WIT COMP1000

Formatting Numeric Print Output

Printing Review

- So far you know about two methods to print the value of variables, as well as anything literal in quotes
  - `System.out.println()`
  - `System.out.print()`
- To sequence values/variables together, use the plus (+) operator
  - `System.out.println("Example: " + x);`

Numeric Output

- What if you wanted to output exactly two decimal places of a number (with rounding), or thousands separators (i.e. 1234 vs. 1,234)?
- These methods are generally useful for printing strings and integers, but Java supports even greater control when printing numbers (particularly large numbers/decimals)
  - `System.out.printf("format", arg1, arg2, …)`

Example

```
public static void main(String[] args) {

    double smallNum = 0.031752;
    int bigNum = 88452;

    System.out.println("Value = " + smallNum); // 0.031752
    System.out.printf("Value = %.3f%n", smallNum); // 0.032
    System.out.printf("Value = %.2e%n", smallNum); // 3.18e-02

    System.out.println();

    System.out.println("Value = " + bigNum); // 88452
    System.out.printf("Value = %,d%n", bigNum); // 88,452

}
```

Format String

- The format string contains literals (items you want outputted verbatim), converters, and flags
  - A converter looks to the arguments to fill in a value
    - Starts with a % and ends with a single character code
  - A flag modifies a converter with options
    - Goes between the % and the converter code
- Each time you use a converter, you must supply a corresponding argument (other than newline)

Some Converters, Flags

Converter	Flag	Description
d		An integer
f		A float (includes double)
e		A float in scientific notation.
n		New line
	+	Includes the sign (positive or negative)
	,	Includes grouping characters
	.3	Three places after the decimal.

Many more options exist:
https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html#syntax

Exercise

Write a program that asks the user for a decimal value – output that value with exactly three decimal places, rounding as necessary.

```
Enter a value: 3.14159

Rounded: 3.142
```

Answer

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    System.out.print("Enter a value: ");
    double value = input.nextDouble();
    System.out.printf("Rounded: %.3f%n", value);
}
```

Wrap Up

- When outputting numbers, the `System.out.printf` method allows you more control than `System.out.print`/ `System.out.println`

- The syntax is first a format string, then any number of arguments (with each non-newline converter having an argument)