# Assignment 02

CS 120: Software Design I

## Program 1: MakeChange

Your first program will be written in a class called `MakeChange`. When engaging in cash transactions, cashiers need to determine the appropriate quantity of each bill (e.g., $20 bill, $10 bill) to give to a customer as change. Your program will determine how many of each bill to give the customer, with the goal being to return the fewest number of bills.

Your program will first prompt the user for an `int` monetary value. The program will then return how many $20, $10, $5, and $1 bills to return; it is acceptable to put “0” for some or all bills as appropriate.

Below are four example runs of the program with input `Highlighted`. Make sure that your output matches the examples below (except for the sections determined by user input).

### Example 1

```
Input a monetary value (whole numbers): 37

1 x $20
1 x $10
1 x $5
2 x $1
```

### Example 2

```
Input a monetary value (whole numbers): 40

2 x $20
0 x $10
0 x $5
0 x $1
```

### Example 3

```
Input a monetary value (whole numbers): 4

0 x $20
0 x $10
0 x $5
4 x $1
```

### Example 4

```
Input a monetary value (whole numbers): 0

0 x $20
0 x $10
0 x $5
0 x $1
```

Some prompts and questions to consider *before* starting to write the program:

- Consider how you would calculate the smallest numbers of bills needed to make change by hand. For example, work out the answer for $52. How do you determine that you need two twenties? How do you then determine you need one ten? What operators can you use to help replicate this thought process in your program?
- What order will the prompts occur in? (e.g., do you prompt for numerator or denominator first?)
- How will you ensure that the prompt and user input are on the same line?
- How will you ensure that there is a space between the prompt and the user input?
- How will you print a blank line?

The following is a high level checklist of requirements for your program:

- □ Your class is named `MakeChange` and is placed in the project created for this assignment
- □ Your code is commented according to the guidelines in the Java Style Guide found on D2L
- □ Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- □ Your code fulfills the functionality outlined above
- □ You have all the prompts listed in the examples in the exact same order
- □ You have a blank line between the prompts and the output
- □ Your strings (both prompts and output) are formatted **exactly** as shown in the examples

## Program 2: `FindReplace`

The second program you will be writing will be in a class named `FindReplace`. Finding and replacing text across a document is a common action by users in text editors (e.g., Notepad, Microsoft Word). When finding and replacing all occurrences of some text, the editor will take in input from the user about what text to find and what to replace it with. It will then find all occurrences of that pattern and replace it, noting the number of changes made.

Your program will first prompt the user for text; this will be the text we are finding and replacing in (e.g., a sentence, a paragraph). Next, the program will prompt the user for the pattern to find, followed by the pattern to replace it with. The program will then print out the new text, with all occurrences of the pattern replaced, and note how many changes were made.

Below are four example runs of the program with input `Highlighted`. Make sure that your output matches the examples below (except for the sections determined by user input).

### Example 1

```
Enter your text: An example sentence for an example program run.

Enter the pattern you want to replace: ex
Enter what you want to replace it with: s

New text: An sample sentence for an sample program run.

”ex” replaced with ”s” 2 times.
```

### Example 2

```
Enter your text: Various CS courses: cs120 Cs220 cS340 CS364

Enter the pattern you want to replace: CS
Enter what you want to replace it with: computer science

New text: Various computer science courses: cs120 Cs220 cS340 computer
   science364

”CS” replaced with ”computer science” 2 times.
```

### Example 3

```
Enter your text: The rain in Spain falls mostly in Spain

Enter the pattern you want to replace: n S
Enter what you want to replace it with: s Pro

New text: The rain is Propain falls mostly is Propain

”n S” replaced with ”s Pro” 2 times.
```

Note that your program will only work when the pattern you want to replace and what you are replacing it with are of different lengths. Why?

Some prompts and questions to consider *before* starting to write the program:

- Examine the examples above, particularly examples 2 and 3, to understand how find/replace works with `String` values on a computer regarding a character's case.
- Note that it is acceptable to use the `replaceAll()` method from class.
- How can you count the number of replacements in the text? Hint: consider how you might use the `length()` method for `String` values to calculate the number of replacements.
- What order will the prompts occur in?
- How will you ensure that the prompt and user input are on the same line?
- How will you ensure that there is a space between the prompt and the user input?
- How will you print a blank line?
- How will you add quotation marks in the output?

The following is a high level checklist of requirements for your program:

- ☐ Your class is named `FindReplace` and is placed in the project created for this assignment
- ☐ Your code is commented according to the guidelines in the Java Style Guide found on D2L
- ☐ Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- ☐ Your code fulfills the functionality outlined above
- ☐ You have all the prompts listed in the examples in the exact same order
- ☐ You have blank lines as indicated in the examples
- ☐ Your strings (both prompts and output) are formatted **exactly** as shown in the examples

## Program 3: `Pythagoras`

Your third program will be for solving the equation associated with the Pythagorean theorem and will be in a class named `Pythagoras`. The Pythagorean theorem equation is used to calculate the length of the hypotenuse of a right triangle given the lengths of the other two sides. Your program will also find the area of the triangle. Below are the equations for the Pythagorean theorem equation and the area of a right triangle:

$$side3 = \sqrt{side1^2 + side2^2}$$

$$area = \frac{1}{2}side1side2$$

Your program will prompt the user for the `int` lengths of two sides of the triangle (neither of which are the hypotenuse) and then calculate the length of the third side (i.e., the hypotenuse). After displaying all three sides from smallest to largest, the program will then display the area of the triangle. All decimal values should be limited to at most two places.

Below are four example runs of the program with input `Highlighted`. Make sure that your output matches the examples below (except for the sections determined by user input).

**Example 1**

```
Input side 1: 3
Input side 2: 4
Side 3 is 5

The three sides are 3, 4, 5
The area is 6
```

**Example 2**

```
Input side 1: 1
Input side 2: 2
Side 3 is 2.24

The three sides are 1, 2, 2.24
The area is 1
```

**Example 3**

```
Input side 1: 24
Input side 2: 19
Side 3 is 30.61

The three sides are 19, 24, 30.61
The area is 228
```

Some prompts and questions to consider *before* starting to write the program:

- How will you ensure at most two decimal places are printed for the output values?
- How will you determine the order of the three sides from smallest to largest? What method(s) learned in class can help you with this?
- Consider the equation for calculating the area of the triangle. Work out the equation using $side1 = 3$ and $side2 = 3$, paying careful attention to integer operations. The expected answer is 4.5; why does your calculation give you the incorrect answer? How can you enter this equation into your program to ensure you receive the correct answer? (there is more than one way to do this)
- What order will the prompts occur in?
- How will you ensure that the prompt and user input are on the same line?
- How will you ensure that there is a space between the prompt and the user input?
- How will you print a blank line?

The following is a high level checklist of requirements for your program:

- □ Your class is named `Pythagoras` and is placed in the project created for this assignment
- □ Your code is commented according to the guidelines in the Java Style Guide found on D2L
- □ Your code is formatted according to the guidelines in the Java Style Guide found on D2L
- □ Your code fulfills the functionality outlined above
- □ You have all the prompts listed in the examples in the exact same order
- □ You have blank lines as indicated in the examples
- □ Your strings (both prompts and output) are formatted **exactly** as shown in the examples