WIT COMP1000 Computer Science I

Prof. Thai

Student:

**Lab5: Loops**

1. **Polygon** (Polygon.java)

Write a program to compute the area, perimeter, and interior angle of a regular polygon. First, have the user supplies and **the side length (any positive number**).

Then, compute the **area** via the following equation:



where *n* is the number of sides and *s* is the side length. Note that both equations are equivalent, with the first equation using degrees and the second equation using radians.

Then, compute the **perimeter** via the following equation:



and then the **interior angle** (in degrees) via the following equation:



A sample run of your program would look like the following:

Enter number of sides: 3

Enter side length: 1

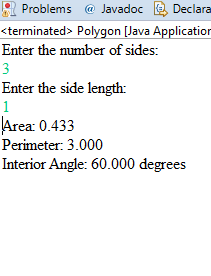
Area: 0.433

Perimeter: 3.000

Interior Angle: 60.000 degrees

You are required to generate the outputs using exactly three decimal places (rounding where necessary). Tips: (i) the **Math.tan()** method will compute the tangent of a value in radians, (ii) the **Math.toRadians()** method will convert a value in degrees to radians, and (iii) the **Math.PI** constant can be used for pi (π).

TAKE A SCREENSHOT of the output console showing the sample run above and paste it here.

****

1. **Inflation** (Inflation.java)

Write a program that deals with inflation, which is essentially the rising cost of general goods over time. That is, the price of goods, such as a packet of cookies, goes up as time goes by. So, you will write a program to gauge the expected cost of an item in a specified number of years.

The program asks for the cost of the item, the number of years, and the rate of inflation. The output is the estimated cost of the item after that number of years, using the given inflation rate. The user enters the inflation rate as a percentage, for example 4.5. You will have to convert the percentage to a fraction (like 0.045), and then use a loop to estimate the item's price adjusted for inflation.

Note that this is the same way compound interest on a credit card account or a mortgage is calculated. You must check each of the values provided by the user to make sure that they are reasonable. Finally, you need to print out the price with exactly two places after the decimal (for the cents) after your calculations are done

To adjust the price for inflation, you need to increase the price by the inflation rate each year. For example, if you have an item that is initially $10, with inflation rate of 10%, the adjusted prices will be:

* After 1 year: $10.00 ∗ (1 + 0.10) = $11.00
* After 2 years: $11.00 ∗ (1 + 0.10) = $12.10
* After 3 years: $12.10 ∗ (1 + 0.10) = $13.31

…

In other words, to calculate the price after another year, you must use the value from the current year, NOT the original value. To do this, you must use a loop. An example of what your program should output:

Enter the current price of the item: $10

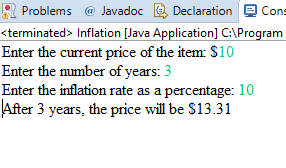
Enter the number of years: 3

Enter the inflation rate as a percentage: 10

After 3 years, the price will be $13.31

You need to test for invalid inputs, such as negative price, negative year, and negative interest rate.

TAKE A SCREENSHOT of the output console showing the sample run above and paste it here.

****

1. **Guessing** (Guessing.java)

Write a program that plays a guessing game with the user. Specifically, your program should randomly pick a number between 1 and 100. Then, ask the user for a guess. You should verify and tell the user if the guess is not a valid guess. Otherwise, tell the user their guess was too high or too low. The program should continue to prompt the user for new guesses until they get the correct number, telling them each time if the guess was too high or too low or invalid.

A template of the main program with the random number generator is provided to you. Notice that the *target* variable contains the value that the user is guessing.

Here’s a sample run of a working version of a program:

Enter your guess (between 1 and 100): 50

Too high!

Enter your guess (between 1 and 100): 0

Invalid guess, try again!

Enter your guess (between 1 and 100): 101

Invalid guess, try again!

Enter your guess (between 1 and 100): 25

Too high!

Enter your guess (between 1 and 100): 12

Too high!

Enter your guess (between 1 and 100): 6

Too high!

Enter your guess (between 1 and 100): 3

Too low!

Enter your guess (between 1 and 100): 4

Too low!

Enter your guess (between 1 and 100): 5

You win!

TAKE A SCREENSHOT of the output console showing the sample run above and paste it here.

