Introduction To Java Programming: Part 2

You will learn how to implement the following structures in Java: branching, looping, random numbers, composites (arrays)

James Tam

Logical Operators

Logical Operation	Python	Java
AND	`and`	`&&`
OR	`or`	`||`
NOT	`not`	`!`

James Tam

Decision Making In Java

- Python decision making structures:
  - `if`
  - `if, else`
  - `if, elif, else`
- Java decision making structures:
  - `if`
  - `if, else`
  - `if, else if, else`
  - *`switch`*

James Tam

Java Relational Operators And Branching

```
if (operand     relational operator     operand)
```

Java operator	Mathematical equivalent	Meaning	Example
<	<	Less than	5 < 3
>	>	Greater than	5 > 3
==	=	Equal to	5 == 3
<=	≤	Less than or equal to	5 <= 5
>=	≥	Greater than or equal to	5 >= 4
!=	≠	Not equal to	x != 5

James Tam

Decision Making: If

Format:

```
if(Boolean Expression)
    Body
```

Example:

```
if(x != y)
    System.out.println("X and Y are not equal");

if ((x > 0) && (y > 0))
{
   System.out.println("X and Y are positive");
}
```

Also note: unlike Python the entire Boolean expression in Java must be enclosed in brackets

- Indenting the body of the branch is an important stylistic requirement of Java but unlike Python it is not enforced by the syntax of the language.
- **What defines the body** is either:
  1. A semi colon (single statement branch).
  2. Braces (a body that consists of single or multiple statements).

James Tam

The 'Body'

- Single statement body

```
if (num > 0)
    System.out.println("Part of body");
    System.out.println("Not part of body");
System.out.println("Still not part of body");
```

- Multi statement (compound) body

```
if (num > 0)
{
    System.out.println("Part of body");
    System.out.println("Part of body");
}
```

James Tam

Checking For Equality: Strings

- Never use the equality operator when comparing objects (e.g., Strings)
  - More on this later
- Instead the `equals()` and `equalsIgnoreCase()` methods should be used instead.
  - These methods return a true or false value
- Quick example:

```
String s1 = "abc";
String s2 = "abc";
if (s1.equals(s2))
    System.out.println("equal");
if (s1.equals(s2) == true)
    System.out.println("equal");
```

James Tam

Decision Making: `If, Else`

Format:

```
if(Boolean expression)
    Body of if
else
   Body of else
```

Example:

```
if (x < 0)
    System.out.println("X is negative");
else
    System.out.println("X is non-negative");
```

James Tam

If, Else-If (Java)

Format:

```
if (Boolean expression)
    Body of if
else if (Boolean expression)
   Body of first else-if
    ...
else if (Boolean expression)
   Body of last else-if
else
    Body of else
```

James Tam

If, Else-If (2)

Name of the complete example: IfElseIfExample.java

```
if (letter == 'A')
    System.out.println("4.0");
else if (letter == 'B')
    System.out.println("3.0");
else if (letter == 'C')
    System.out.println("2.0");
else if (letter == 'D')
    System.out.println("1.0");
else if (letter == 'F')
    System.out.println("0.0");
else
    System.out.println("Letter grade must be one of A, B, "
      +"C, D, F");
```

James Tam

Alternative To Multiple `Else-Ifs`: `Switch`

- Use when checking for equality of: integer numbers or characters.

James Tam

Alternative To Multiple Else-If Structure: Switch

Format (character-based switch):

```
switch (character variable name)
{
     case '<character value>':
          Body
          break;

     case '<character value>':
          Body
          break;
        :
    default:
          Body
}
```

Important! The break is mandatory to separate Boolean expressions (must be used in all but the last scenario).

The break transfers execution out of the switch structure, otherwise cases will 'fall-through' in some versions of Java.

1 The type of variable in the brackets can be a byte, char, short, int or long

James Tam

Alternative To Multiple Else-If Structure: Switch (2)

Format (integer based switch):

```
switch (integer variable name)
{
     case <integer value>:
          Body
          break;

     case <integer value>:
          Body
          break;
        :
    default:
          Body
}
```

1 The type of variable in the brackets can be a byte, char, short, int or long

James Tam

Alternative To Multiple Else-If Structure: Switch (3)

- **Name of the complete example**: SwitchExample.java

```
switch (letter)
{
    case 'A':
    case 'a':
        gpa = 4;
        break;

    case 'B':
    case 'b':
        gpa = 3;
        break;

    case 'C':
    case 'c':
        gpa = 2;
        break;
```

James Tam

Alternative To Multiple Else-If Structure: Switch (4)

```
        case 'D':
        case 'd':
           gpa = 1;
           break;

        case 'F':
        case 'f':
           gpa = 0;
           break;

        default:
           gpa = -1;
    }
```

James Tam

The Switch Using An Integer For The Cases

- **Name of the complete example**: SwitchExample2.java

```
switch (gpa)
{
    case 4:
        letter = 'A';
        break;

    case 3:
        letter = 'B';
        break;

    case 2:
        letter = 'C';
        break;
```

James Tam

The Switch Using An Integer For The Cases

```
        case 1:
            letter = 'D';
            break;

        case 0:
            letter = 'F';
            break;

        default:
           letter = 'E';
    }
```

James Tam

Switch: Benefit (Cleaner Code)

- Benefit (when to use):
  - It may produce simpler code than using an if, else-if (e.g., if there are multiple compound conditions)
  - Contrast

```
If ((menu == 'a') ||
    (menu == 'A') ||
    (menu == 'N') ||
    (menu == 'n'))
    System.out.println("New player");
else if ((menu == 'q') ||
         (menu == 'Q'))
```

```
switch(menu)
{
  case 'a':
  case 'A':
  case 'N':
  case 'n':
    System.out.println("New player");
    break;

  case 'Q':
  case 'q':
```

James Tam

Looping Java

- Python looping structures:
  - `for`
  - `while`
- Java looping structures:
  - `for`
  - `while`
  - *`Do-while`*

James Tam

Looping Examples

- **Name of the file containing complete**: `LoopingExamples.java`
- The example contains examples of all 4 loops: `while`, `for` and two versions of the `do-while` loop.

James Tam

While Loops

Format:

```
while (Boolean expression)
{
    Body
}
```

Example (L: Java, R: Python):

```
i = 1;
while (i <= 5)
{
    System.out.print(i + " ");
    i = i + 1;
}
```

```
i = 1
while (i <= 5):
    print(i, end = "")
    i = i + 1
```

James Tam

For Loops

Format:

```
for (initialization; Boolean expression; update control)
{
    Body
}
```

Example (L: Java, R: Python):

```
for (i = 1; i < 6; i++)
{
    System.out.print(i + " ");
}
```

```
for i in range (1, 6, 1):
    print(i, end = "")
```

James Tam

For Loops: Java Vs. Python

- Unlike Python with most languages for loops are generally used for counting (e.g., up down).
- Iterating through other series (such as lines in a file) is not possible.
- Python example not possible in other languages

```
inputFile = open("input.txt", "r")
for line in inputFile:
    print(line)
```

- In Java however the **loop control update** can be any mathematical expression (even randomly assigned) [1].

```
for (i = 1; i <= 100; i = i * 5)
```

1 The update can be any valid mathematical expression (e.g. addition, subtraction, multiplication, division etc.) or a method return value that is numeric.

James Tam

For Loops: Java Vs. Python (2)

- Also note that with the Java `for`-loop that the **stopping boundary** can be made explicit.

```
for (i = 1; i <= 10; i++)
-Vs.
for i in range (1, 11, 1):
```

James Tam

Post-Test Loop

- Pre-test loops: `For` and `while` loops check the stopping condition **before** executing the body.
  - That is, the loop body executes zero or more times.
  - Example that never executes:

```
i = 10;
while (i <= 5)
{
    System.out.println(i);
    i = i + 1;
}
```

- Post test loops (not directly implemented in Python): check the stopping condition **after** executing the body.
  - That is, the loop body executes one or more times.
  - (Of course the behavior of post test loop can be simulated with a pre-test loop such as a while loop).

James Tam

Do-While Loop: Structure

- **Format**:

```
do
    body
while (Boolean Expression);
```

- **Example**:

```
i = i + 1;
do
{
    System.out.print(i + " ");
    i = i + 1;
} while (i <= 5);
```

James Tam

Do-While Loop Examples

```
System.out.println("\nFirst Do-While loop");
i = 1;
do
{
    System.out.print(i + " ");
    i = i + 1;
} while (i <= 5)


 System.out.println("\nSecond Do-While loop");
 i = 888;
 do
 {
     System.out.print(i + " ");
     i = i + 1;
 } while (i <= 5);
```

James Tam

Common Mistake: Branches/Loops

- Forgetting how a body is defined:
  - using braces,
  - OR
  - with single statement bodies the end of the body is marked with a semi-colon.
- (Partial) examples:

```
while (i < 10)
     System.out.println(i);
     i = i + 1;
```

Body (logic error)

```
while (i < 10);
{
     System.out.println(i);
     i = i + 1;
}
```

Body

James Tam

Many Pre-Created Classes Have Been Created

- **Rule of thumb of real life**: Before writing new program code to implement the features of your program you should check to see if a class has already been written with the features that you need.
- Note: **for many assignments in this class** you may have to implement all features yourself rather than use pre-written code.
  - **You may receive little or no credit otherwise**.
  - (Remember you are in this course to learn and develop Java and Object-Oriented programming skills and one learns by writing code not just using code).
- The Java API is Sun Microsystems's collection of pre-built Java classes:
  - http://java.sun.com/javase/8/docs/api/

James Tam

Example: Generating Random Numbers (Probabilities)

- Name of the (more complete example): `DiceExample.java`

```
import java.util.Random;
public class DiceExample
{
    public static void main(String [] args)
    {
        final int SIDES = 6;
        Random generator = new Random();
        int result = -1;
        result = generator.nextInt(SIDES) + 1;
        System.out.println("1d6: " + result);
        result = generator.nextInt(SIDES) + 1;
        result = result + generator.nextInt(SIDES) + 1;
        result = result + generator.nextInt(SIDES) + 1;
        System.out.println("3d6: " + result);
    }
}
```

James Tam

Arrays

- They are similar to Python lists.
  - Specified with square brackets
  - Indexed from 0 to (number elements-1)
- Some differences between Java arrays and Python lists:
  - All elements must be of the same type e.g., array of integers cannot mix and match with floats
  - Python has methods associated with lists although an array in Java has a '`length`' attribute associated with it.
  - Arrays cannot be dynamically resized (new array must be created).

James Tam

Creating An Array

- **Format**:
  - <*type*> $[]_1$ <*name*> = new <*type*> [<*Number of elements*>];
- **Example (common approach)**:

```
final int MAX = 100;
int [] grades = new int[MAX];
```

- **Example (Fixed size array declared and initialized – rarely used approach)**:

```
int [] array = {1,2,3};
```

1 Each dimension must be specified by a set of square brackets e.g., two dimensional array requires two sets of brackets

James Tam

Arrays: Complete Example

- Name of the (more complete example):

GradesExampleArray.java

```
public class GradesExampleArray
{
    public static void main(String [] args)
    {
        final int MAX = 10;
        int [] grades = new int[MAX];
        int i = 0;
        Random generator = new Random();
```

James Tam

Arrays: Complete Example (2)

```
        for (i = 0; i < MAX; i++)
        {
            grades[i] = generator.nextInt(101);
        }

        for (i = 0; i < grades.length; i++)
        {
            System.out.println("Element #" + i + " grade " +
              grades[i]);
        }
    }
}
```

Unlike Python lists you cannot pass an entire Java array to a display method such as `print` or `println` in order to display the elements:

```
System.out.println(grades);
```

```
[I@c351f6d
```

James Tam

After This Section You Should Now Know

- The structure and syntax of decision making and looping structures.
- How to generate random numbers.
- How to create and work with Java arrays containing simple types (e.g. boolean, int, float, char etc.).

James Tam

Copyright Notification

- "Unless otherwise indicated, all images in this presentation are used with permission from Microsoft."

James Tam