

CS-200: Programming I
Fall 2017
Northeastern Illinois University
PLTL: Week of 09/11/17
Selection Statements/Math

Practice Problem #1

- Write a program that has the class name `IsFactor` and that has the `main` method.
- The program should ask the user to enter two integers greater than 1.
- The program should determine whether the smaller of the two integers is a factor of the larger of the two integers (regardless of which order the two numbers are entered in).
- Based on whether either one is a factor of another, print out the appropriate message.
- As a reminder, a factor of one number is any whole number that can divide the number into evenly. For example, 8 can divide 64 evenly, therefore 8 is a factor of 64.
- Several sample runs are provided for you below. Format your output to match the sample output.

```
Enter the first number: 5
Enter the second number: 65
5 is a factor of 65
```

```
Enter the first number: 35
Enter the second number: 2
2 is not a factor of 35
```

Practice Problem #2

- Write a program that has the class name `RightAngledTriangle` and that has the `main` method.
- The program should ask the user to enter three numbers `a`, `b` and `c` and check if they are the sides of right angled triangle.
- Using the pythagorean theorem, you can check if it is a right angled triangle or not. If it is a right angled triangle then print `Right angled triangle` if it is not then print `Not a right angled triangle`.
- As per the theorem the sum of the square of two sides(legs) is equal to the square of the longest side(hypotenuse).
- Below is the formula given to calculate the hypotenuse and you must use `Math.Sqrt` and `Math.pow` to calculate it. check the style with prof.

$$h = \sqrt{b^2 + p^2}$$

- Several sample runs are provided below. Format your output to match the sample output.

```
Enter s1: 3
Enter s2: 5
Enter s3: 4
Right angled triangle
```

```
Enter s1: 9
Enter s2: 5
Enter s3: 12
Not a right angled triangle
```

```
Enter s1: 4.4
Enter s2: 3.3
Enter s3: 5.5
Right angled triangle
```

Practice Problem #3

- Write a program that has the class name EvenOrOdd and that has the main method.
- The program should ask the user to enter a 4 bit binary number (1's & 0's) and then convert it into the decimal number.
- For example, if the input is 1101, the output should be 13 (Hint: Break the number into digits and then convert each digit to a value for a single digit)
- If the digits are n_1, n_2, n_3 and n_4 , the decimal equivalent is $8n_1 + 4n_2 + 2n_3 + 1n_4$.
- After you get the decimal value of the 4 bit number, print out the value and check if the value is even or odd and print accordingly.
- Several sample runs are provided for you below. Format your output to match the sample output.

```
Enter a 4-bit integer: 0011
The decimal value of 0011 is 3
3 is an odd number
```

```
Enter a 4-bit integer: 1010
The decimal value of 1010 is 10
10 is an even number
```