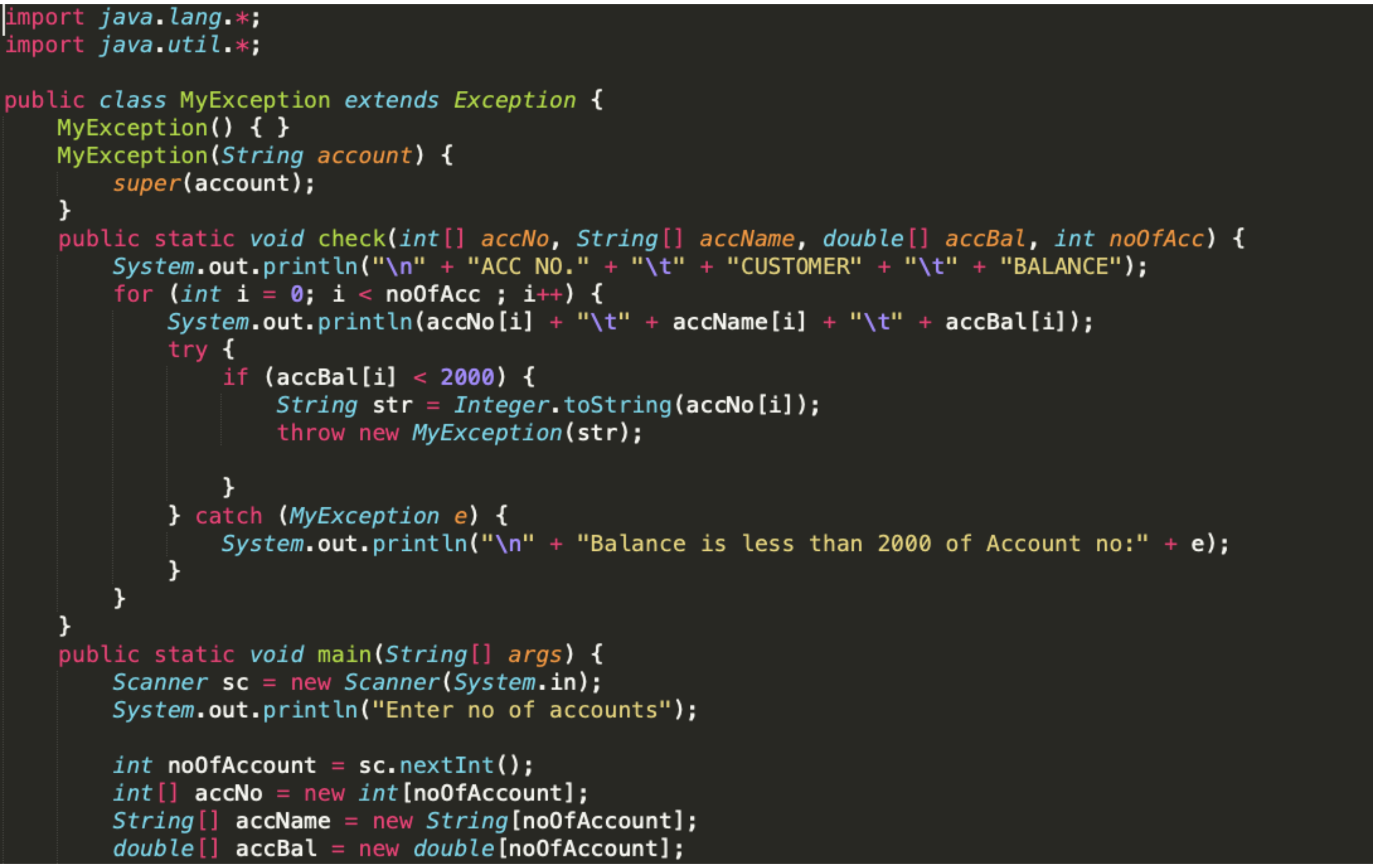
Web Technologies (CSN 463): Assignment-4

Outputs:

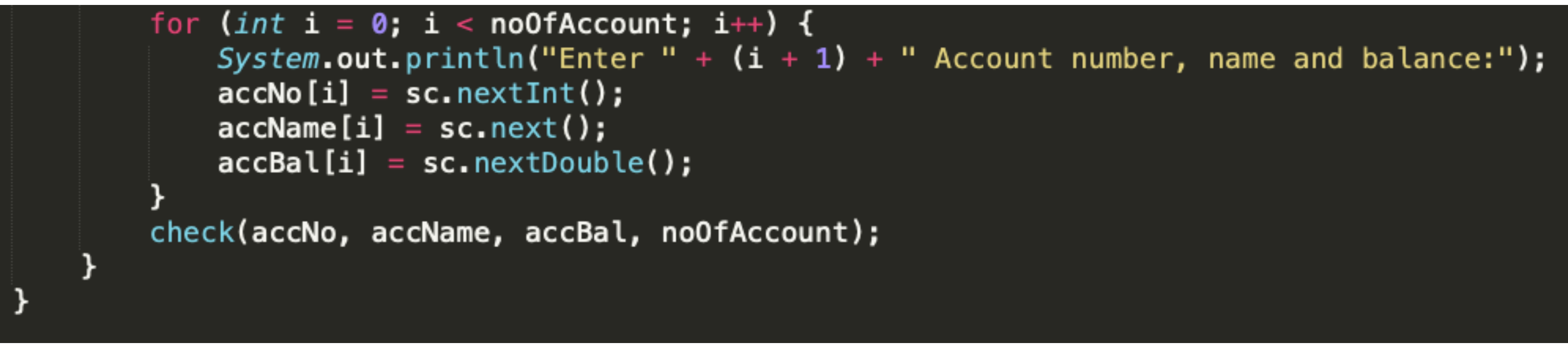
1. Write a program to create own exception class MyException that performs the following task:

- Details of account numbers, customer names, and balance amounts are taken in the form of three arrays.
- In main() method, the details are displayed using a for-loop. At this time, check is done if in any account the balance amount is less than the minimum balance amount to be apt in the account.
- If it is so, then MyException is raised and a message is displayed "Balance amount is less".

Code:

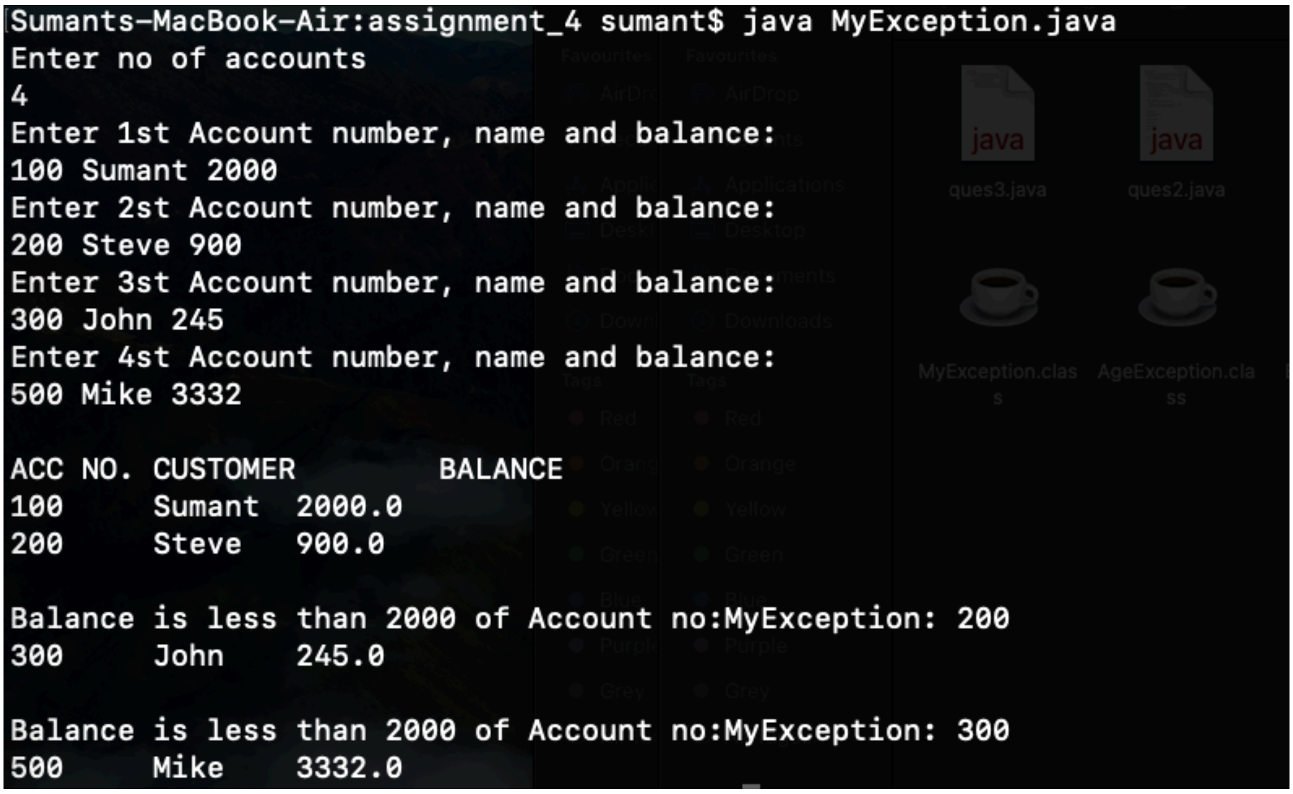


```java
import java.lang.*;
import java.util.*;

public class MyException extends Exception {
    MyException() { }
    MyException(String account) {
        super(account);
    }
    public static void check(int[] accNo, String[] accName, double[] accBal, int noOfAcc) {
        System.out.println("\n" + "ACC NO." + "\t" + "CUSTOMER" + "\t" + "BALANCE");
        for (int i = 0; i < noOfAcc ; i++) {
            System.out.println(accNo[i] + "\t" + accName[i] + "\t" + accBal[i]);
            try {
                if (accBal[i] < 2000) {
                    String str = Integer.toString(accNo[i]);
                    throw new MyException(str);

                }
            } catch (MyException e) {
                System.out.println("\n" + "Balance is less than 2000 of Account no:" + e);
            }
        }
    }
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter no of accounts");

        int noOfAccount = sc.nextInt();
        int[] accNo = new int[noOfAccount];
        String[] accName = new String[noOfAccount];
        double[] accBal = new double[noOfAccount];
```



```
        for (int i = 0; i < noOfAccount; i++) {
            System.out.println("Enter " + (i + 1) + " Account number, name and balance:");
            accNo[i] = sc.nextInt();
            accName[i] = sc.next();
            accBal[i] = sc.nextDouble();
        }
        check(accNo, accName, accBal, noOfAccount);
    }
}
```

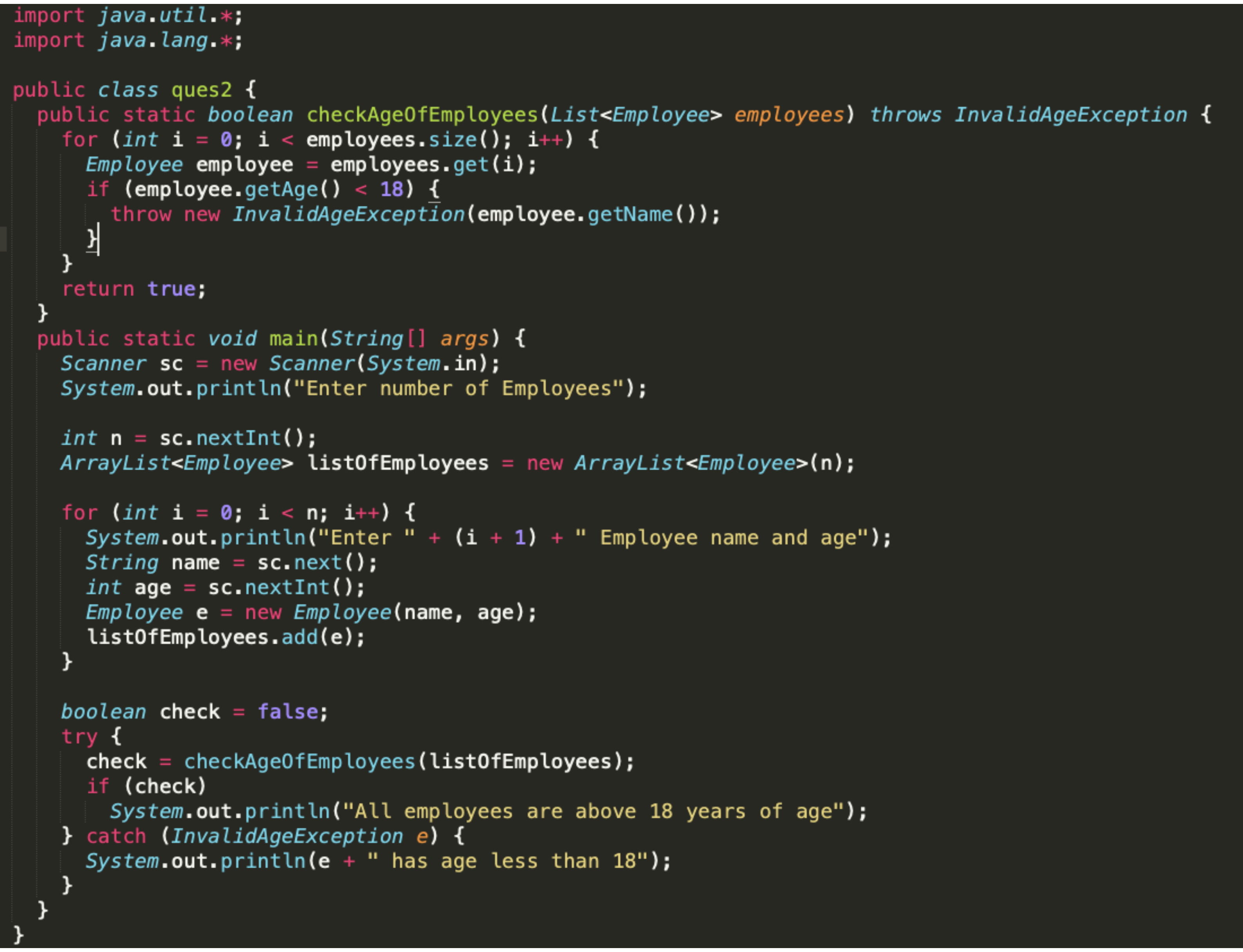
Output:



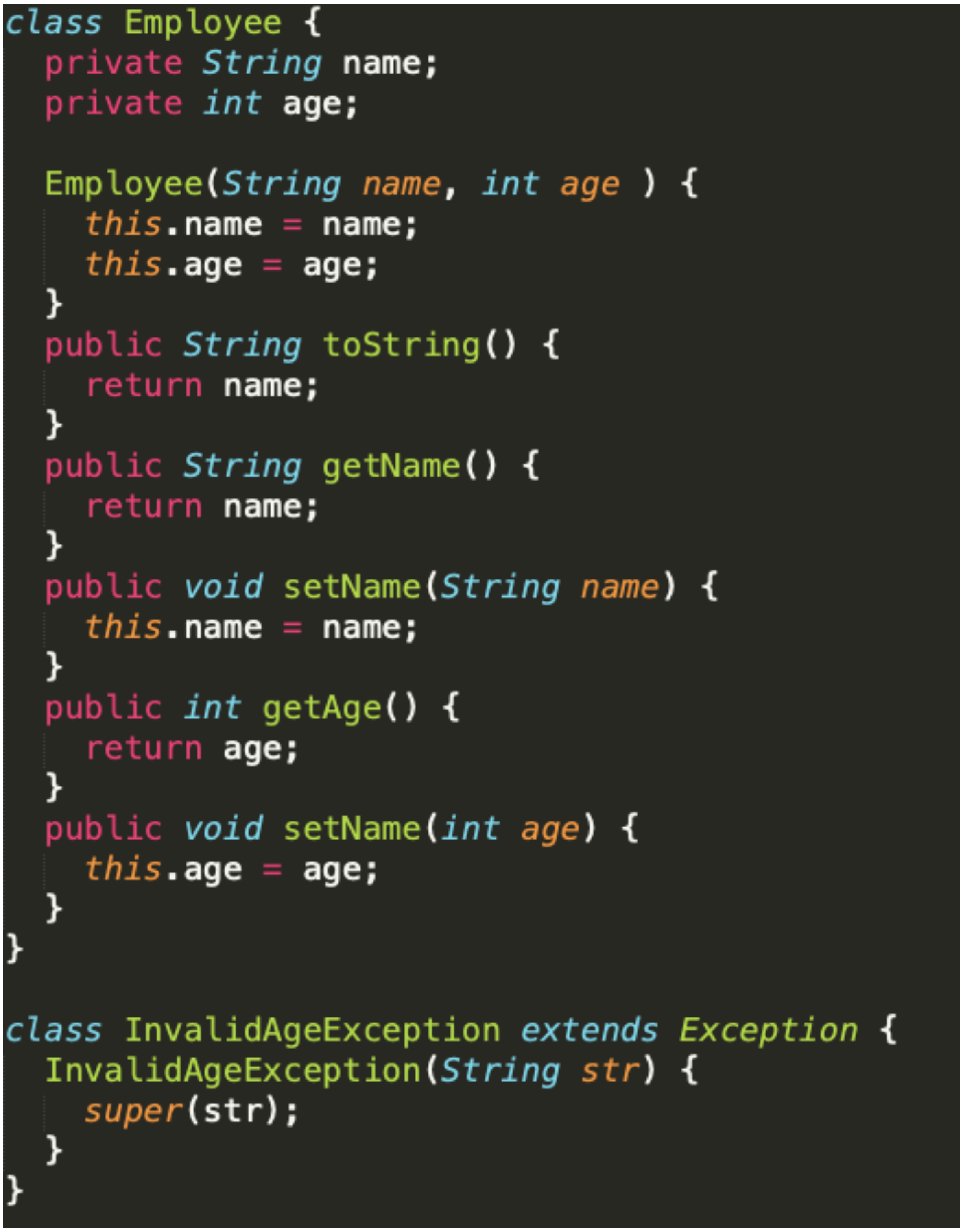
```
Sumants-MacBook-Air:assignment_4 sumant$ java MyException.java
Enter no of accounts
4
Enter 1st Account number, name and balance:
100 Sumant 2000
Enter 2st Account number, name and balance:
200 Steve 900
Enter 3st Account number, name and balance:
300 John 245
Enter 4st Account number, name and balance:
500 Mike 3332

ACC NO. CUSTOMER        BALANCE
100     Sumant  2000.0
200     Steve   900.0

Balance is less than 2000 of Account no:MyException: 200
300     John    245.0

Balance is less than 2000 of Account no:MyException: 300
500     Mike    3332.0
```

2. You have a list of employees and if any employee's age in list of employees is less than 18, then you need to throw invalidAgeException(Our custom exception).

Code:

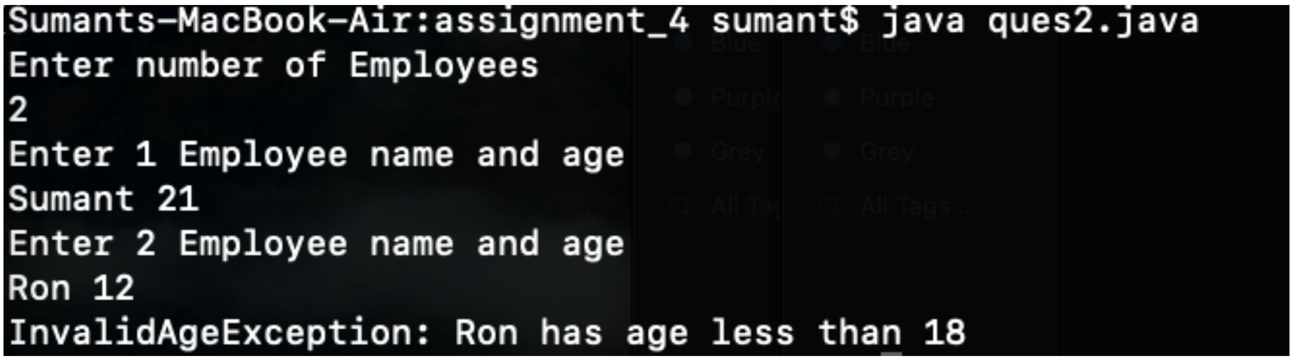


```java
import java.util.*;
import java.lang.*;

public class ques2 {
  public static boolean checkAgeOfEmployees(List<Employee> employees) throws InvalidAgeException {
    for (int i = 0; i < employees.size(); i++) {
      Employee employee = employees.get(i);
      if (employee.getAge() < 18) {
        throw new InvalidAgeException(employee.getName());
      }
    }
    return true;
  }
  public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.println("Enter number of Employees");

    int n = sc.nextInt();
    ArrayList<Employee> listOfEmployees = new ArrayList<Employee>(n);

    for (int i = 0; i < n; i++) {
      System.out.println("Enter " + (i + 1) + " Employee name and age");
      String name = sc.next();
      int age = sc.nextInt();
      Employee e = new Employee(name, age);
      listOfEmployees.add(e);
    }

    boolean check = false;
    try {
      check = checkAgeOfEmployees(listOfEmployees);
      if (check)
        System.out.println("All employees are above 18 years of age");
    } catch (InvalidAgeException e) {
      System.out.println(e + " has age less than 18");
    }
  }
}
```

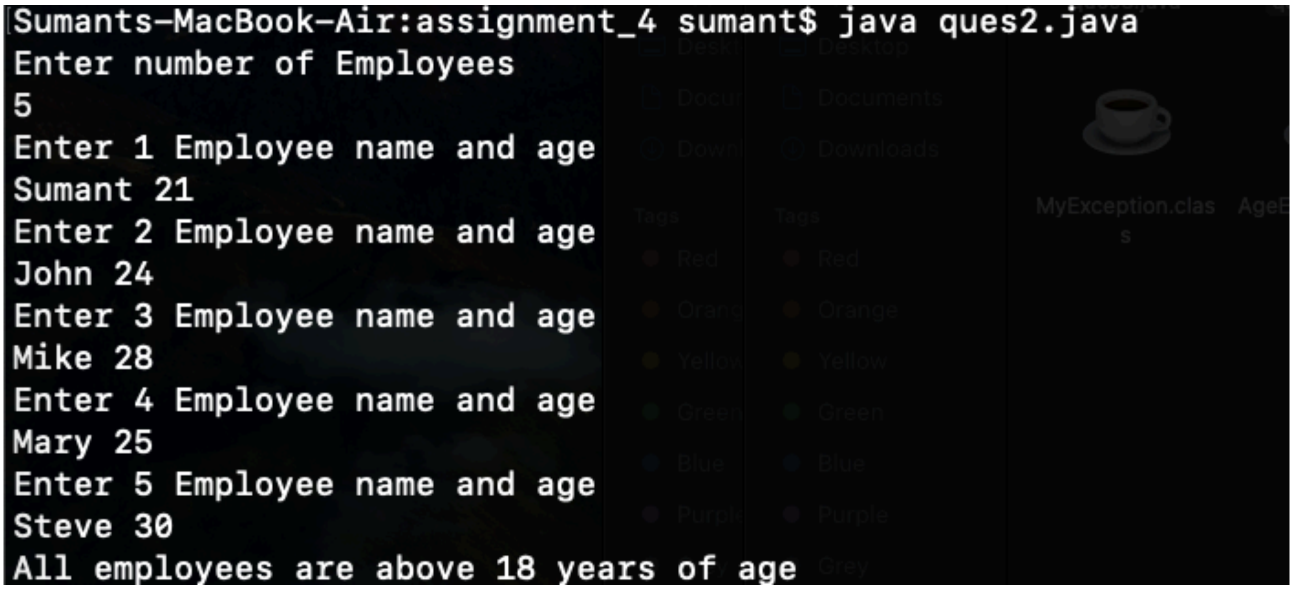


```java
class Employee {
  private String name;
  private int age;

  Employee(String name, int age ) {
    this.name = name;
    this.age = age;
  }
  public String toString() {
    return name;
  }
  public String getName() {
    return name;
  }
  public void setName(String name) {
    this.name = name;
  }
  public int getAge() {
    return age;
  }
  public void setName(int age) {
    this.age = age;
  }
}

class InvalidAgeException extends Exception {
  InvalidAgeException(String str) {
    super(str);
  }
}
```

Output:

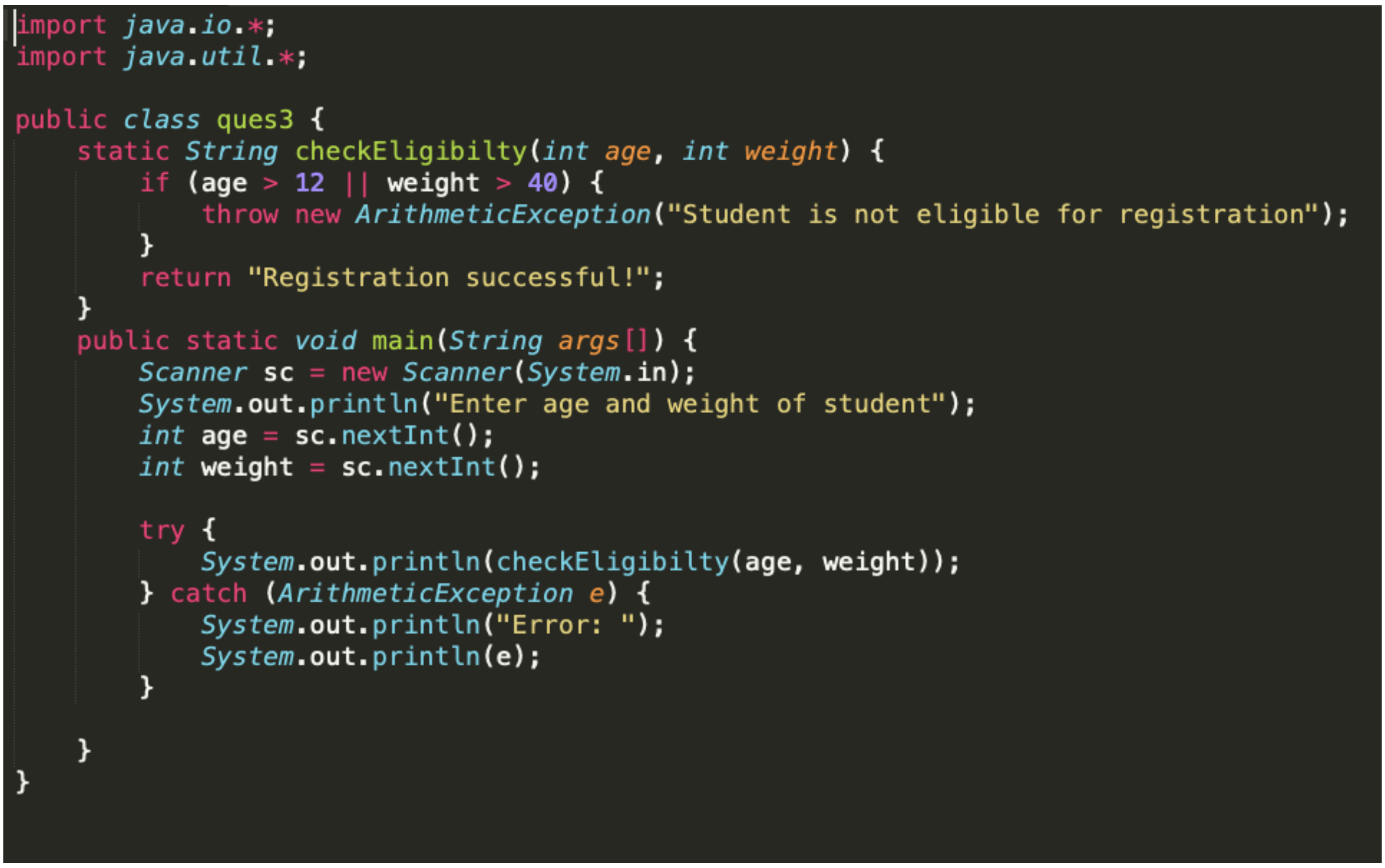


```
Sumants-MacBook-Air:assignment_4 sumant$ java ques2.java
Enter number of Employees
2
Enter 1 Employee name and age
Sumant 21
Enter 2 Employee name and age
Ron 12
InvalidAgeException: Ron has age less than 18
```



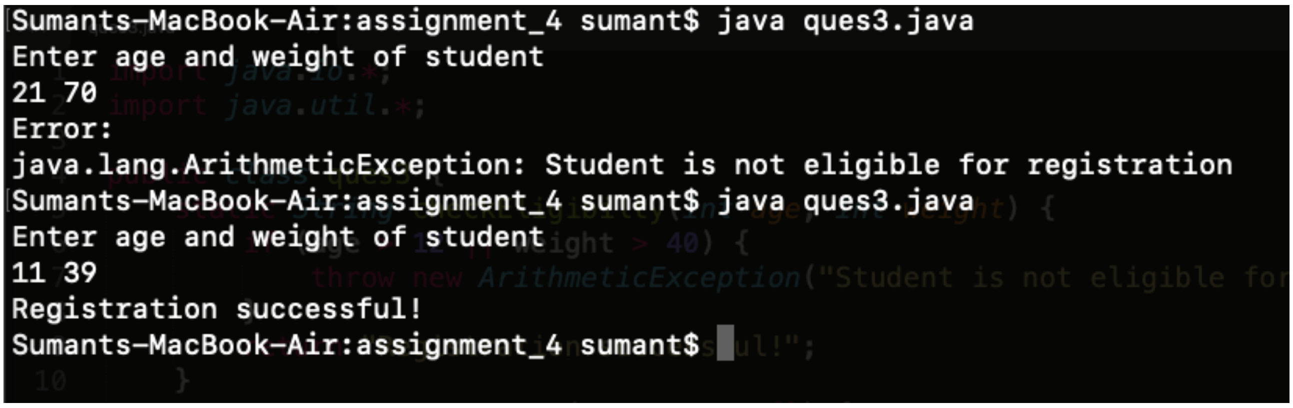
```
Sumants-MacBook-Air:assignment_4 sumant$ java ques2.java
Enter number of Employees
5
Enter 1 Employee name and age
Sumant 21
Enter 2 Employee name and age
John 24
Enter 3 Employee name and age
Mike 28
Enter 4 Employee name and age
Mary 25
Enter 5 Employee name and age
Steve 30
All employees are above 18 years of age
```

3. Let's say we have a requirement where we need to only register the students when their age is less than 12 and weight is less than 40, if any of the condition is not met then the user should get an ArithmeticException with the warning message "Student is not eligible for registration". Implement the logic by placing the code in the method that checks student eligibility. If the entered student age and weight doesn't met the criteria, then throw the exception using throw keyword.



```java
import java.io.*;
import java.util.*;

public class ques3 {
    static String checkEligibilty(int age, int weight) {
        if (age > 12 || weight > 40) {
            throw new ArithmeticException("Student is not eligible for registration");
        }
        return "Registration successful!";
    }
    public static void main(String args[]) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter age and weight of student");
        int age = sc.nextInt();
        int weight = sc.nextInt();

        try {
            System.out.println(checkEligibilty(age, weight));
        } catch (ArithmeticException e) {
            System.out.println("Error: ");
            System.out.println(e);
        }

    }
}
```

Output:



```
Sumants-MacBook-Air:assignment_4 sumant$ java ques3.java
Enter age and weight of student
21 70
Error:
java.lang.ArithmeticException: Student is not eligible for registration
Sumants-MacBook-Air:assignment_4 sumant$ java ques3.java
Enter age and weight of student
11 39
Registration successful!
Sumants-MacBook-Air:assignment_4 sumant$
```