Outline

1) Java Basics

2) Our First Program

3) Java Code Structure

4) Control Flow

5) Examples

6) Arrays

A bit about Java

1) Java is a statically typed language which is compiled into bytecode and then runs on the Java Virtual Machine (JVM). Python on the other hand is interpreted at runtime.

2) Descendant of C++, a lot of syntax and data type overlap/reuse from intro CS classes.

3) Considered the de-facto Object oriented language.

4) Built primarily for large scale programming.

Our First Java Program

Greeting.java requirements -

1) Read input from the user.
2) All variables must be created in advance along with the datatype.
3) We use System.out.print(String) for printing
4) If statement

Let's create and run a java program called "Greeting.java" that takes in a user's name and age, and then prints out if they are an "Adult" or "Child" (above or below 18).

Python Vs Java

> Let’s you do more with less code

> No talk about types

> No help preventing bugs

> Can be used to build big projects but is primarily used for scripting.

> Do more with lots of code

> Must always declare types

> Will fight you over some bugs which python will ignore.

> Used to build MASSIVE projects!

Java Code Structure

1) Package Declaration (For organizing files in large codebases)
2) Import statements (For external libraries/classes)
3) Class Declaration (Java is object oriented so everything is inside a class)
4) Main method (Entry point for execution)
5) Methods & variables (Here is where we implement the program logic)

Let’s check these attributes out in a simple program (codestructure.java)

Printing in java

Python printing -

```
print (“Hello World”)
```

> Straightforward and to the point

Java printing -

```
System.out.println(“Hello World”);
```

> lots of words just to print “Hello World”

> There is a reason for this!

System Class

1) System is a built-in class that provides access to system-related resources.

2) System.out is a variable in the System class.
   Where do you want to send the data? out, error, file?
   We need to print to someplace!

3) System.out.println prints the input string to a new line.

4) System.out.print skips the newline.

Reading inputs

Python -

```
user_input = input("Enter a number");
```

> Always returns a string

> You need to convert it to a desire type.

Java -

```
Scanner sc =  new Scanner(System.in);
System.out.println( "Enter a number");
Int a = sc.nextInt();
```

> using the nextInt() method we only accepted an int input.

> Scanner class has methods like nextLine() to input strings, nextDouble() to input doubles etc.

Using Random Numbers

1) We generate random numbers in Java using an object similar to inputs.

2) java.util.Random (class Random in the java.util package)

3) Import the Random Class, create a Random variable, create a Random object, and use the Random object to generate the random variable.

4) It generates a random number between 0 (Inclusive) and 100 (Exclusive).

Let’s work on this in a file RandomNumbers.java

Variables in Java

```
  // Variable declaration -- initial value is optional
  <modifiers> <type> <name>;
 // Example
double myAge = 31.98886579387;
```

1) Declare any time before use
2) Strongly - typed
3) If you don't pick a default value, each type has one.
4) Two possible variables : Objects or Primitive types

Java Primitive Types

1) boolean (false)
2) char (unicode, not ascii)
3) int (0)
4) double (0)
5) Use short, byte for smaller ints, and float for smaller doubles.
6) Type names are often lowercase and variable names are often camelCase.

Java Objects

String(text)
 > Only object that gets special syntax
 > immutable
 > represents string text material

We also created a Scanner object in the past slides.

1) Variables store location, but not the object itself.
2) variable defaults are null (meaning “ not pointing to an object”)
3) Type names are always Capitalized as seen in Scanner

Math Operations

Operator	Python	Java
+, *, -	Same	Same
/	floating-point result	Depends on type
//	integer division	N/A
<, >, <=, >=	Same	Same

Mathematical Operations

Equality: ==, !=

> Behave as expected for primitive types
> Behave differently for objects (checks for memory locations)

Example:

```
String a = “Hello World”;
String b = “Hello World”;;
System.out.println (a == b) // false
```

Control Flow

Similarly to python, we have if, while and for loops

if

```
 if (boolean expression) {

} else if (boolean expression) {

} else {

}
```

1) Boolean expression must evaluate to true or false
2) Curly braces are optional but a good practice
3) Can have as many else if as needed
4) Only one else

for

```
for(<init>; <condition>; <update>) {

}
```

1) check condition and enter loop
2) after loop runs update the variable
3) run till condition is met

While

```
while ( boolean expression) {

}
```

1) check condition, if true enter loop
2) at end of loop, re-evaluate condition

Examples

1) Print each number from 1 to 20 inclusive.
2) Print powers of 3 upto 6561.
3) Trick Question

In file ControlFlow_examples.java

Examples

```
int i;
for (i = 0; i < 5; i++) {
    for (int i = 0; i < 3; i++) {
        System.out.println("Hello world! " + i);
    }
}
```

What is the output to this nested for loop?

Java Object Syntax

1) The variable is not it’s value.
2) Variables reference objects.
3) Use the “new” keyword to create a new object.
4) Java takes care of delete.
5) Similar to pointers in C.

String Syntax

1) String literals are declared in double quotes unlike python.
   Example: String s = “Hello World”;

2) + can concatenate string.

3) + can be used across types, it always results in a new string.
   Example: “Hello World” + 9 -> “Hello World9”

4) Strings are immutable. Use .charAt(index) to get a specific character and .length() to get length of a string.

String Example

```
String s = “Hello World”;
char c = s.charAt(6); // ‘W’
s = s + “Hi Again!” // “Hello WorldHiAgain!”
s = s + 9; // “Hello WorldHiAgain!9”
```

Arrays

1) Arrays are like Lists in Python but have a fixed size and have a single type.
2) Arrays always behave like objects.
3) <Type> array = Type [];
4) int variable => int[] x
5) Uses default values for primitive data types and null for objects.
6) Another Way => int[] x = {values};

Arrays

Fun exercise to challenge our For Loop and Array indexing knowledge:

```
public class Main {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30, 40, 50};

        // Find length indirectly and traverse in reverse order
        for (int i = (arr.length * arr.length) % 10 + 4; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
}
```

Accessing an Array

```
// Syntax:
myArray[index]
```

1) Index can be computed.

2) no negative indexing like python.

3) Java throws an out-of-bounds exception.

Array Example

Let’s try a challenge on different ways to create arrays in Java.

Remember :- In Java, arrays are Objects!

- In file ArrayExamples.java

Array Properties

1) Array equality checks identity
2) Array printing is not the best idea, let’s look at this!
3) Use loops to print
4) Use loops to check equality, or use Array.equals(arr 1, arr 2)

Array Properties

```
public static boolean isArrayEqual(int[] left, int[])
	if (left.length != right.length) {
		return false;
	} else {
	for (int i = 0; i < left.length; i++) {
		if (left[i] != right[i]) {
			return false;
		}
	}
	return true;
	}
```

Functions and Arrays

Functions can modify and return arrays:

```
public static char[] replaceChars (char [] arr) {
    for (int i =0; i < arr.length; i++) {
        if (Character.isVowel(arr[i]) {
            arr[i] = ‘z’;
        }
    }
    return arr;
}
```

Let’s test our Java Knowledge

Let us create a java program with the following functionality:

1) Generate a random array of 10 characters using Math.random(), picking random ASCII characters from (‘!’ to ‘~’, ASCII values 33 to 126).
2) Count the number of vowels.
3) Count the number of uppercase characters.
4) Find the smallest ASCII value character in the array.
5) Print all these values.

Let’s code this out the AsciiChallenge.java

Feedback Form

This is my first time teaching, I would greatly appreciate feedback from everyone!

Scan this QR code or visit the following link : https://forms.gle/Wxcev7Rip1tUNxji6

End!

https://forms.gle/Wxcev7Rip1tUNxji6

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline

Outline