

# WES-CS GROUP MEETING #7

## Exercise 1: Loops

Remember that in Java there are three kinds of loops:

- **for** loops
- **while** loops, and
- **do-while** loops.

Here are examples of each kind of loop; they all print the numbers from 10 down to 1:

```
for (int x = 10; x > 0; x--) {  
    System.out.println("x: " + x);  
}
```

```
int y = 10;  
while (y > 0) {  
    System.out.println("y: " + y);  
    y--;  
}
```

```
int z = 10;  
do {  
    System.out.println("z: " + z);  
    z--;  
} while (z > 0);
```

**Part (a):** Use one or more *for*-loops to print each minute between 1:00pm and 11:59pm; i.e., your code should print:

```
1:00pm  
1:01pm  
...  
1:59pm  
2:00pm  
2:01pm  
11:58pm  
11:59pm
```

**Part (b):** Use one or more *while*-loops to print each minute between 4:30am and 8:30pm.

**Part (c):** Use one or more *do-while* loops to print all of the 15-minute interval times between 1:00pm and 10:45pm (i.e., 1:00pm, 1:15pm, 1:30pm, etc).

## Exercise 2: If and Switch Statements

Use the yellow and pink cards for this exercise. Each yellow card has a code fragment that includes an *if* statement, and each pink card has a code fragment that includes a *switch* statement. Your job is to match each *if*-fragment with one or more equivalent *switch*-fragments (some *switch* fragments have no matching *if*-fragment).

Be careful, some are tricky!

(For the cards, print pages 6–8 on yellow paper, print pages 9–17 on pink paper, and cut out the individual cards.)

### Exercise 3: Programming with Switch Statements

For this exercise, you will write a program that displays the favorite color, or the birthday or the number of siblings of someone in your group.

Here's an example of what will happen when you run the program.

- First, the program will display:

```
Whose information would you like to access?  
Enter A, B, C, or D to choose a person, or press Q to exit.  
A) Bob  
B) Jill  
C) Mike  
D) Ashley  
Enter your choice now:
```

- After you enter A, B, C, or D, the program will display:

```
What would you like to know about <name of person>?  
Please select one of the following:  
A) Favorite color  
B) Birthday  
C) Number of siblings  
Enter your choice now:
```

- After you enter A, B, C, or D, the program will display the desired information; for example, if you chose 'C' for Mike and 'A' for favorite color, the program will print Mike's favorite color.

- The program should continue to ask you to choose a person and then a piece of information, until you type Q.

**Part (a):** Start by defining a *Person* class to store one person's name, favorite color, birthday, and number of siblings (you might consider several different ways to represent the birthday).

Write a constructor that has four arguments (the four pieces of information about the person). Also write accessor methods that return the name, favorite color, birthday, and number of siblings.

Write a main method to test your *Person* class. Create one *Person* object for each person in your group, then print all of the information for each person.

**Part (b):** Now change the main method so that after creating one Person for each person in your group it prints the menus shown above (that allow the user to select one person and one piece of information) and the requested information. Make use of switch statements that switch on the letter typed in to decide which Person object to use, and which of its methods to call.

**Part (c):** Finally, add a loop so that as long as you don't type a Q the program keeps asking you for your choices.

## Exercise 4: Logical Thinking

Today's logical-thinking exercise is an old chess puzzle. The board (shown below) is a 3x3 part of the chessboard. The goal is to swap the red and blue knights, using a sequence of legal moves (of course, two knights may not occupy the same square at any time).

**B**: blue knight

**R**: red knight

<b>B</b>		<b>B</b>
<b>R</b>		<b>R</b>

If you can solve the puzzle, consider the following additional questions:

1. What is the minimum number of moves required to solve the puzzle?
2. Is it possible starting with the original configuration to arrange the knights as shown below?

<b>R</b>		<b>B</b>
<b>B</b>		<b>R</b>

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

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

```
if ( x == 4 ) {  
    x = x + 4;  
} else if ( x <= 7 ) {  
    x = 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 ( character != ' ' ) Action1();  
else Action4();
```

```
if ( x == 8 ) x = 0;  
else if ( x*2 == 8 ) x = 2;  
else if ( x - 1 == 13 ) x = 3;
```

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

```
if ( i > 0 && i < 4 ) x = 10;  
else if ( i <= 0 || i > 7 ) x = 40;  
else if ( i == 4 ) x = 20;  
else if ( i > 4 && i <= 7 ) x = 30;
```

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

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

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

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

```
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!");
```

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

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

```
switch( character ) {
    case ' ' :   Action4(); break;
    default  :   Action1();
}
```

```
switch( character ) {  
    case ' ' :    Action1(); break;  
    default  :    Action4();  
}
```

```
switch ( x ) {  
    case 8: x = 0; break;  
    case 4: x = 2; break;  
    case 14: x = 3; break;  
}
```

```
switch ( x ) {  
    case 8: x = 0;  
    case 4: x = 2;  
    case 14: x = 3;  
}
```

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

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

```
switch ( i ) {  
    case 1 :  
    case 2 :  
    case 3 : x = 10; break;  
    case 4 : x = 20; break;  
    case 5 :  
    case 6 :  
    case 7 : x = 30; break;  
    default : x = 40;  
}
```

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