COMP10062: Week 2 Guide

Sam Scott, Mohawk College, 2021

0. Reading for this Week

For this week, you should **read sections 3.1 to 3.4 and 4.1 to 4.3** (skip “The Conditional Operator” and “The Exit Method” in 3.1, “Enumerations” in 3.3, “Dialog Box” in 3.4, “Using a Comma” in 4.1 and “The For-Each Statement” in 4.1, “Assertion Checks” in 4.2).

This is probably the biggest chunk of reading you will be asked to do in one week, but keep in mind that you already know how to use loops and if statements from Programming Fundamentals. These sections are just teaching you what you already know, but in Java. You can probably skim over a lot of it.

1. Basic Flow of Control

a. The If Statement (pp. 141-165)

In Python

```
if x > 5:
    x = x / 2
    print(x)
elif x < 5:
    x = x * 2
    print (x)
else:
    print("error")
print("done")
```

In Java

```
if (x > 5) {
    x = x / 2;
    System.out.println(x);
} else if (x < 5) {
    X = x * 2;
    System.out.println(x);
} else
    System.out.println("error");
System.out.println("done");
```

Java Notes

Boolean expressions in `(…)`
Indenting is for readability

No `elif`.

You can use a code block (with `{…}`) or a single statement after an if or else.

Advice: ALWAYS USE { … }.

b. Boolean Expressions (pp. 148-153)

In Python

```
x = 5
good = False
if (x > 5 and x < 10) or
(x > 15 and x < 20):
    good = True
    print("in range")

if not good:
    x = 8


if x == "5":
    print("x is a string")
```

In Java

```
int x = 5;
boolean good = false;
if ((x > 5 && x < 10) ||
(x > 15 && x < 20)) {
    good = true;
    System.out.println("in range");
}
if (!good) {
    x = 8;
}

if (x == "5")
    System.out.println("x is a
string");
```

Java Notes

|| = OR
&& = AND
() around boolean expressions

! = NOT

- syntax error! Operands must be compatible (i.e. castable)

c. Comparing Strings – a Java “Gotcha” (pp. 153-158)

In Python

```
name = input()
if name == "Sam":
    print("Welcome!")
else:
    print("Go away!")
```

In Java

```
Scanner sc = new Scanner(System.in);
String name = sc.nextLine();
if (name.equals("Sam"))
    System.out.println("Welcome!");
else
    System.out.println("Go away!");
```

Notes

For Strings, using == will usually result in `false`, even if the contents are the same. Use `s1.equals(s2)`.

(See **Extra, Week 2 - Comparing Strings** and sections 2.2 and 3.2 of the text for more.)

d. The While Loop (pp. 203-206, section 4.2)

In Python

```
count = 10
while count >= 1:
    print(count)
    count -= 1

print("blast off")
```

In Java

```
int count = 10;
while (count >= 1) {
    System.out.println(count);
    count -= 1;
}
System.out.println("blast off");
```

"ITACL”

Initialize loop variable
Test loop variable
Act
Change loop variable
Loop back to the top

More Notes

You can also use the **`break`** statement to exit a loop (see p. 238)

e. Variable Scope

In Python

```
count = 10
while count >= 1:
    show = count * 2
    print(show)
    count -= 1

print("blast off")
print(show)
```

In Java

```
int count = 10;
while (count >= 1) {
    int show = count * 2;
    System.out.println(show);
    count -= 1;
}
System.out.println("blast off");
System.out.println(show);
```

Notes

`show` is a local variable

Syntax error in Java

More Notes

Python has **function scope:** Variables introduced inside functions are local to that function
Java has **block-level scope:** All variables are local to the block in which they are declared

- The block might be a method declaration block
- Or it might be part of a while loop or if statement
- Nested blocks can access the variable, but code outside of the variable’s block cannot.

Quick Exercise

Paste the Java code from the example above into a main method and verify that it contains a syntax error. Then fix the code so that the final print statement will work (don’t change the print statement, change something else).

2. The For Loop (pp. 219-226, section 4.2)

While Loop

```
int x = 1;
while (x <= 10) {
    System.out.println(x);
    x++;
}
System.out.println("blast off");
```

For Loop

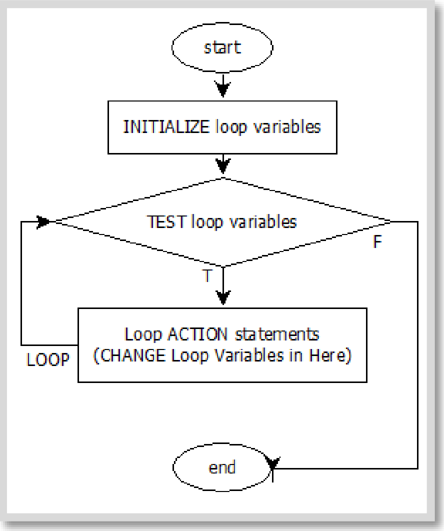
```

for(int x = 1; x <= 10; x++) {
   System.out.println(x);
}

System.out.println("blast off");
```

Notes

The for loop and while loop are almost equivalent.

The only difference above is that the scope of the `x` variable is restricted: Adding `System.out.println(x)` as the last line of the For Loop example above would cause a syntax error.

What is ++? Java contains two operators `++` and `--`. They are add or subtract 1 from a variable. They are equivalent to the Python `+=1` and `-=1` respectively (both of which are also allowed in Java).



While Loop

```
Initialize
while (Test) {
    Act
    Change
}
```

For Loop

```

for(Initialize; Test; Change) {
   Act
}
```

Notes

The `for` loop is an "ITACL" loop, just like the `while` loop. It is preferred by some programmer because it puts all the loop logic (Initialize, Test and Change) into a header instead of spreading it out in multiple places.

3. The Switch Statement (pp. 176-182)

If ... else chain

```
if (choice == 1) {
   // do choice 1;

} else if (choice == 2)
{
   // do choice 2;

} else if (choice == 3)
{
   // do choice 3;

} else {
   // do default;
}
```

Switch statement

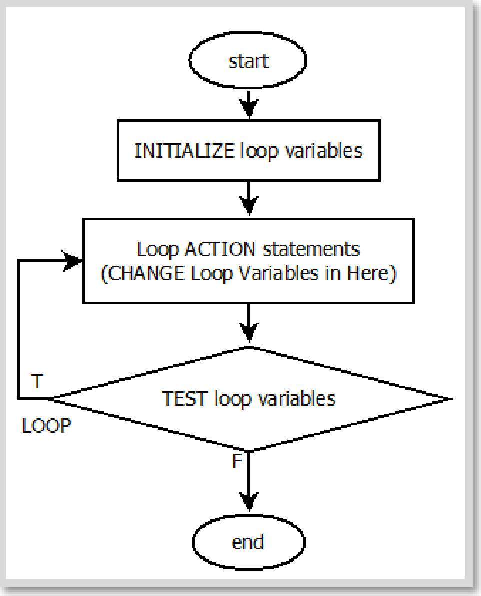
```
switch (choice) {
   case 1:
      // do choice 1
      break;
   case 2:
      // do choice 2
      break;
   case 3:
      // do choice 3
      break;
   default:
      // do default
}
```

Notes

Replaces an if ... else chain, but only if it is using simple == expressions and only for primitive types. From Java 8 onwards, switch will also work for Strings.

The break statement is not optional. Leaving it out will cause multiple cases to execute.

4. The Do...While Loop (pp. 206-211, section 4.2)



It's "IACTL" Instead of "ITACL"

```
Initialize
do {
   Act
   Change
} (Test);
```

Example: Sentinel Controlled Loop

```
Scanner sc = new Scanner(System.in);
int input;
do {
   n = sc.nextInt();
   System.out.println(n*2);
} while (n > 0);
```

When to Use this Loop...

When you know you will always go through at least one iteration. For example, when collecting user input.