

Lab 03

Submit a single .zip file containing all the .java and .doc/.pdf/.txt files.

A - Object-Oriented Programming in Java Basics - Class Design (2 points)

Define the attributes and behaviors for the following classes. You do not need to code.

A Fraction

B - Object-Oriented Programming in Java Basics - Basic Design, Test, Code (using objects)

For each of the following problems complete an Input-Process-Output Design, Test Table, Java Program. They all require using some of the following:

- Java provided static classes (Math, Integer, Double, NumberFormat)
- Java provided objects (Scanner, DecimalFormat, String, Random)
- User defined classes and objects

Question 1a - Input-Process-Output Design - body surface area (1 point)

Answer the following questions for the below problem. Type your answers in the text box:

1. What are the inputs?
 - What format are they in? (e.g., integer, floating-point, character, or string)
 - Any valid or invalid values? (e.g. positive, negative, valid range, certain values)
 - How do you get them? (e.g., prompt user or read from file)
2. How do you get from inputs to the outputs you want (process)?
 - What are the calculation steps?
 - To follow these steps, what else do you need?
 - Other variables, constants, conversion? (besides input and output variables)
 - Libraries (e.g., for calculations).
3. What are the outputs?
 - What format are they in?
 - How do you output them? (e.g., on screen or to a file)

Write a program that calculates a person's body surface area (BSA) and cost to buy enough material to make a tight-fitting superhero costume to cover their body. The equation commonly used in medicine to compute your body surface area, or the "skin your in" is

$$\text{body surface area} = \text{weight}^{0.425} * \text{height}^{0.725} * 0.007184$$

which yields the body surface area in square meters (m²) where the user inputs their **integer** weight in kilograms (kg) and their **integer** height in centimeters (cm). The user also will input a **real number** cost for spandex in dollars/m². The program will output the person's body surface area (to five decimal places with trailing zeroes) and spandex cost (formatted correctly in dollars and cents), one to a line with no message.

Question 1b - Test Table - body surface area (1 point)

Create a test plan with sample data for the below problem. Make a table like this with enough rows. Recall that types of test cases to consider:

- for each different possible output
- for either side of a boundary, and on the boundary, implied in the problem
- for common known answers (like 32 F = 0 C)
- for low, average and high input values

At this time, we are not writing test cases for invalid input values. We are just identifying those in the Input-Process-Output design.

Test Case Reason	Sample Data	Expected Result (manually calculate)

Write a program that calculates a person's body surface area (BSA) and cost to buy enough material to make a tight-fitting superhero costume to cover their body. The equation commonly used in medicine to compute your body surface area, or the "skin your in" is

$$\text{body surface area} = \text{weight}^{0.425} * \text{height}^{0.725} * 0.007184$$

which yields the body surface area in square meters (m²) where the user inputs their **integer** weight in kilograms (kg) and their **integer** height in centimeters (cm). The user also will input a **real number** cost for spandex in dollars/m². The program will output the person's body surface area (to five decimal places with trailing zeroes) and spandex cost (formatted correctly in dollars and cents), one to a line with no message.

Question 1c - Write Code - body surface area (4 points)

Implement your Input-Process-Output Design in Java. Use Scanner for user input (no prompting) and format your output to correct number of decimal places, if applicable. Make sure to utilize these programming concepts correctly when appropriate:

- comments
- descriptive identifier names
- correct data types
- unit conversion
- String literals for output using System.out.println()
- constants
- order of operations
- mixed type arithmetic (implicit type casting)
- % operator (mod or remainder)
- explicit type casting

Make sure to test your code on all the test cases provided.

Write a program that calculates a person's body surface area (BSA) and cost to buy enough material to make a tight-fitting superhero costume to cover their body. The equation commonly used in medicine to compute your body surface area, or the "skin your in" is

$$\text{body surface area} = \text{weight}^{0.425} * \text{height}^{0.725} * 0.007184$$

which yields the body surface area in square meters (m²) where the user inputs their **integer** weight in kilograms (kg) and their **integer** height in centimeters (cm). The user also will input a **real number** cost for spandex in dollars/m². The program will output the person's body surface area (to five decimal places with trailing zeroes) and spandex cost (formatted correctly in dollars and cents), one to a line with no message.

See BSA.java

EXPECTED OUTPUT (for various inputs):

```
Test Case 1      88      175      9.0
2.03692
$18.33
```

```
Test Case 2      75      165     10.50
1.82366
$19.15
```

```
Test Case 3     107     190     22.50
2.34937
$52.86
```

```
Test Case 4      56     172     27.25
1.65999
$45.23
```

Question 2a - Input-Process-Output Design – quadratic (1 point)

Answer the following questions for the below problem. Type your answers in the text box:

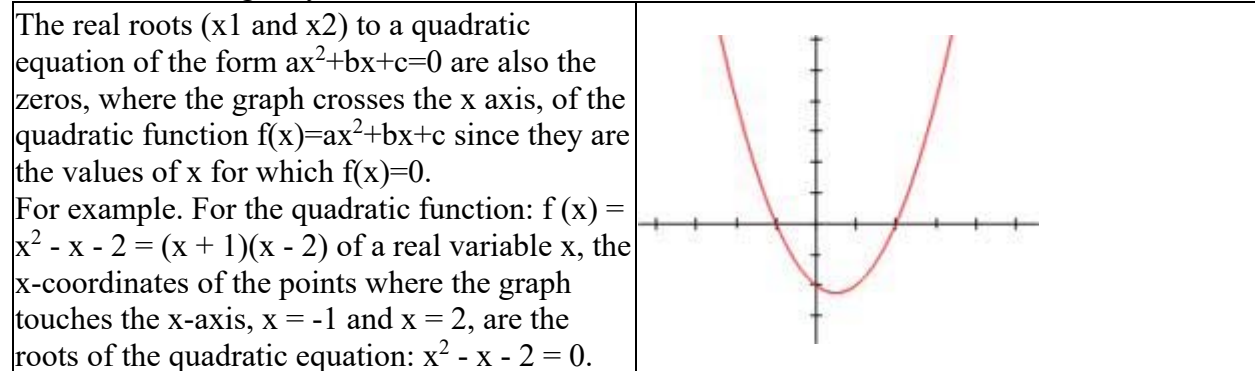
1. What are the inputs?
 - What format are they in? (e.g., integer, floating-point, character, or string)
 - Any valid or invalid values? (e.g. positive, negative, valid range, certain values)
 - How do you get them? (e.g., prompt user or read from file)
2. How do you get from inputs to the outputs you want (process)?
 - What are the calculation steps?
 - To follow these steps, what else do you need?
 - Other variables, constants, conversion? (besides input and output variables)
 - Libraries (e.g., for calculations).
3. What are the outputs?
 - What format are they in?
 - How do you output them? (e.g., on screen or to a file)

We will be writing a program that finds the two real number solutions (roots) to a quadratic equation of the form $ax^2+bx+c=0$ given the user inputting three **integer** coefficients a, b, and c (assume no imaginary roots). Use the quadratic formula.

$$x1=(-b+\text{squareroot}(b^2-4ac))/2a$$

$$x2=(-b-\text{squareroot}(b^2-4ac))/2a$$

Assume the user will input values for a, b, and c so that b^2-4ac is greater than 0 so we have real solutions, not imaginary.



Let's now assume you are building a fence between these two points where the parabola crosses the x-axis, the user also will input a **real number** cost for the fence in dollars/m². So after calculating the two roots, also calculate the cost to build a fence between the two roots on the x-axis. The program will output, one to a line, the two roots (to five decimal places with trailing zeroes) and the fence cost (formatted correctly in dollars and cents).

Question 2b - Test Table – quadratic (1 point)

Create a test plan with sample data for the below problem. Make a table like this with enough rows. Recall that types of test cases to consider:

- for each different possible output
- for either side of a boundary, and on the boundary, implied in the problem
- for common known answers (like 32 F = 0 C)
- for low, average and high input values

At this time, we are not writing test cases for invalid input values. We are just identifying those in the Input-Process-Output design.

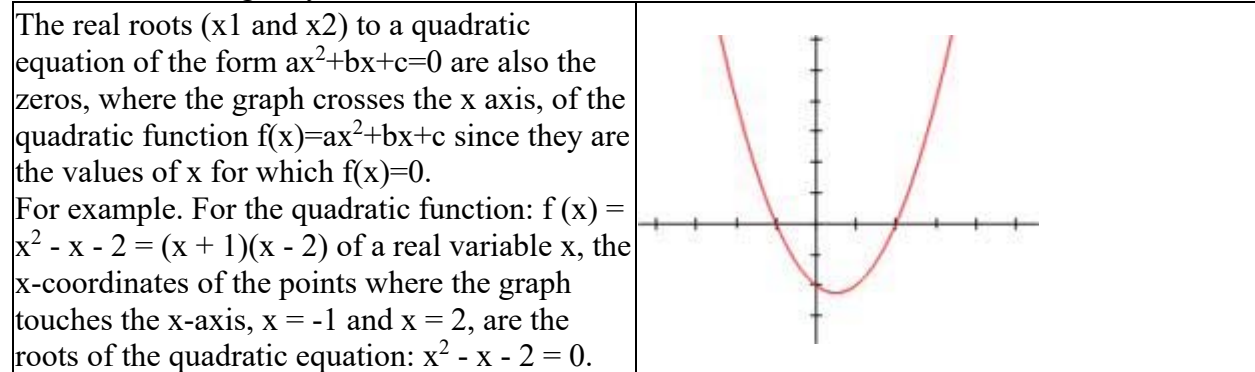
Test Case Reason	Sample Data	Expected Result (manually calculate)

We will be writing a program that finds the two real number solutions (roots) to a quadratic equation of the form $ax^2+bx+c=0$ given the user inputting three **integer** coefficients a, b, and c (assume no imaginary roots). Use the quadratic formula.

$$x1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Assume the user will input values for a, b, and c so that $b^2 - 4ac$ is greater than 0 so we have real solutions, not imaginary.



Let's now assume you are building a fence between these two points where the parabola crosses the x-axis, the user also will input a **real number** cost for the fence in dollars/m². So after calculating the two roots, also calculate the cost to build a fence between the two roots on the x-axis. The program will output, one to a line, the two roots (to five decimal places with trailing zeroes) and the fence cost (formatted correctly in dollars and cents).

Question 2c - Write Code – quadratic (4 points)

Implement your Input-Process-Output Design in Java. Use Scanner for user input (no prompting) and format your output to correct number of decimal places, if applicable. Make sure to utilize these programming concepts correctly when appropriate:

- comments
- descriptive identifier names
- correct data types
- unit conversion
- String literals for output using System.out.println()
- constants
- order of operations
- mixed type arithmetic (implicit type casting)
- % operator (mod or remainder)
- explicit type casting

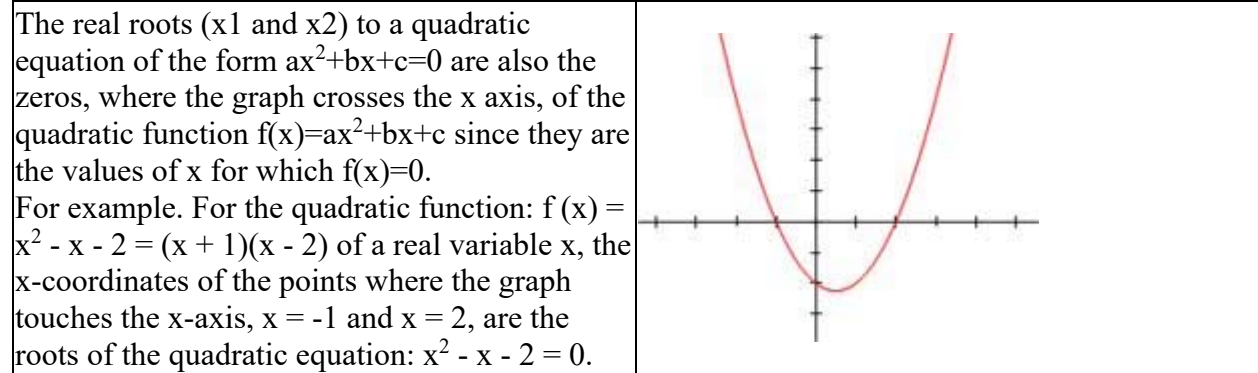
Make sure to test your code on all the test cases provided.

We will be writing a program that finds the two real number solutions (roots) to a quadratic equation of the form $ax^2+bx+c=0$ given the user inputting three **integer** coefficients a, b, and c (assume no imaginary roots). Use the quadratic formula.

$$x1 = (-b + \sqrt{b^2 - 4ac}) / 2a$$

$$x2 = (-b - \sqrt{b^2 - 4ac}) / 2a$$

Assume the user will input values for a, b, and c so that $b^2 - 4ac$ is greater than 0 so we have real solutions, not imaginary.



Let's now assume you are building a fence between these two points where the parabola crosses the x-axis, the user also will input a **real number** cost for the fence in dollars/m². So after calculating the two roots, also calculate the cost to build a fence between the two roots on the x-axis. The program will output, one to a line, the two roots (to five decimal places with trailing zeroes) and the fence cost (formatted correctly in dollars and cents).

See Quadratic.java

EXPECTED OUTPUT (for various inputs):

Test Case 1: 1 -1 -2 47.50
 2.00000
 -1.00000
 \$142.50

Test Case 2: 1 7 10 10
 -2.00000
 -5.00000
 \$30.00

Test Case 3: 1 -7 12 12.50
 4.00000
 3.00000
 \$12.50

Test Case 4: 1 9 0 13.75
 .00000
 -9.00000
 \$123.75

C - Object-Oriented Programming in Java Basics - Updating and Testing a User Defined Class – Room (6 points)

Add and test two new methods for the user defined Room class given. The Room class ensures the three dimensions of the room are positive, and provides constants for the length and width of a window and length and width of a door. You should not make any other changes to the Room attributes or methods.

- add and test a new method, `getWallArea(int windowCount, int doorCount)`, that returns a double, the calculate area of the walls of the room. You should subtract the area of the windows and doors. You can assume the windowCount and doorCount are positive and limited by logical limits for the number of windows and doors a room can have.
- add and test a new method, `getMoldingLength(int windowCount, int doorCount)`, that returns a double, the total length of the baseboard around the room (minus the holes for the door), plus three sides of each door (two sides and top), plus 4 sides of each window. You can assume the windowCount and doorCount are positive and limited by logical limits for the number of windows and doors a room can have.

Make sure to test your code on all the test cases provided.

See Room.java and RoomTest.java