Java – Input with Scanner

Purpose

These labs were designed to introduce you to reading data from the user to perform calculations. Be sure to import the Scanner class(import java.util.Scanner;) and declare an object by

Scanner kb = new Scanner(System.in);

Always use meaningful names like kb, keyboard, or input.

3.01 Arithmetic.java

Write a program that reads 3 integers from the user and outputs the sum, product and average. As a bonus(optional), print out the largest and smallest numbers.

Sample Execution

```
Enter first number: 57
Enter second number: 3
Enter third number: 24

Numbers: 57, 3, 24
Sum: 84
Product: 4104
Average Rounded Down: 28
Smallest: 3
Largest: 57
```

3.02 Name.java

Write a program that reads your full name and outputs it in First MI Last format. The Scanner class does not have a nextChar() method so read the entire middle name as a String. The String method charAt(int) will return the character at the specified index. "Hello".charAt(0) returns 'H'.

Sample Execution

```
Enter your first name: Joseph
Enter your middle name: Robinette
Enter your last name: Biden
Joseph R Biden
```

3.03 Years.java

Write a program that calculates the number of years(assume 365 days) and days given an input of minutes. Use % and /.

Sample Execution

```
Enter number of mins: 5000000000
5,000,000,000 minutes is 9512 years and 342 days
```

3.04 MPH.java

Write a program that calculates the rate of travel given the distance(in miles), hours and minutes. The formula is distance = rate * time. Produce the output below(bonus: use a loop).

Sample Execution

```
Enter the distance :: 45
Enter the hours :: 0
Enter the minutes :: 32
45 miles in 0 hour and 32 minutes = 84.375 MPH.

Enter the distance :: 96
Enter the hours :: 1
Enter the minutes :: 43
96 miles in 1 hour and 43 minutes = 55.922 MPH.

Enter the distance :: 100
Enter the hours :: 2
Enter the minutes :: 25
100 miles in 2 hour and 25 minutes = 41.379 MPH.

Enter the distance :: 50
Enter the hours :: 2
Enter the minutes :: 25
50 miles in 2 hour and 25 minutes = 20.690 MPH.
```

3.05 CMYKtoRGB.java

The primary color format for LCD displays, digital cameras, and web pages—known as the RGB format—specifies the level of red (R), green (G), and blue (B) on an integer scale from 0 to 255. The primary format for publishing —known as the CMYK format—specifies the level of cyan (C), magenta (M), yellow (Y), and black (K) on a real scale from 0.0 to 1.0.

Write a program that converts from CMYK format to RGB format using these mathematical formulas:

white = 1 − black

red = 255 × white × (1 − cyan)

green = 255 × white × (1 − magenta)

blue = 255 × white × (1 − yellow)

Technically the red, green, and blue values should be rounded but for the purposes of this lab round down using casting (int).

Sample Executions

```
Enter cyan: 1
Enter magenta: 1
Enter yellow: 0
Enter black: 0
red   = 0
green = 0
blue  = 255

Enter cyan: .6
Enter magenta: .25
Enter yellow: .75
Enter black: .3
red   = 71
green = 133
blue  = 44
```