Logic and loops

“if” conditional

- A conditional is a programming statement executed only if a specific condition is met.
- Java provides the "if" keyword to be used in an expression that results to either a boolean "true" or "false".
- Example:

```
if (args.length < 1) {
    System.out.println(“Not enough arguments”);
}
```

“if” conditional

- Optional "else if" and "else" keywords can be used.

  Example:

```
String server;
if (args.length < 1) {
    System.out.println(“Not enough arguments”);
} else if (args.length > 1) {
    System.out.println(“Too many arguments”);
} else {
    server = args[0];
}
```

“switch” conditional

- Used to test a variable against a number of values.

Using the “if” statement:

```
if (operation == ‘+’) {
  add(object1, object2);
} else if (operation == ‘-’) {
  subtract(object1, object2);
} else if (operation == ‘*’) {
  multiply(object1, object2);
} else if (operation == ‘/’) {
  divide(object1, object2);
}
```

Using the “switch” statement:

```
switch (operation) {
  case ‘+’:
    add(object1, object2);
    break;
  case ‘-’:
    subtract(object1, object2);
    break;
  case ‘*’:
    multiply(object1, object2);
    break;
  case ‘/’:
    divide(object1, object2);
    break;
}
```

“switch” conditional

- The test variable can be of types byte, char, short, int or String (since Java 7)

  Example:

```
switch (x) {
    case 2:
    case 4:
    case 6:
    case 8:
        System.out.println(“x is an even number”);
        break;
    default: System.out.println(“x is an odd number”);
}
```

Example: DayCounter

“for” loops

- Used to repeat a statement or a block of statements until a condition is met.
- Structure:

```
for (initialization; test; increment) {
    statement;
}
```

  - **Initialization**: expression that initializes the so called loop index at the start.
  - **Test**: a boolean expression that is evaluated at each pass of the loop and must result to "true" in order for the loop to execute.
  - **Increment**: An expression or function call that changes the value of the loop index at each pass of the loop.

“for” loops

Example:

```
for (int i = 0; i < 10; i++) {
    System.out.println(i);
}
```

Example:

```
for (i = 4001; notPrime(i); i += 2) {
    System.out.println(i);
}
```

Example: HalfLooper

“while” loop

- Repeats a block of statements for as long as particular condition stays "true".

  Example:

```
while (i < 13) {
    x = x * i++; // the body of the loop
}
```

Example: ArrayCopier

“do-while” loop

- Much the same like the while loop with the exception that the body (block of statements) is executed at least once.

  Example:

```
long i = 1;
do {
    i *= 2;
    System.out.print(i + “ “);
} while (i < 3000000000000L);
```

Breaking out of loops

- The break keyword can be used to break out of loops early before the entire loop completes.

  Example:

```
int count = 0;
while (count < array1.length) {
    if (array1[count] == 1) {
        break;
    }
    array2[count] = (float) array2[count++];
}
```

The “continue” keyword

- The **continue** keyword starts the loop over at the next iteration.

  Example:

```
int count = 0;
int count2 = 0;
while (count++ <= array1.length) {
    if (array1[count] == 1) {
        continue;
    }
    array2[count2++] = (float) array1[count];
}
```

Labeled loops

- An optional label tells Java where to resume execution of a program.

  Example:

```
out: for (int i = 0; i <10; i++) {
    while (x < 50) {
        if (i * x++ > 400) }
            break out;
        }
        // inner loop here
    }
    // outer loop here
}
```

The “conditional” (“ternary”) operator

- A short alternative to the "if-else" that is an expression i.e. returns a value.

  Example:

```
int ourBestScore = myScore > yourScore ? myScore : yourScore;
```

  which is the same as

```
int ourBestScore;
if (myScore > yourScore) {
    ourBestScore = myScore;
} else {
    ourBestScore = yourScore;
}
```

Exercises

Exercise: YearlyCalendar

- Using the countDays() method from the DayCounter application, create an application that stores every date in a given year from January 1 to December 31 as Strings in an array and prints it out.