

1.

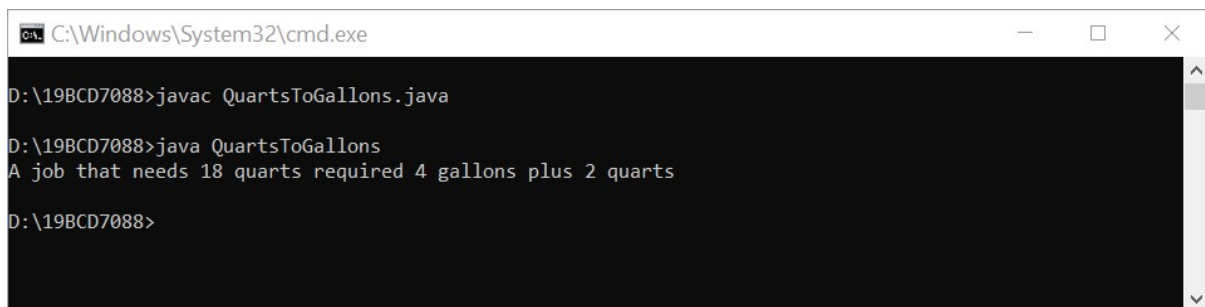
a.

Valiveti manikanta bhuvanesh

19BCD7088

Write a program that declares a named constant to hold the number of quarts in a gallon (4). Also declare a variable to represent the number of quarts needed for a painting job, and assign an appropriate value—for example, 18. Compute and display the number of gallons and quarts needed for the job. Display explanatory text with the values—for example, A job that needs 18 quarts requires 4 gallons plus 2 quarts. Save the program as QuartsToGallons.java.

```
public class QuartsToGallons{  
  
    public static void main(String args[]) {  
  
        int qg = 4;  
  
        int n = 18;  
  
        System.out.println("A job that needs " + n + " quarts required " + n/qg + " gallons plus " + n%qg + "  
quarts");  
  
    }  
  
}
```



```
C:\Windows\System32\cmd.exe  
D:\19BCD7088>javac QuartsToGallons.java  
D:\19BCD7088>java QuartsToGallons  
A job that needs 18 quarts required 4 gallons plus 2 quarts  
D:\19BCD7088>
```

b.

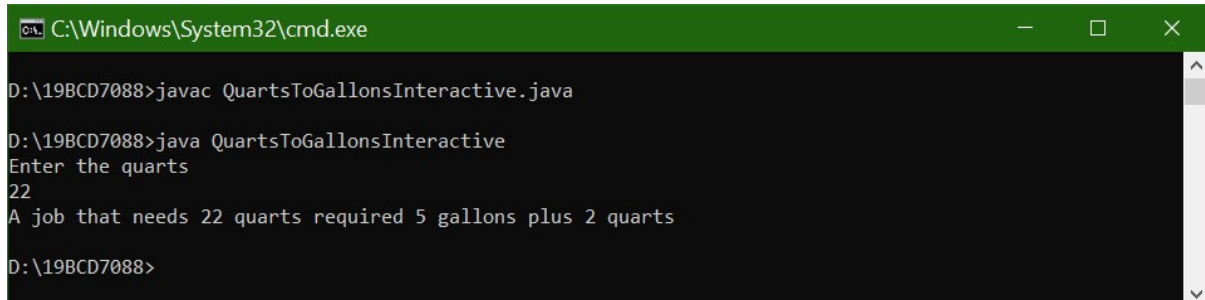
Convert the QuartsToGallons program to an interactive application. Instead of assigning a value to the number of quarts, accept the value from the user as input. Save the revised program as QuartsToGallonsInteractive.java.

```
import java.util.Scanner;  
  
public class QuartsToGallonsInteractive{  
  
    public static void main(String args[]) {  
  
        Scanner sc=new Scanner (System.in);  
  
        int qg = 4;  
  
        System.out.println("Enter the quarts");  
  
        int n = sc.nextInt();
```

```

        System.out.println("A job that needs " + n + " quarts required " + n/qg + " gallons plus "
+n%qg + " quarts");
    }
}

```



The screenshot shows a Windows command prompt window titled "C:\Windows\System32\cmd.exe". The user has entered the following commands and received the following output:

```

D:\19BCD7088>javac QuartsToGallonsInteractive.java
D:\19BCD7088>java QuartsToGallonsInteractive
Enter the quarts
22
A job that needs 22 quarts required 5 gallons plus 2 quarts
D:\19BCD7088>

```

c.

Now, add exception-handling capabilities to this program and continuously reprompt the user while any nonnumeric value is entered. Save the file as QuartsToGallonsWithExceptionHandling.java.

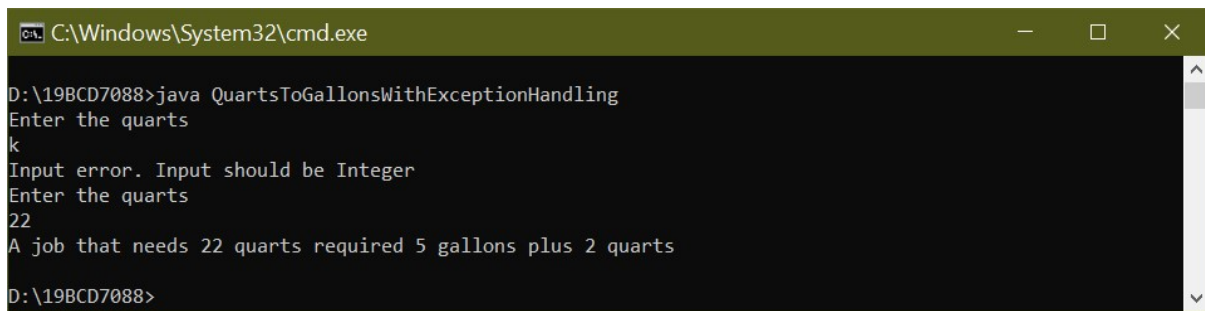
```

import java.util.*;

public class QuartsToGallonsWithExceptionHandling{
    public static void main(String[] args) {
        Scanner sc=new Scanner (System.in);
        int qg = 4;
        int p=1;
        while(p!=0){
            try{
                System.out.println("Enter the quarts");
                int n = Integer.parseInt(sc.nextLine());
                System.out.println("A job that needs " + n + " quarts required " + n/qg + "
gallons plus " +n%qg + " quarts");
                p=0;
            }
            catch(Exception e){
                System.out.println("Input error. Input should be Integer");
            }
        }
    }
}

```

}



```
C:\Windows\System32\cmd.exe
D:\19BCD7088>java QuartsToGallonsWithExceptionHandling
Enter the quarts
k
Input error. Input should be Integer
Enter the quarts
22
A job that needs 22 quarts required 5 gallons plus 2 quarts
D:\19BCD7088>
```

2.

a.

Allow a user to enter any number of double values up to 15. The user should enter 99999 to quit entering numbers. Display an error message if the user quits without entering any numbers; otherwise, display each entered value and its distance from the average. Save the file as DistanceFromAverage.java.

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

```
public class DistanceFromAverage{
```

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

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

```
        double[] a=new double[15];
```

```
        double sum=0;
```

```
        int i=0,k=0;
```

```
        double avg,n=0;
```

```
        System.out.println("For exiting enter 9999 as value");
```

```
        while(i<a.length){
```

```
            System.out.println("Enter the value");
```

```
            a[i]=Double.parseDouble(sc.nextLine());
```

```
            sum=sum+a[i];
```

```
            n=a[i];
```

```
            if(n==99999){
```

```
                k=i;
```

```
                i=a.length;
```

```
            }
```

```
            i++;
```

```
        }
```

```

        if(n==99999){
            i=k;

            System.out.println((a.length-i) + " values not entered");
        }

        else{

            avg=sum/(i+1);

            for(int j=0;j<a.length;j++){

                System.out.println(a[j] + " is ," + (a[j]-avg) + " distance away from
average "+avg);

            }

        }

    }
}

```

```

C:\Windows\System32\cmd.exe
D:\19BCD7088>java DistanceFromAverage
For exiting enter 9999 as value
Enter the value
1
Enter the value
2
Enter the value
3
Enter the value
4
Enter the value
5
Enter the value
6
Enter the value
7
Enter the value
8
Enter the value
9
Enter the value
10
Enter the value
11
Enter the value
12
Enter the value
13
Enter the value
14
Enter the value
15
1.0 is ,-6.5 distance away from average 7.5
2.0 is ,-5.5 distance away from average 7.5
3.0 is ,-4.5 distance away from average 7.5
4.0 is ,-3.5 distance away from average 7.5
5.0 is ,-2.5 distance away from average 7.5
6.0 is ,-1.5 distance away from average 7.5
7.0 is ,-0.5 distance away from average 7.5
8.0 is ,0.5 distance away from average 7.5
9.0 is ,1.5 distance away from average 7.5
10.0 is ,2.5 distance away from average 7.5
11.0 is ,3.5 distance away from average 7.5
12.0 is ,4.5 distance away from average 7.5
13.0 is ,5.5 distance away from average 7.5
14.0 is ,6.5 distance away from average 7.5
15.0 is ,7.5 distance away from average 7.5
D:\19BCD7088>

```

b.

Now, modify that program to first prompt the user to enter an integer that represents the array size. Java generates a `NumberFormatException` if you attempt to enter a noninteger value using `nextInt()`; handle this exception by displaying an appropriate error message. Create an array using the integer entered as the size.

Java generates a `NegativeArraySizeException` if you attempt to create an array with a negative size; handle this exception by setting the array size to a default value of five.

If the array is created successfully, use exception-handling techniques to ensure that each entered array value is a double before the program calculates each element's distance from the average. Save the file as `DistanceFromAverageWithExceptionHandling.java`.

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

```
public class DistanceFromAverageWithExceptionHandling{
```

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

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

```
        double[] a=new double[5];
```

```
        double sum=0;
```

```
        int i=0,k=0;
```

```
        double avg,n=0;
```

```
        int size=5;
```

```
        int p=1;
```

```
        while(p!=0){
```

```
            try {
```

```
                System.out.println("Enter array size");
```

```
                size=Integer.parseInt(sc.nextLine());
```

```
                p=0;
```

```
            }
```

```
            catch (NumberFormatException e) {
```

```
                System.out.println("Size should be a number");
```

```
            }
```

```
        }
```

```
        try{
```

```
            a= new double[size];
```

```
        }
```

```

        catch(NegativeArraySizeException e){
            size=5;
            System.out.println("Array size can not be negative value taking 5 as default
value of size");
        }
        while(i<a.length){
            try{
                System.out.println("Enter the value");
                a[i]=Double.parseDouble(sc.nextLine());
                sum=sum+a[i];
                n=a[i];
                if(n==9999){
                    k=i;
                    i=a.length;
                }
                i++;
            }
        }
        catch(Exception e){
            System.out.println("Value should be double");
        }
    }
    if(n==9999){
        i=k;
        System.out.println((a.length-i) + " values not entered");
    }
    else{
        avg=sum/(i+1);
        for(int j=0;j<a.length;j++){
            System.out.println(a[j] + " is ," + (a[j]-avg) + " distance away from
average "+avg);
        }
    }
}

```

```

    }
}

```

```

C:\Windows\System32\cmd.exe
D:\19BCD7088>javac DistanceFromAverageWithExceptionHandling.java
D:\19BCD7088>java DistanceFromAverageWithExceptionHandling
Enter array size
k
Size should be a number
Enter array size
-4
Array size can not be negative value taking 5 as default value of size
Enter the value
25
Enter the value
36
Enter the value
24
Enter the value
15
Enter the value
95
25.0 is ,-7.5 distance away from average 32.5
36.0 is ,3.5 distance away from average 32.5
24.0 is ,-8.5 distance away from average 32.5
15.0 is ,-17.5 distance away from average 32.5
95.0 is ,62.5 distance away from average 32.5
D:\19BCD7088>

```

3.

a.

Create a `CourseException` class that extends `Exception` and whose constructor receives a `String` that holds a college course's department (for example, CIS), a course number (for example, 101), and a number of credits (for example, 3). Save the file as `CourseException.java`. Create a `Course` class with the same fields and whose constructor requires values for each field. Upon construction, throw a `CourseException` if the department does not consist of three letters, if the course number does not consist of three digits between 100 and 499 inclusive, or if the credits are less than 0.5 or more than 6. Save the class as `Course.java`. Write an application that establishes an array of at least six `Course` objects with valid and invalid values. Display an appropriate message when a `Course` object is created successfully and when one is not. Save the file as `ThrowCourseException.java`.

```
class CourseException extends Exception{
```

```
    CourseException(String msg) {
```

```
        super(msg);
```

```
    }
```

```
}
```

```
class Course {
```

```
    String dept;
```

```

int cn;

double c;

public Course(String dept, int cn, double c) throws CourseException {
    if(dept.length()!=3 || (cn<100 || cn>499) || (c<0.5 || c>6)) {
        throw new CourseException("Error in given details");
    }

    this.dept = dept;
    this.cn = cn;
    this.c = c;
    System.out.println("Created successfully");
}
}

public class ThrowCourseException{
    public static void main(String[] args) throws CourseException{
        Course c[]=new Course[6];
        c[0]=new Course("BCE",350,5);
        c[1]=new Course("BCD",260,3);
        c[2]=new Course("BCI",275,4);
        c[3]=new Course("BCN",150,0.7);
        c[4]=new Course("BCR",185,1);
        c[5]=new Course("BCB",230,7);
    }
}

```

```

C:\Windows\System32\cmd.exe
D:\19BCD7088>javac Course.java

D:\19BCD7088>javac ThrowCourseException.java

D:\19BCD7088>java ThrowCourseException
Created successfully
Created successfully
Created successfully
Created successfully
Created successfully
Exception in thread "main" CourseException: Error in given details
    at Course.<init>(Course.java:7)
    at ThrowCourseException.main(ThrowCourseException.java:9)

D:\19BCD7088>

```

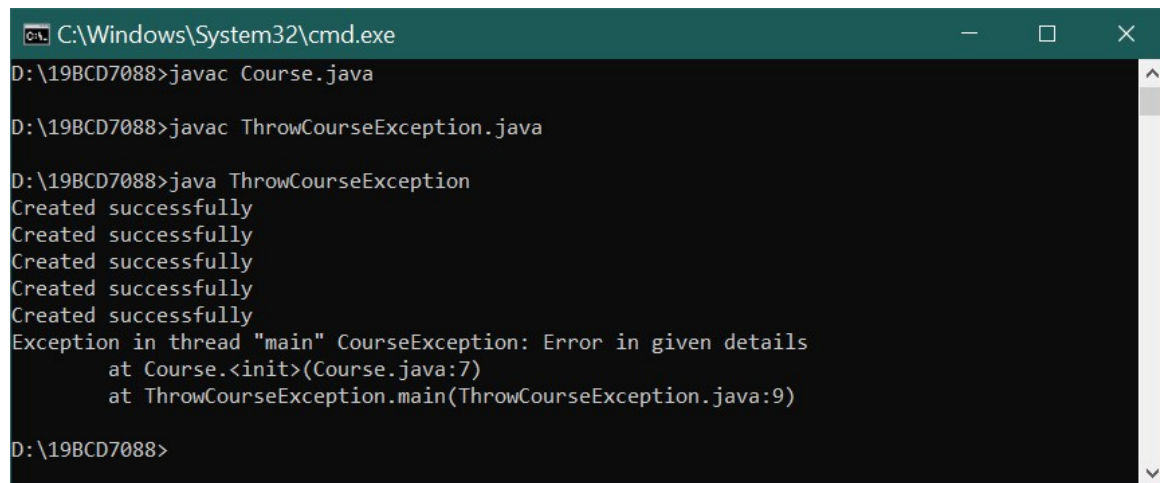

3.b.

Modify the CourseException class to extend RuntimeException class and identify the differences.

```
class CourseException extends RuntimeException{
    CourseException(String msg) {
        super(msg);
    }
}
.....
class Course {
    String dept;
    int cn;
    double c;
    public Course(String dept, int cn, double c) throws CourseException {
        if(dept.length()!=3 || (cn<100 || cn>499) || (c<0.5 || c>6)) {
            throw new CourseException("Error in given details");
        }
        this.dept = dept;
        this.cn = cn;
        this.c = c;
        System.out.println("Created successfully");
    }
}

public class ThrowCourseException{
    public static void main(String[] args) throws CourseException{
        Course c[]=new Course[6];
        c[0]=new Course("BCE",350,5);
        c[1]=new Course("BCD",260,3);
        c[2]=new Course("BCI",275,4);
        c[3]=new Course("BCN",150,0.7);
        c[4]=new Course("BCR",185,1);
        c[5]=new Course("BCB",230,7);
    }
}
```

}



```
C:\Windows\System32\cmd.exe
D:\19BCD7088>javac Course.java
D:\19BCD7088>javac ThrowCourseException.java
D:\19BCD7088>java ThrowCourseException
Created successfully
Created successfully
Created successfully
Created successfully
Created successfully
Exception in thread "main" CourseException: Error in given details
    at Course.<init>(Course.java:7)
    at ThrowCourseException.main(ThrowCourseException.java:9)
D:\19BCD7088>
```