CS 5000 – Summer 2025
Assignment #2, 50 Points

Selections – Chapter 3

Note: If you re-upload the files, you must re-upload **ALL** files as the system keeps the most recent uploaded submission only. No zip files!

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:       Summer 2025
// Name:       ...
// Instructor: Dr. Haddad
// Assignment: 2
// IDE Name:   ...
```

Program #1 (17 points): Write a Java program, named ***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) followed by 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 the 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. Make sure your code displays the outputs following the test data format.

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

Program #2 (17 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, named ***PalindromeNumber***, 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. 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. Make sure your code displays the outputs following the test data format.

First test:

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

Second test:

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

Third test:

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

Fourth test:

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

Program #3 (16 points): Write a Java program, named ***BestAppleDeal***, to determine the best deal when buying a small box of apples vs. a large box of apples. 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. 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. Make sure your code displays the outputs following the test data format.

First test:

```
Small box weight:    5 Pounds
Small box price:     10 Dollars
Large box weight:    12 Pounds
Large box price:     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:    8 Pounds
Large box price:     16 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 price:     28 Dollars
Judgment:            The smaller box is a better deal.
```

Submission:

1. Before submitting your programs, make sure you review the assignment submission requirements and grading guidelines posted in D2L. The grading guidelines explain some of the common errors found in programming assignments.
2. The assignment due date is posted in D2L.
3. Please compile, run, and test your code right before you upload your java files to the assignment submission folder in D2L.
4. Please upload only the .java files (total 3 files).