

Student Name(s): \_\_\_\_\_

Assignment 02	(420-JA4-AB) Programming I
Overview	Basic Elements of Java
Outcomes	After completing this project, you will: ▲ Learn about basic elements of Java ▲ Learn about control structures ▲ learn how to use arrays
What you'll need	To complete this assignment, you will need: ▲ the worksheet below
Precautions	None
This is a team project assignment. You must complete all the three projects.	

### ■ Project-1 Classify Numbers (ClassifyNumbers.java)

This program reads a given set of integers and then prints the number of odd integers, the number of even integers, and the number of zeros.

The program reads 20 integers, but you can easily modify it to read any set of numbers. In fact, you can modify the program so that it first prompts the user to specify how many integers are to be read.

**Input:** A 20 integers—positive, negative, or zeros.

**Output:** The number of zeros, even numbers, and odd numbers.

### Problem analysis and algorithm design

After reading a number, you need to check whether it is even or odd. Suppose the value is stored in the variable number. Divide the number by 2 and check the remainder. If the remainder is zero, the number is even. Increment the even count and then check whether the number is zero. If it is, increment the zero count. If the remainder is not zero, increment the odd count.

The program uses a switch statement to decide whether number is odd or even.

Suppose that number is odd. Dividing by 2 gives the remainder 1 if number is positive, and the remainder -1 if negative. If the number is even, dividing it by 2 gives the remainder 0 whether number is positive or negative. You can use the mod operator, %, to find the remainder.

For example:

$6 \% 2 = 0$ ,  $-4 \% 2 = 0$ ,  $-7 \% 2 = -1$ ,  $15 \% 2 = 1$

Repeat the preceding process of analyzing a number for each number in the list.

Translates this discussion into an algorithm:

**Write your step-by-step algorithm.**

### Variables



Let's first describe the variables that are necessary to develop the program. This will simplify the discussion that follows.

Because you want to count the number of zeros, even numbers, and odd numbers, you need three variables of the type int – say, zeros, evens, and odds – to track the counts. You also need a variable – say, number – to read and store the number to be analyzed, and another variable – say, counter – to count the numbers analyzed.

Clearly, you must initialize the variables zeros, evens, and odds to zero. You can initialize these variables right after you declare them.

List your variables here:

## Named Constants

## Formulas

## Main Algorithm

Based on the preceding discussion, you can now write the main detailed algorithm.

Your main algorithm goes here.

## Source Code

### Sample Run:

In this sample run, the user input is shaded.)

Please enter 20 integers, positive, negative, or zeros.

0 0 -2 -3 -5 6 7 8 0 3 0 -23 -8 0 2 9 0 12 67 54

0 0 -2 -3 -5 6 7 8 0 3 0 -23 -8 0 2 9 0 12 67 54

There are 13 evens, which also includes 6 zeros.

Total number of odds is: 7.

```
Please enter 20 integers, positive, negative, or zeros.
```

```
0 0 -2 -3 -5 6 7 8 0 3 0 -23 -8 0 2 9 0 12 67 54
```

```
0 0 -2 -3 -5 6 7 8 0 3 0 -23 -8 0 2 9 0 12 67 54
```

```
There are 13 evens, which also includes 6 zeros
```

```
Total number of odds is: 7
```



## ■ Project-2 Student Grade (StudentGrade.java)

Write a program that reads a student's first and last names followed by five test scores. The program should output the student's first name, last name, the five test scores, and the average test score. Output the average test score with two decimal places.

The data to be read is stored in a file named test.txt; the output should be stored in a file named testavg.txt.

**Input:** A file containing the student's first name, last name, and the five test scores.

**Output:** The student's first name, last name, five test scores, and the average of the five test scores, saved to a file.

### Problem analysis and algorithm design

To find the average of the five test scores, you add the test scores and divide the sum by 5. The input data is in the following form: the student's first name, followed by the last name, followed by the five test scores. Therefore, we read the student's first name, followed by the last name, followed by the five test scores.

This problem analysis translates into the following algorithm:

Write your step-by-step algorithm.

You output the average test score in the fixed-decimal format with two decimal places.

### Variables

The program needs to read a student's first name, last name, and five test scores.

Therefore, you need two variables to store the student's first name and last name, and five variables to store the five test scores. To find the average, you must add the five test scores and then divide the sum by 5. Thus, you also need a variable to store the average test score. Furthermore, because the input data is in a file and the output is to be stored in a file, you must declare and initialize the appropriate variables.

List your variables here.

### Named Constants

### Formulas

### Main Algorithm

In the preceding sections, we analyzed the problem and determined the formulas to perform the calculations. We also determined the necessary variables. Now we can expand the algorithm given in the Problem Analysis and Algorithm Design section to solve the Student Grade problem given at the beginning of this programming case.

Based on the preceding discussion, you can now write the main detailed algorithm.

Your main algorithm goes here.



## Source Code

### Sample Run:

Input File (contents of the file test.txt):

Don Johnson 87.5 89.00 65 37.5 98.0

Output File (contents of the file testavg.txt):

Student Name: Don Johnson

Test scores: 87.50 89.00 65.00 37.50 98.00

Average test score: 75.40



## ■ Project-3 Movie Ticket Sale and Donation to Charity (MovieTicketSale.java)

A movie in a local theater is in great demand. The theater owner has decided to donate to a local charity a portion of the gross amount generated from the movie.

This project case designs and implements a program that prompts the user to input the movie name, adult ticket price, child ticket price, number of adult tickets sold, number of child tickets sold, and percentage of the gross amount to be donated to the charity. Note that the decimal numbers are output with two decimal places.

**Input:** The input to the program consists of the movie name, adult ticket price, child ticket price, number of adult tickets sold, number of child tickets sold, and percentage of the gross amount to be donated to the charity.

**Output:** See below the Sample run.

### Problem analysis and algorithm design

To calculate the amount donated to the local charity and the net sale, you first need to determine the gross amount. To calculate the gross amount, you multiply the number of adult tickets sold by the price of an adult ticket, multiply the number of child tickets sold by the price of a child ticket, and then add these two numbers:

```
grossAmount = adultTicketPrice * noOfAdultTicketsSold + childTicketPrice * noOfChildTicketsSold;
```

Next, you determine the percentage of the amount donated to the charity, and then calculate the net sale amount by subtracting the amount donated from the gross amount. The formulas to calculate the amount donated and the net sale amount are given below.

```
amountDonated = grossAmount * percentDonation / 100;
```

```
netSaleAmount = grossAmount - amountDonated;
```

Write your step-by-step algorithm.

### Variables

From the preceding discussion, it follows that you need variables to store the movie name, adult ticket price, child ticket price, number of adult tickets sold, number of child tickets sold, percentage of the gross amount donated to the charity, gross amount, amount donated, and net sale amount. You also need a variable to get the string containing the sales data and a string to format the output.

List your variables here.

### Named Constants

### Formulas

```
grossAmount = adultTicketPrice * noOfAdultTicketsSold + childTicketPrice * noOfChildTicketsSold;
amountDonated = grossAmount * percentDonation / 100;
netSaleAmount = grossAmount - amountDonated;
```

### FORMATTING THE OUTPUT



Format the output as shown in the Sample run.

### MAIN ALGORITHM

In the preceding sections, we analyzed the problem and determined the formulas to do the calculations. We also determined the necessary variables and the output string. We can now expand the algorithm given in the section Problem Analysis and Algorithm Design to solve the problem given at the beginning of this programming project case.

Your main algorithm goes here.

### Source Code

### Sample Run:

Enter the movie name: Dino King: Journey to Fire Mountain

Enter the price of an adult ticket: 12.00

Enter the price of a child ticket: 7.00

Enter the number of adult ticket sold: 800

Enter the number of child ticket sold: 1850

Enter the percentage of the donation: 10

```
Movie Name: Dino King: Journey to Fire Mountain
Number of Tickets Sold: 2650
Gross Amount: $22550.00
Percentage of the Gross Amount Donated: 10.00%
Amount Donated: $2255.00
Net Sale: $20295.00
```

