Computer Science 121

Lab 6

In this lab, you have four questions. You will practice writing programs using conditional operators.

You need to finish question1 and 2 in the lab and demonstrate to your instructor. You need to submit question 3 and 4 in e-campus before the next lab and you can demonstrate during the next lab.

Also include the following comment at the beginning of your code:

```
/*
* Your Name
* Submission Date
* CS 121 Lab
* Name of Program
*
* Describe what the program does in one to two sentences.
*/
```

Q1. **Palindrome (25 points)**

Problem Description

Write a program that prompts the user to enter a five-digit integer and determines whether it is a palindrome integer.

An integer is palindrome if it reads the same from right to left and from left to right. So if it is a five-digit integer, the first digit must be same as last (or the fifth) digit and second digit must be same as the secondlast (or the fourth) digit.

For example: 12321, 25652, 45654

The program prompts the user to input a 5 digit integer and if the number is palindrome print that the given number is palindrome. If the number is not palindrome output which of the digits are not matching. If both are not matching, print that both are not matching.

Some sample runs:

```
> run Palindrome
Enter a five-digit integer:  96569

96569 is a palindrome
>
```

```
> run Palindrome
Enter a five-digit integer:  19593

19593 is not a palindrome
First and last digit do not match
>
```

```
> run Palindrome
Enter a five-digit integer:  12351

12351 is not a palindrome
Second and second last digit do not match
>
```

```
> run Palindrome
Enter a five-digit integer:  12345

12345 is not a palindrome
None of the digits match.
>
```

Q2. **CompareCosts (25 points)**

Problem Description

Suppose you want to shop for a product. The product is available in three stores but in different packages and different prices. You would like to write a program to compare the cost and find the product with highest cost per lbs. among the three to avoid from that store. The cost might be high in any one store, any two stores or all the three might have same cost. So your program should handle these conditions:

1) Price/lbs. is highest in one of the stores
2) Price/lbs. is same for two stores and higher than third one
3) Price/lbs is same for all stores.

The program prompts the user to enter the weight in pounds (lbs.) and price of each package in dollar ($) and displays the one with the highest price. Remember we need to compare the price per pound to get the actual price.

Some example runs:

```
> run CompareCosts
Enter weight (lb) and price ($) for shop 1: 10
60
Enter weight(lb) and price ($) for shop 2: 10
50
Enter weight (lb) and price ($) for shop 3: 10
50
Shop1 has highest price.
```

```
> run CompareCosts
Enter weight (lb) and price ($) for shop 1: 12.23
20
Enter weight(lb) and price ($) for shop 2: 15
30
Enter weight (lb) and price ($) for shop 3: 30
60
Shop 2 and Shop3 both have highest price
>
```

```
> run CompareCosts
Enter weight (lb) and price ($) for shop 1: 10
50
Enter weight(lb) and price ($) for shop 2: 10
50
Enter weight (lb) and price ($) for shop 3: 10
50
All shops have the same price
```

Q3. **Taxes (25 points)**

Create a new Java program, and name the class `Taxes`. We are given the following tax schedule:

If your status is	and if the taxable income is greater than	but not greater than	the tax is	of the income over
Single(0)	$0	$8,000	10%	$0
	$8,000	$32,000	$800 + 15%	$8,000
	$32,000		$4,400 + 25%	$32,000
Married(1)	$0	$16,000	10%	$0
	$16,000	$64,000	$1,600 + 15%	$16000
	$64,000		$8,800 + 25%	$64,000

For example, if the user is single, and the income is $15,000, the user's tax would be

$$\$800 + 15\% * (\$15,000 - \$8,000)$$

$$= \$800 + 15\% * \$7,000$$
$$= \$800 + \$1,050$$
$$= \$1,850$$

We are interested in translating the above tax schedule into a Java program. The program should prompt the user to enter the marital status (single or married) as an `integer` where 0 represents single, and 1 represents married. The program should also prompt the user to enter the taxable income. The program should output the total amount of tax. Make sure to print error message if the user inputs invalid status (any status not 0 or 1).

Here are some sample runs:

```
> run Taxes
Enter the status Single (0) or Married (1)): 1

Enter the income: 90000
The amount owed in taxes is 15300.0
```

```
> run Taxes
Enter the status Single (0) or Married (1)): 0

Enter the income: 15000
The amount owed in taxes is 1850.0
```

```
> run Taxes
Enter the status Single (0) or Married (1)): 5

Enter the income: 20000
Invalid status. Please input correct status.
The amount owed in taxes is 0.0
```

Q4.ScissorRockPaper.java (25 points)

Problem Description

Write a program that plays the popular scissor-rock-paper game. The rule of the game is as follows:

A scissor can cut a paper, a rock can knock a scissor and a paper can wrap a rock. So scissor wins paper, rock wins scissor and paper wins rock.

The program randomly generates a number 0, 1 or 2 representing scissor, rock and paper. The program prompts the user to enter a number 0, 1 or 2 and displays a message indicating whether the user or the computer wins, loses or draws.

Some example runs:

```
> run ScissorRockPaper
Enter your choice: scissor (0), rock (1), paper (2): 1
The computer is paper. You are rock. You lost>
```

```
> run ScissorRockPaper
Enter your choice: scissor (0), rock (1), paper (2): 0
The computer is scissor. You are scissor too. It is a draw>
```

```
> run ScissorRockPaper
Enter your choice: scissor (0), rock (1), paper (2): 5
Invalid choice.
>
```