CS 5000 – Spring 2020
Assignment #2

Selections – Chapter 3

Develop a complete Java program for each of the following problems. Please name the programs as indicated and add proper program headers and output labels as shown below. ***Please use only concepts and programming constructs/syntax we discuss to date.***

Make sure you **include a header for each program (replace the dots in the header section with your full name and the name of your IDE, such as Jgrasp).**

```
// Class:      CS 5000
// Term:       Spring 2020
// Name:       ...
// Instructor: Dr. Haddad
// Assignment: 2
// IDE Name:   ...
```

Program #1 (10 points): Write a Java program (name it `IncomeTax`) that reads annual income from the user (as integer value) and then, on separate lines, prints out the entered income amount along with the tax bracket (using the following tax table) and the tax amount:

```
            Annual income <= $30,000       tax bracket: 3%
$30,000  <  Annual income <= $70,000       tax bracket: 10%
$70,000  <  Annual income <= $150,000      tax bracket: 15%
$150,000 <  Annual income <= $300,000      tax bracket: 20%
$300,000 <  Annual income <= $900,000      tax bracket: 35%
$900,000 <  Annual income                  tax bracket: 40%
```

Make sure to properly label your output, format the output (using $ and %), and use escape character (\t) to line-up the outputs after the labels as shown below. Document your code; and organize and properly space out the outputs. For the sample output below, notice that tax calculation is 30k at 3%, 40k at 10%, and 50k at 15%, that is $900+$4,000+$7,500)). The sample test data below does not show the input prompts.

```
Your income:          $120000
Your tax bracket:     15%
Your tax amount:      $12400
```

Program #2 (10 points): Write a Java program (name it `PhoneBill`) that calculates the bill for a cellular telephone company. The company offers two types of service: regular and premium. The rates vary depending on the type of service. The rates are computed as follows:

Regular service: $20.00 fee plus first 50 minutes free. Charges for over 50 minutes are computed at the rate of $0.20 per minute.

Premium service: $10.00 fee plus:

a. For daytime calls (between 6:00AM to 6:00PM), the first 75 minutes are free; charges for over 75 minutes are computed at the rate of $0.10 per minute.

b. For nighttime calls (between 6:00PM to 6:00AM), the first 100 minutes are free; charges for over 100 minutes are computed at the rate of 0.05 per minute.

Your program should prompt the user to enter an account number, a service code (type char), and the number of minutes the service was used. A service code **r** (or **R**) means regular service; while code **p** (or **P**) means premium service.

For the premium service (code **P**), the customer may be using the service during both the day and at night. Therefore, you must ask the user to input the number of minutes used during daytime and nighttime.

The program should output, on separate lines and with proper labels, the account number, service type, number of minutes the service was used, and the bill amount due from the user as shown below. Sample test data below does not show the input prompts.

First test:

```
Account Number:   12345
Service type:     Regular
Total minutes:    51
Amount due:       20.20
```

Second test:

```
Account Number:    1234567
Service type:      Premium
Daytime minutes:   77
Nighttime minutes: 102
Amount due:        $10.30
```

Program #3 (10 points): The concept of a 5-digits palindrome number is that its digits read the same from left to right and from right to left. For example, 12121, 45454, 14741, etc… Write a java program that reads a 5-digit number from the user (as integer value) and then determines whether the entered number is a palindrome or not. (**Do not use loops or treat the entered number as string of characters**). Manipulate the input number **mathematically** using proper math operators (division and remainder) to determine if it is a palindrome or not. An input of less than or greater than 5 digits should be rejected as invalid input before being processed. Name the program `PalindromNumber`. Document your code, use proper label for the input prompt (e.g., *Please enter a 5-digits integer value:*), and display the outputs similar to the following examples, Use escape character (\t) to line up the outputs after the labels as shown below. Sample test data below does not show the input prompts.

First test:

```
Input value:  122
Judgment:     Invalid input, must be 5 digits number.
```

Second test:

```
Input value:  14127
Judgment:     Not Palindrome
```

Third run:

```
Input value:  94249
Judgment:     Palindrome
```

Fourth run:

```
Input value:  164461
Judgment:     Invalid input, must be 5 digits number.
```

Program #4 (10 points): Write a Java program to determine the best deal when buying a small box of oranges vs. a large box of oranges. The program asks the user to enter the weight and price of each box and then determines which box has the best better value. The boxes may have the same value. Name the program `BestOrangeDeal`. Sample runs are shown below. Document your code, and properly label the input prompts and the outputs. Use escape character (\t) to format the outputs as shown below. Sample test data below does not show the input prompts.

First test:

```
Small box weight:    5 Pounds
Small box price:     10 Dollars
Large box weight:    12 Pounds
```

```
Large box prices:    18 Dollars
Judgment:            The large box is a better deal
```

Second test:

```
Small box weight:    5 Pounds
Small box price:     10 Dollars
Large box weight:    5 Pounds
Large box prices:    10 Dollars
Judgment:            Both boxes are of the same value
```

Third test:

```
Small box weight:    5 Pounds
Small box price:     10 Dollars
Large box weight:    10 Pounds
Large box prices:    28 Dollars
Judgment:            The smaller box is a better deal
```

Program #5 (10 points): Write a Java program to determine if a circle is either completely inside, overlapping with, or completely outside another circle. The program asks the user to enter the center point (X1, Y1) and the radius (R1) for the first circle C1, and the center point (X2, Y2) and the radius (R2) for the second circle C2. The program then determines if the second circle C2 is either completely inside, or overlapping with, or completely outside the first circle C1. Hint: use the sum of R1 and R2 and the distance between the centers to solve the problem. Name the program `Circles`. Follow the sample runs below to format your outputs. Document your code, properly label the inputs prompts, and organize the outputs using escape character (\t) to display as shown below. Sample test data below does not show the input prompts.

First test:

```
Circle 1 center:   (0,0)
Circle 1 radius:   6
Circle 2 center:   (1,1)
Circle 2 radius:   1
Judgment:          Circle 2 is completely inside circle 1
```

Second test:

```
Circle 1 center:   (0,0)
Circle 1 radius:   2
Circle 2 center:   (7,7)
Circle 2 radius:   1
Judgment:          Circle 2 is completely outside circle 1
```

Third test:

```
Circle 1 center:   (0,0)
Circle 1 radius:   3
Circle 2 center:   (1,1)
Circle 2 radius:   2
Judgment:          Circle 2 is overlapping with circle 1
```

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines on the course webpage. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment is due no later than **5:00pm** on the due day posted in D2L.
3. Please compile and run your java files (only the .java files) right before you upload to the assignment submission folder in D2L.