**Java - Programming Assignments**

**Objectives**

1. To give students practice calling functions from the math library.

2. To learn how to use the if statement to solve problems.

**Problem A: Land Enclosure**

You are a farmer and need a certain amount of land in the shape of a rectangle. In particular, the information you know about the enclosure of land you are buying is how much area (in square feet) the enclosure will comprise and the ratio of the length to the width of the enclosure. Given this information, the farmer needs you to write a program for him to determine the length and width of the fence (in feet) that is needed for the enclosure.

**Input Specification**

1. The area of the enclosure will be a positive real number less than 1,000,000,000.

2. The ratio of the length to the width of the enclosure will be expressed as two positive integers L:W, where both values are less than 100.

**Output Specification**

Output the following three values: length of the enclosure, width of the enclosure, and the total amount of fence needed for the enclosure. All three values should be rounded to two decimal places. Please format your output as follows:

Your enclosure has dimensions X feet by Y feet.

You will need Z feet of fence total.

where X is the length of the enclosure, Y is the width, and Z is the perimeter of the enclosure, all rounded to two decimal places.

**Output Sample**

Below is one sample output of running the program. Note that this sample is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

**Sample Run**

**What area do you need for your enclosure in square feet?**

*300*

**What is the ratio of the length to the width of your enclosure?**

***3 1***

Your enclosure has dimensions 30.00 feet by 10.00 feet.

You will need 80.00 feet of fence total.

**Problem B: Pay Calculator**

Your aunt runs a business selling knives. She hires door-to-door salespeople in her business and has created a system of pay to give the salespeople great incentives for excellent performance. In particular, the following chart shows how much each salesperson gets paid per sale, depending on how many sales he or she has made in the month:

|  |  |
| --- | --- |
| Number of Knives Sold | Pay |
| Less than or equal to 100 | $5 |
| In between 101 and 200 | $10 |
| More than 200 | $15 |

If a salesperson sells 70 knives in a month, he receives 70x$5 = $350.

If a salesperson sells 120 knives in a month, he receives 100x$5 + 20x$10 = $700, since he receives $10 for the last 20 knives he sold.

If a salesperson sells 400 knives in a month, he receives 100x$5 + 100x$10 + 200x$15 = $4500 for the month.

Your aunt has asked you to write a program that asks the salesperson how many knives they sold for the month and prints out how much he gets paid for the month.

**Input Specification**

1. The number of knives a salesperson sells in a month will be a non-negative integer less than 1,000,000.

**Output Specification**

Output the salesperson’s pay for the month in the following format:

You made $X selling knives this month.

where X is the number of dollars the salesperson made for the month.

**Output Samples**

Below is one sample output of running the program. Note that this sample is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

**Sample Run**

**How many knives did you sell this month?**

*70*

**You made $350 selling knives this month.**

**Problem C: Best Deal**

It’s the end of the term and you have some money left on your meal card. Unfortunately, if you don’t spend the money before you leave for home, you’ll lose it. Your job is to spend as much of the money left on the card as possible. When you go to the convenience store on campus where you are allowed to use your card, you see three items that are on sale that you would potentially use. There is only one of each item, so you can’t buy more than one of a particular item. Given the prices of each of the three items, determine what is the least amount of money you have to leave on your meal card. For example, if you have $20 left on your card and the three items cost $7, $10, and $14, then the least amount of money you can leave on your card is $3. You can achieve this by buying the item that costs $7 and the item that costs $10.

**Input Specification**

1. All four input values, the amount left on the meal card and the cost of each of the three items will be positive integer (in dollars) less than 1000.

**Output Specification**

Output a single line with the following format:

The least amount of money you can leave on your card is $X.

**Output Samples**

Below is one sample output of running the program. Note that this sample is NOT a comprehensive test. You should test your program with different data than is shown here based on the specifications given above. In the sample run below, for clarity and ease of reading, the user input is given in *italics* while the program output is in bold.

**Sample Run**

**How much money is left on your meal card (in dollars)?**

*20*

**What is the cost in dollars of the three items you may buy?**

*7 10 14*

**The least amount of money you can leave on your card is $3.**

**Note: The cost of the three items may be entered in any order. In particular, they may not be sorted in ascending order as shown in this sample.**

**Deliverables**

One Project with all three specifications from above. You will submit a design tool that will sit in your Java SRC folder in your project. You will zip up your entire project and submit it to Blackboard.

**Grading Details**

Your programs will be graded upon the following criteria:

1) Your correctness

2) Your programming style and use of white space. Even if you have a plan and your program works perfectly, if your programming style is poor or your use of white space is poor, you could get 10% or 15% deducted from your grade.