sub3Marks = sub3;
        this.sub4Marks = sub4;
    }
}
```



```
package Program1;
import java.util.Scanner;

import Program1.Year1.Year_I_Marks;
import Program1.Year2.Year_II_Marks;

public class Student {
    String name;
    int rollno;
    Student(String name,int rollno){
        this.name = name;
        this.rollno = rollno;
    }
        public static void main(String[] args) {
            String name;
            int rollno,sub1,sub2,sub3,sub4;
            Scanner scan =  new Scanner(System.in);
            System.out.print("Enter the number of students : ");
            int n = scan.nextInt();
            Student[] student = new Student[n];
            Year_I_Marks[] y1 = new Year_I_Marks[n];
            Year_II_Marks[] y2 = new Year_II_Marks[n];
            for (int i = 0; i < n; i++) {
                System.out.print("\nEnter the Roll-No : ");
                rollno = scan.nextInt();
```

```java
                System.out.print("Enter the name : ");
                name= scan.next();
                System.out.print("Enter Year I marks : ");
                sub1 = scan.nextInt();
                sub2 = scan.nextInt();
                y1[i] = new Year_I_Marks(sub1,sub2);
                System.out.print("Enter Year II marks : ");
                sub3 = scan.nextInt();
                sub4 = scan.nextInt();
                y2[i] = new Year_II_Marks(sub3,sub4);
                student[i] = new Student(name,rollno);
            }

            for (int i = 0; i < n; i++) {
                System.out.println("\nName : "+ student[i].name+ "\nRoll : "+ student[i].rollno);
                System.out.println("Year I Marks  : "+ y1[i].sub1Marks
                                    + ((y1[i].sub1Marks > 50) ? "-Pass" : "-Fail")
                                    + "  " + y1[i].sub2Marks
                                    + ((y1[i].sub2Marks > 50) ? "-Pass" : "-Fail")
                                  );
                System.out.println("Year II Marks : "+ y2[i].sub3Marks
                                    + ((y2[i].sub3Marks > 50) ? "-Pass" : "-Fail")
                                    + "  " + y1[i].sub2Marks
                                    + ((y2[i].sub4Marks > 50) ? "-Pass" : "-Fail")
                                  );
            }
        }


}
```

4)

```java
// File structure:
// - com
//   - transact
//     - Transaction.java
//   - loan
//     - LoanAccount.java
//   - Main.java

// Transaction.java
package com.transact;

public class Transaction {
    public static void credit(double amount) {
        // Logic for crediting amount
        System.out.println("Credited amount: " + amount);
    }

    public static void debit(double amount) {
        // Logic for debiting amount
        System.out.println("Debited amount: " + amount);
    }
}

// LoanAccount.java
package com.loan;
import com.transact.Transaction;

public class LoanAccount {
    public void doTransaction(double amount, String transactionType) {
        if (transactionType.equalsIgnoreCase("credit")) {
            Transaction.credit(amount);
        } else if (transactionType.equalsIgnoreCase("debit")) {
            Transaction.debit(amount);
        } else {
            System.out.println("Invalid transaction type.");
        }
    }
}

// Main.java
package com;
```

```java
import com.loan.LoanAccount;

public class Main {
    public static void main(String[] args) {
        if (args.length < 2) {
            System.out.println("Usage: java com.Main <amount> <transactionType>");
            return;
        }

        double amount = Double.parseDouble(args[0]);
        String transactionType = args[1];

        LoanAccount loanAccount = new LoanAccount();
        loanAccount.doTransaction(amount, transactionType);
    }
}
```

5)

```java
// QueueOperations.java
public interface QueueOperations {
    void enqueue(int item);
    int dequeue();
    void display();
}

// MyQueue.java
public class MyQueue implements QueueOperations {
    private int[] queue;
    private int front;
    private int rear;
    private int capacity;

    public MyQueue(int size) {
        capacity = size;
        queue = new int[capacity];
        front = 0;
        rear = -1;
    }

    public void enqueue(int item) {
        if (rear == capacity - 1) {
            System.out.println("Queue is full. Cannot enqueue " + item);
        } else {
            queue[++rear] = item;
            System.out.println("Enqueued " + item);
        }
    }

    public int dequeue() {
        if (front > rear) {
            System.out.println("Queue is empty.");
            return -1;
        } else {
            int dequeuedItem = queue[front++];
            System.out.println("Dequeued " + dequeuedItem);
            return dequeuedItem;
        }
    }
    public void display() {
        if (front > rear) {
            System.out.println("Queue is empty.");
        } else {
            System.out.print("Queue: ");
            for (int i = front; i <= rear; i++) {
                System.out.print(queue[i] + " ");
            }
            System.out.println();
        }
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        MyQueue queue = new MyQueue(5);
        queue.enqueue(10);
        queue.enqueue(20);
        queue.enqueue(30);
        queue.display();
        queue.dequeue();
        queue.display();
        queue.enqueue(40);
        queue.enqueue(50);
        queue.enqueue(60); // This will show that the queue is full
        queue.display();
    }
}
```

6)

```java
// Student.java
public class Student {
    private String name;
    private int rollNo;

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

    public String getName() {
        return name;
    }

    public int getRollNo() {
        return rollNo;
    }
}

// Result.java
public class Result extends Student {
    private int subject1Marks;
    private int subject2Marks;
    private int subject3Marks;

    public Result(String name, int rollNo, int subject1Marks, int subject2Marks, int subject3Marks) {
        super(name, rollNo);
        this.subject1Marks = subject1Marks;
        this.subject2Marks = subject2Marks;
        this.subject3Marks = subject3Marks;
    }

    public int getTotalMarks() {
        return subject1Marks + subject2Marks + subject3Marks;
    }

    public String getResult() {
        int total = getTotalMarks();
        return (total >= 120) ? "Pass" : "Fail";
    }
}
```

```java
// Sports.java
public class Sports extends Result {
    private int sportsPoints;

    public Sports(String name, int rollNo, int subject1Marks, int subject2Marks, int subject3Marks, int
sportsPoints) {
        super(name, rollNo, subject1Marks, subject2Marks, subject3Marks);
        this.sportsPoints = sportsPoints;
    }

    public void displayResult() {
        System.out.println("Name: " + getName());
        System.out.println("Roll No: " + getRollNo());
        System.out.println("Total Marks: " + getTotalMarks());
        System.out.println("Result: " + getResult());
        System.out.println("Sports Points: " + sportsPoints);
        System.out.println("--------------------------------------");
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        Sports student1 = new Sports("Alice", 101, 80, 85, 75, 10);
        Sports student2 = new Sports("Bob", 102, 90, 95, 70, 5);
        Sports student3 = new Sports("Charlie", 103, 70, 65, 80, 15);

        student1.displayResult();
        student2.displayResult();
        student3.displayResult();
    }
}
```

7)

```java
import java.util.Scanner;

abstract class Car {
    private String regNo;
    private String model;
    private String regDate;

    public Car(String regNo, String model, String regDate) {
        this.regNo = regNo;
        this.model = model;
        this.regDate = regDate;
    }

    public abstract void displayDetails();
}

class TransportVehicle extends Car {
    private String validityNo;
    private String startDate;
    private String period;

    public TransportVehicle(String regNo, String model, String regDate, String validityNo, String startDate,
String period) {
        super(regNo, model, regDate);
        this.validityNo = validityNo;
        this.startDate = startDate;
        this.period = period;
    }

    public void displayDetails() {
        System.out.println("Transport Vehicle Details:");
        System.out.println("Registration No: " + super.regNo);
        System.out.println("Model: " + super.model);
        System.out.println("Registration Date: " + super.regDate);
        System.out.println("Validity No: " + validityNo);
        System.out.println("Start Date: " + startDate);
        System.out.println("Period: " + period);
        System.out.println("-----------------------------");
    }
}
```

```java
class PrivateVehicle extends Car {
    private String ownerName;
    private String ownerAddress;

    public PrivateVehicle(String regNo, String model, String regDate, String ownerName, String ownerAddress)
{
        super(regNo, model, regDate);
        this.ownerName = ownerName;
        this.ownerAddress = ownerAddress;
    }

    public void displayDetails() {
        System.out.println("Private Vehicle Details:");
        System.out.println("Registration No: " + super.regNo);
        System.out.println("Model: " + super.model);
        System.out.println("Registration Date: " + super.regDate);
        System.out.println("Owner Name: " + ownerName);
        System.out.println("Owner Address: " + ownerAddress);
        System.out.println("-----------------------------");
    }
}

public class Main {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter the number of vehicles: ");
        int n = scanner.nextInt();
        scanner.nextLine(); // Consume the newline character

        Car[] vehicles = new Car[n];

        for (int i = 0; i < n; i++) {
            System.out.println("Enter vehicle type (1 for Transport Vehicle, 2 for Private Vehicle): ");
            int choice = scanner.nextInt();
            scanner.nextLine(); // Consume the newline character

            System.out.print("Enter registration number: ");
            String regNo = scanner.nextLine();
            System.out.print("Enter model: ");
            String model = scanner.nextLine();
            System.out.print("Enter registration date: ");
            String regDate = scanner.nextLine();
            if (choice == 1) {
                System.out.print("Enter validity number: ");
                String validityNo = scanner.nextLine();
                System.out.print("Enter start date: ");
                String startDate = scanner.nextLine();
                System.out.print("Enter period: ");
                String period = scanner.nextLine();

                vehicles[i] = new TransportVehicle(regNo, model, regDate, validityNo, startDate, period);
            } else if (choice == 2) {
                System.out.print("Enter owner name: ");
                String ownerName = scanner.nextLine();
                System.out.print("Enter owner address: ");
                String ownerAddress = scanner.nextLine();

                vehicles[i] = new PrivateVehicle(regNo, model, regDate, ownerName, ownerAddress);
            } else {
                System.out.println("Invalid choice. Please enter 1 or 2.");
                i--; // Repeat the iteration
            }
        }

        System.out.println("\nPrivate Vehicles:");
        for (Car vehicle : vehicles) {
            if (vehicle instanceof PrivateVehicle) {
                vehicle.displayDetails();
            }
        }

        System.out.println("Transport Vehicles:");
        for (Car vehicle : vehicles) {
            if (vehicle instanceof TransportVehicle) {
                vehicle.displayDetails();
            }
        }
    }
}
```

8)

```java
// CreditCardInterface.java
public interface CreditCardInterface {
    double viewCreditAmount();
    int viewPin();
    void changePin(int newPin);
    void payBalance(double amount);
}

// Customer.java
public class Customer implements CreditCardInterface {
    private String name;
    private String cardNumber;
    private int pin;
    private double creditAmount;

    public Customer(String name, String cardNumber, int pin) {
        this.name = name;
        this.cardNumber = cardNumber;
        this.pin = pin;
        this.creditAmount = 0;
    }

    @Override
    public double viewCreditAmount() {
        return creditAmount;
    }

    @Override
    public int viewPin() {
        return pin;
    }

    @Override
    public void changePin(int newPin) {
        pin = newPin;
        System.out.println("Pin changed successfully for card number: " + cardNumber);
    }

    @Override
    public void payBalance(double amount) {
        creditAmount -= amount;
        System.out.println("Paid " + amount + " from card number: " + cardNumber);
    }
}

// Main.java
public class Main {
    public static void main(String[] args) {
        Customer[] customers = new Customer[5]; // Creating an array of 5 customers

        // Initializing customer objects
        customers[0] = new Customer("Alice", "1111 2222 3333 4444", 1234);
        customers[1] = new Customer("Bob", "2222 3333 4444 5555", 5678);
        customers[2] = new Customer("Charlie", "3333 4444 5555 6666", 9876);
        customers[3] = new Customer("David", "4444 5555 6666 7777", 4321);
        customers[4] = new Customer("Eve", "5555 6666 7777 8888", 1357);

        // Performing actions
        customers[0].payBalance(500);
        customers[1].changePin(7890);
    }
}
```

9)

```java
import java.util.ArrayList;
import java.util.Random;

public class ArraySeparation {
    public static void main(String[] args) {
        int[] array = new int[20];
        Random random = new Random();

        int sum = 0;
        for (int i = 0; i < array.length; i++) {
            array[i] = random.nextInt(81) + 10;
            sum += array[i];
        }

        double average = (double) sum / array.length;

        ArrayList<Integer> belowAverage = new ArrayList<>();
        ArrayList<Integer> aboveAverage = new ArrayList<>();

        for (int value : array) {
            if (value < average) {
                belowAverage.add(value);
            } else {
                aboveAverage.add(value);
            }
        }

        System.out.print("Below Average: ");
        for (int value : belowAverage) {
            System.out.print(value + " ");
        }
        System.out.println();

        System.out.print("Above Average: ");
        for (int value : aboveAverage) {
            System.out.print(value + " ");
        }
        System.out.println();
    }
}
```

10)

1)

```java
import java.util.Scanner;

public class EmailGenerator {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.print("Enter first name: ");
        String firstName = scanner.nextLine().trim();
        System.out.print("Enter last name: ");
        String lastName = scanner.nextLine().trim();
        String email = generateEmail(firstName, lastName);
        System.out.println("Generated email address: " + email);
    }

    public static String generateEmail(String firstName, String lastName) {
        String firstPart = firstName.substring(0, Math.min(firstName.length(), 3));
        String secondPart = lastName.substring(0, Math.min(lastName.length(), 4));
        return firstPart + "." + secondPart + "@example.com";
    }
}
```

2)

```java
abstract class Stack {
    protected int[] stackArray;
    protected int top;
    protected int capacity;

    public Stack(int capacity) {
        this.capacity = capacity;
        stackArray = new int[capacity];
        top = -1;
    }

    public abstract void push(int element);

    public abstract int pop();

    public boolean isFull() {
        return top == capacity - 1;
    }
    public boolean isEmpty() {
        return top == -1;
    }
}

class IntegerStack extends Stack {
    public IntegerStack(int capacity) {
        super(capacity);
    }

    @Override
    public void push(int element) {
        if (!isFull()) {
            stackArray[++top] = element;
            System.out.println("Pushed: " + element);
        } else {
            System.out.println("Stack overflow");
        }
    }

    @Override
    public int pop() {
        if (!isEmpty()) {
            int popped = stackArray[top--];
            System.out.println("Popped: " + popped);
            return popped;
        } else {
            System.out.println("Stack underflow");
            return -1;
        }
    }
}

public class Main {
    public static void main(String[] args) {
        IntegerStack stack = new IntegerStack(5); // Creating a stack of capacity 5

        stack.push(10);
        stack.push(20);
        stack.push(30);
        stack.push(40);
        stack.push(50);

        stack.push(60); // Trying to push when stack is already full

        stack.pop();
        stack.pop();
        stack.pop();
        stack.pop();
        stack.pop();

        stack.pop(); // Trying to pop when stack is empty
    }
}
```

11

a)

```
class HelloInputException extends Exception {
    public HelloInputException(String message) {
        super(message);
    }
}

public class ExceptionHandling {
    public static void main(String[] args) {
        try {
            String input = "hello";

            if (input.equalsIgnoreCase("hello")) {
                throw new HelloInputException("Exception: User input is 'hello'");
            } else {
                System.out.println("User input: " + input);
            }
        } catch (HelloInputException e) {
            System.out.println(e.getMessage());
        }
    }
}
```

b)

```
import java.util.InputMismatchException;
import java.util.Scanner;

public class ExceptionHandling {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        try {
            System.out.println("Enter first number: ");
            int num1 = scanner.nextInt();

            System.out.println("Enter second number: ");
            int num2 = scanner.nextInt();

            int sum = addNumbers(num1, num2);
            System.out.println("Sum: " + sum);
        } catch (InputMismatchException e) {
            System.out.println("Exception: Invalid input. Please enter valid integers.");
        }
    }

    public static int addNumbers(int num1, int num2) {
        return num1 + num2;
    }
}
```

12)

```java
class AgeNotWithinRangeException extends Exception {
    public AgeNotWithinRangeException(String message) {
        super(message);
    }
}

class NameNotValidException extends Exception {
    public NameNotValidException(String message) {
        super(message);
    }
}

class Doctor {
    private int id;
    private String name;
    private int age;
    private String department;

    public Doctor(int id, String name, int age, String department) throws AgeNotWithinRangeException,
NameNotValidException {
        if (age < 25 || age > 65) {
            throw new AgeNotWithinRangeException("Age should be between 25 and 65.");
        } else {
            this.age = age;
        }

        if (!name.matches("^[a-zA-Z\\s]+$")) {
            throw new NameNotValidException("Name should contain only letters and spaces.");
        } else {
            this.name = name;
        }

        this.id = id;
        this.department = department;
    }

    public int getId() {
        return id;
    }

    public String getName() {
        return name;
    }

    public int getAge() {
        return age;
    }

    public String getDepartment() {
        return department;
    }
}

public class Main {
    public static void main(String[] args) {
        try {
            int id = 101;
            String name = "John Doe";
            int age = 70;
            String department = "Cardiology";

            Doctor doctor = new Doctor(id, name, age, department);
            System.out.println("Doctor details:");
            System.out.println("ID: " + doctor.getId());
            System.out.println("Name: " + doctor.getName());
            System.out.println("Age: " + doctor.getAge());
            System.out.println("Department: " + doctor.getDepartment());
        } catch (AgeNotWithinRangeException | NameNotValidException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }
}
```

13)

```java
class PositiveIntegerException extends Exception {
    public PositiveIntegerException(String message) {
        super(message);
    }
}

class SecondIntegerLargerException extends Exception {
    public SecondIntegerLargerException(String message) {
        super(message);
    }
}

public class PrimeNumbers {
    public static void main(String[] args) {
        try {
            int num1 = Integer.parseInt(args[0]);
            int num2 = Integer.parseInt(args[1]);

            if (num1 <= 0 || num2 <= 0) {
                throw new PositiveIntegerException("Both numbers should be positive integers.");
            }

            if (num2 <= num1) {
                throw new SecondIntegerLargerException("Second number should be larger than the first.");
            }

            System.out.println("Prime numbers between " + num1 + " and " + num2 + ":");
            for (int i = num1; i <= num2; i++) {
                if (isPrime(i)) {
                    System.out.print(i + " ");
                }
            }
        } catch (NumberFormatException e) {
            System.out.println("Exception: Please enter valid integers as arguments.");
        } catch (PositiveIntegerException | SecondIntegerLargerException e) {
            System.out.println("Exception: " + e.getMessage());
        }
    }

    public static boolean isPrime(int num) {
        if (num <= 1) {
            return false;
        }
        for (int i = 2; i <= Math.sqrt(num); i++) {
            if (num % i == 0) {
                return false;
            }
        }
        return true;
    }
}
```

14)

```java
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;


public class WordCount {

    public static void main(String[] args) {

        for (String fileName : args) {

            Thread thread = new Thread(() -> {

                try {

                    int count = countWords(fileName);

                    System.out.println(fileName + ": " + count);

                } catch (IOException e) {

                    System.out.println("Error reading file: " + fileName);

                }

            });

            thread.start();

        }

    }


    public static int countWords(String fileName) throws IOException {

        int count = 0;

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

            String line;

            while ((line = br.readLine()) != null) {

                if (!line.trim().isEmpty()) {

                    String[] words = line.split("\\s+");

                    count += words.length;

                }

            }

        }

        return count;

    }

}
```

15)

```java
import java.io.BufferedReader;

import java.io.FileReader;

import java.io.IOException;


public class LineCounts {

    public static void main(String[] args) {

        for (String fileName : args) {

            try {

                int count = countLines(fileName);

                System.out.println(fileName + ": " + count);

            } catch (IOException e) {

                System.out.println("Error reading file: " + fileName);

            }

        }

    }


    public static int countLines(String fileName) throws IOException {

        int count = 0;

        try (BufferedReader br = new BufferedReader(new FileReader(fileName))) {

            while (br.readLine() != null) {

                count++;

            }

        }

        return count;

    }

}
```

16)

```
import java.sql.*;
public class WaitingListStatus {
    public static void main(String[] args) throws SQLException{
        String pnrNumber = "ABC123";

        Connection c = DriverManager.getConnection("jdbc:mysql://localhost:3306/your_database",
"your_username","your_password");
        String query = "SELECT * FROM WaitingList WHERE pnr_number = ?";
        PreparedStatement st = c.prepareStatement(query);
        st.setString(1, pnrNumber);

        ResultSet resultSet = st.executeQuery();

        if (resultSet.next()) {
            String trainName = resultSet.getString("train_name");
            String waitingStatus = resultSet.getString("waiting_status");

            System.out.println("PNR Number: " + pnrNumber);
            System.out.println("Train Name: " + trainName);
            System.out.println("Waiting Status: " + waitingStatus);
        } else {
            System.out.println("PNR Number not found in the waiting list.");
        }

    }
}
```

17)

a)

```java
public class PriorityDemo {
    public static void main(String[] args) {
        Thread highPriorityThread = new Thread(new PriorityTask(), "HighPriorityThread");
        Thread lowPriorityThread = new Thread(new PriorityTask(), "LowPriorityThread");

        highPriorityThread.setPriority(Thread.MAX_PRIORITY);
        lowPriorityThread.setPriority(Thread.MIN_PRIORITY);

        highPriorityThread.start();
        lowPriorityThread.start();
    }
    static class PriorityTask implements Runnable {
        public void run() {
            for (int i = 0; i < 5; i++) {
                System.out.println(Thread.currentThread().getName() + " is running iteration " + i);
                try {
                    Thread.sleep(100);
                } catch (InterruptedException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

b)

```java
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;

public class ConvertToLowerCase {
    public static void main(String[] args) {
        String inputFile = "input.txt";
        String outputFile = "output.txt";

        try {
            FileReader fileReader = new FileReader(inputFile);
            BufferedReader bufferedReader = new BufferedReader(fileReader);

            FileWriter fileWriter = new FileWriter(outputFile);
            BufferedWriter bufferedWriter = new BufferedWriter(fileWriter);

            String line;
            while ((line = bufferedReader.readLine()) != null) {

                String[] words = line.split("\\s+");
                for (String word : words) {

                    bufferedWriter.write(word.toLowerCase());
                    bufferedWriter.write(" ");
                }
                bufferedWriter.newLine();
            }

            System.out.println("File conversion completed.");

        } catch (IOException e) {
            e.printStackTrace();
        }
    }
}
```

18)

a)

```java
import java.util.ArrayList;

import java.util.List;


public class OddIntegersCounter {

    public static <T extends Number> int countOddIntegers(List<T> list) {

        int count = 0;

        for (T element : list) {

            if (element.intValue() % 2 != 0) {

                count++;

            }

        }

        return count;

    }


    public static void main(String[] args) {

        List<Integer> integerList = new ArrayList<>();

        integerList.add(1);

        integerList.add(2);

        integerList.add(3);

        integerList.add(4);

        integerList.add(5);


        int oddCount = countOddIntegers(integerList);

        System.out.println("Number of odd integers: " + oddCount);

    }

}
```

b)

```java
public class ElementPositionExchanger {

    public static <T> void exchangeElements(T[] array, int pos1, int pos2) {

        if (pos1 >= 0 && pos1 < array.length && pos2 >= 0 && pos2 < array.length) {

            T temp = array[pos1];

            array[pos1] = array[pos2];

            array[pos2] = temp;

        } else {

            System.out.println("Invalid positions provided.");

        }

    }
    public static void main(String[] args) {
        Integer[] array = {1, 2, 3, 4, 5};

        System.out.println("Before exchange: " + Arrays.toString(array));

        exchangeElements(array, 1, 3);

        System.out.println("After exchange: " + Arrays.toString(array));
    }
}
```

c)

```java
import java.util.List;

public class MaxElementFinder {
    public static <T extends Comparable<T>> T findMaxElementInRange(List<T> list, int begin, int end) {
        if (begin >= 0 && end < list.size() && begin <= end) {
            T max = list.get(begin);
            for (int i = begin + 1; i <= end; i++) {
                T current = list.get(i);
                if (current.compareTo(max) > 0) {
                    max = current;
                }
            }
            return max;
        } else {
            throw new IllegalArgumentException("Invalid range provided.");
        }
    }

    public static void main(String[] args) {
        List<Integer> integerList = Arrays.asList(5, 2, 9, 1, 7, 3);

        int begin = 1;
        int end = 4;

        Integer maxElement = findMaxElementInRange(integerList, begin, end);
        System.out.println("Max element in range [" + begin + ", " + end + "]: " + maxElement);
    }
}
```

19)

```
import javax.swing.*;

import java.awt.event.*;


class Converter_19 extends JFrame implements ActionListener

{

     JLabel l1, l2;

     JTextField text1, text2;

     JButton b1;

     Converter_19()

     {


          l1 = new JLabel("Miles : ", SwingConstants.RIGHT);

          l2 = new JLabel("Kilometers : ",SwingConstants.RIGHT);

          l1.setBounds(20,20,70,30);

          l2.setBounds(20,50,100,30);


          text1=new JTextField("",15);

          text2=new JTextField("",15);

          text1.setBounds(130,20,50,30);

          text2.setBounds(130,60,50,30);

          b1 = new JButton("Convert!");

          b1.setBounds(100,120,120,20);


          add(l1);

          add(text1);

          add(l2);

          add(text2);

          add(b1);


          b1.addActionListener(this);


          setLayout(null);

          setSize(350,200);

          setTitle("Converter");

          setVisible(true);

          setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

     }

     public void actionPerformed(ActionEvent ae)

     {

          double m = Double.parseDouble(text1.getText());

          double km = m*1.609;
          text2.setText(Double.toString(km));
     }
     public static void main(String as[])
     {
          new Converter_19();
     }
}
```

20)

```java
import javax.swing.*;

import java.awt.event.*;


public class CourseRegistrationForm extends JFrame implements ActionListener {


    JLabel nameLabel, addressLabel, phoneLabel, genderLabel, departmentLabel, courseLabel;

    JTextField nameField, addressField, phoneField;

    JComboBox<String> genderBox, departmentBox, courseBox;

    JButton submitButton;

    String[] gender = {"Male","Female"};

    String[]   dept = {"CSE", "ECE", "EEE", "Mech", "Civil"};

    String[] course = {"C", "C++", "JAVA", "PYTHON"};


    public CourseRegistrationForm() {

        setTitle("Course Registration Form");

        setSize(400, 300);

        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        setLayout(null);


        nameLabel = new JLabel("Name:");

        nameLabel.setBounds(20, 20, 80, 20);

        add(nameLabel);


        nameField = new JTextField();

        nameField.setBounds(120, 20, 200, 20);

        add(nameField);


        addressLabel = new JLabel("Address:");

        addressLabel.setBounds(20, 50, 80, 20);

        add(addressLabel);


        addressField = new JTextField();

        addressField.setBounds(120, 50, 200, 20);

        add(addressField);


        phoneLabel = new JLabel("Phone:");

        phoneLabel.setBounds(20, 80, 80, 20);

        add(phoneLabel);


        phoneField = new JTextField();

        phoneField.setBounds(120, 80, 200, 20);
        add(phoneField);

        genderLabel = new JLabel("Gender:");
        genderLabel.setBounds(20, 110, 80, 20);
        add(genderLabel);

        genderBox = new JComboBox<>(gender);
        genderBox.setBounds(120, 110, 100, 20);
        add(genderBox);

        departmentLabel = new JLabel("Department:");
        departmentLabel.setBounds(20, 140, 80, 20);
        add(departmentLabel);

        departmentBox = new JComboBox<>(dept);
        departmentBox.setBounds(120, 140, 100, 20);
        add(departmentBox);

        courseLabel = new JLabel("Course:");
        courseLabel.setBounds(20, 170, 80, 20);
        add(courseLabel);

        courseBox = new JComboBox<>(course);
        courseBox.setBounds(120, 170, 100, 20);
        add(courseBox);

        submitButton = new JButton("Submit");
        submitButton.setBounds(150, 210, 100, 30);
        add(submitButton);

        submitButton.addActionListener(this);
        setVisible(true);
    }

    @Override
    public void actionPerformed(ActionEvent e) {
        String name = nameField.getText();
        String course = (String) courseBox.getSelectedItem();
        JOptionPane.showMessageDialog(this, name + ", you have successfully enrolled in " + course);
    }

    public static void main(String[] args) {
        new CourseRegistrationForm();
    }
}
```