

WES-CS GROUP MEETING #7

Exercise 1: If and Switch Statements

Use the yellow and pink, or blue and green cards for this exercise. Each yellow/blue card has a code fragment that includes an *if* statement, and each pink/green card has a code fragment that includes a *switch* statement. Your job is to match each *if*-fragment with an equivalent *switch*-fragment. There will be some *switch*-fragments that have **no** matching *if*-fragment. For the *switch*-fragments that are similar but not equivalent to the *if*-fragments, explain what is wrong.

Exercise 2: Loops

Part (a)

Write code that prints each minute between 1:00pm and 2:59pm using one or more loops; i.e., your code should print:

```
1:00pm
1:01pm
...
1:59pm
2:00pm
2:01pm
2:59pm
```

Swap your code with your neighbor and see if either of you can find any errors. Once you think you both have correct code, you can test it by typing it in, compiling it, and seeing if it works.

Part (b)

Write code to print all of the 15-minute interval times between 11:00am and 2:45pm (i.e., 11:00am, 11:15am, ..., 2:00pm, 2:45pm). Note that noon is 12:00pm.

If you want to do another exercise, make the upper limit 2:15 instead of 2:45.

Exercise 3: PigLatin

The rules for translating an English word to Pig-Latin are as follows:

- If the word starts with a vowel, it is unchanged.
- Otherwise, all of the consonants at the beginning of the word (up to the first vowel) are moved to the end of the word, preceded by a dash (for readability), and followed by “ay”.

For example, the sentence “I love Madison in the springtime” would be translated to “I ove-lay adison-May in e-thay ingtime-spray”.

For this exercise, you will complete the *PigLatin* class started on the next page, which will allow users of the class to translate an English word to Pig-Latin.

First, make sure you understand the translation rules by translating the following phrases into Pig-Latin:

Hello world

I love WESCS

the car says ooga ooga

Part (a): Take a look at the incomplete *PigLatin* class on the next page. Make sure you understand what each method is supposed to do, and discuss the reasons for making some of the methods private and others public.

```

class PigLatin {
    private String englishWord;

    // constructor
    public PigLatin( String s ) {
        englishWord = s.toLowerCase();
    }

    // translate: translate the English word to Pig-Latin
    public String translate( ) {
        /* part (c) */
    }

    // firstVowelPos: return the position of the first vowel
    //                  in the English word; return -1 if there
    //                  is NO vowel in the word
    //
    private int firstVowelPos( ) {
        /* part (b) */
    }
}

```

Part (b): Write the *firstVowelPos* method, which returns the position of the first vowel in the *englishWord* field, or returns -1 if there is no vowel in the word. Here are some examples:

englishWord	Result of calling <i>firstVowelPos</i>
hello	1
ice	0
spring	3
qyzzx	-1

To test your code, write a *main* method for the *pigLatin* class that creates several *PigLatin* objects, calls their *firstVowelPos* methods and prints the results.

Part (c): Now describe in English how the *translate* method of the *PigLatin* class should work (using the rules for translating from English to Pig-Latin given above, and making use of the *firstVowelPos* method). Then write the actual code. Also write a *TestPigLatin* class with a *main* method that asks the user of the program to type in a word, reads the word, and prints the translation. Compile your code and run it!

Exercise 4: More Loops

Remember the *Artist* class we used for our first set of exercises? It provided the following methods:

```
void drawLineDown( int length )
```

Draw a vertical line of the given length (in inches), starting from the current position and going straight down. The current position is changed to be at the bottom end of the line.

```
void drawLineUp( int length )
```

Draw a vertical line of the given length, starting from the current position and going straight up. The current position is changed to be at the top end of the line.

```
void drawLineRight( int length )
```

Draw a horizontal line of the given length, starting from the current position and going straight to the right. The current position is changed to be at the right end of the line.

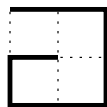
```
void drawLineLeft( int length )
```

Draw a horizontal line of the given length, starting from the current position and going straight to the left. The current position is changed to be at the left end of the line.

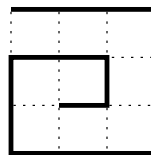
The pictures below show four *clockwise square spirals*, each of which starts at the upper-left corner of the square, and stops when drawing the next line would run into a line that's already been drawn (the dotted lines are just there to show you a grid of inches).



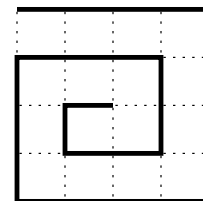
1-by-1 inch square



2-by-2 inch square



3-by-3 inch square



4-by-4 inch square

Write the code that would make an *Artist* object called *picasso* draw each of these spirals. Look for the pattern, then write a method with an *int* parameter *n* that uses a loop to make *picasso* draw a spiral in an *n*-by-*n* inch square. Test your code using the *Artist* class.

Yellow or Blue Cards

```
if (x == 0) y = 1;  
else if (x == 1) y = 2;  
else y = 3;
```

```
if ( x == 4 ) {  
    x += 4;  
} else if ( (x > 4) && (x <= 7) ) {  
    x += 7;  
}
```

```
if ( x == 1 || x == 3 ) {  
    System.out.println("Good Morning!");  
} else {  
    System.out.println("Good Afternoon!");  
}
```

```
if (x == 0 && y == 0) doSomething();  
else if (x == 0 && y == 1) doAnotherThing();  
else doSomethingElse();
```

```
if ( x > 2 && x < 4 ) x = 10;  
if ( x <= 0 || x > 4 ) x = 100;
```

Pink or Green Cards

```
switch (x) {  
    case 0: y = 1;  
           break;  
    case 1: y = 2;  
           break;  
    default: y = 3;  
}
```

```
switch (x) {  
    case 0: y = 1;  
    case 1: y = 2;  
    default: y = 3;  
}
```

```
switch ( x ) {  
    case 4 : x = x + 4; break;  
    case 5 :  
    case 6 :  
    case 7 : x = x + 7; break;  
}
```

Pink or Green Cards

```
switch(x) {  
    case 5:  
    case 6:  
    case 7: x = x + 7;  
           break;  
    default: x = x + 4;  
}
```

```
switch(x) {  
    case 1:  
    case 3: System.out.println("Good Morning!");  
           break;  
    default: System.out.println("Good Afternoon!");  
}
```

```
switch(x) {  
    case 1:  
    case 3: System.out.println("Good Morning!");  
}  
System.out.println("Good Afternoon!");
```

Pink or Green Cards

```
switch (x) {  
  case 0: switch (y) {  
    case 0: doSomething();  
           break;  
    case 1: doAnotherThing();  
           break;  
    default:doSomethingElse();  
  }  
  break;  
  default: doSomethingElse();  
}
```

```
switch (x) {  
  case 0: switch (y) {  
    case 0: doSomething();  
           break;  
    case 1: doAnotherThing();  
  }  
  break;  
  default: doSomethingElse();  
}
```

Pink or Green Cards

```
switch (x) {  
    case 0: if (y==0) doSomething();  
            else doAnotherThing();  
            break;  
    default: doSomethingElse();  
}
```

```
switch ( x ) {  
    case 1:  
    case 2:  
    case 4: break;  
    default : x = 100;  
}
```

```
switch ( x ) {  
    case 1:  
    case 2:  
    case 4: break;  
    case 3: x = 10; break;  
    default : x = 100;  
}
```

Pink or Green Cards

```
switch ( x ) {  
    case 3: x = 10; break;  
    default : x = 100;  
}
```