

1st lesson

Before Starting

- Record this meeting
- No previous experience teaching, please bear with me
- I will try to cover the points in <https://amazontechnicalacademy.com/employee/technical-assessment>
- I do not have plans to go over the prerequisite courses but I can try to answer doubts about them. (How to add provided libraries)
- Current aim is questions (x min) + quick theory (30-x min) + exercises (30 min), though I do not know how much the theory will take, maybe less. I will ask for feedback and we will prob shift the distribution of the time.
- You can send me questions offline preferably to pampero+ata@amazon.com.

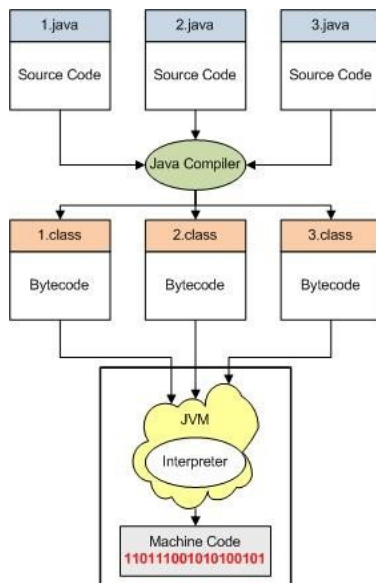
Key Points:

- Installing IDE and "Hello World!"
- Variables and types
- conditions (if-else, and, or)

People with programming experience will probably not learn anything new with this. You can either try <https://www.hackerrank.com/interview/preparation-kits> (choose any of the preparation kits. Once inside a problem,

select Java in the upper right deployable list), work on the prerequisite course or anything else. If you have any doubts I will try to answer them after finishing the explanation

Introduction to programming

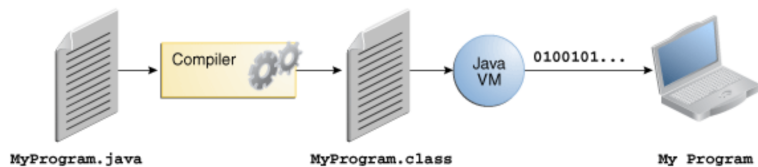


A programming language is to give instructions to a computer. Computers will do what we tell them to do, but they do not understand our normal language. We have to learn the language they speak to give them orders.

Main parts of a computer relevant to a programmer would be CPU and memory. Programming languages offer the possibility of abstracting from those so that writing programs is easier.

Programming languages offer different levels of abstraction. e.g.

- In C, C++ we have to manually manage the memory being allocated for data in several cases. Programmer is in charge on allocating and freeing memory.
- Java manages memory automatically. It has a garbage collector that is in charge of freeing memory from unused variables.



Java

[https://en.wikipedia.org/wiki/Java_\(programming_language\)](https://en.wikipedia.org/wiki/Java_(programming_language))

Java is a high-level, class-based, object-oriented programming language. Intended to let [programmers](#) write once, run anywhere meaning that compiled Java code can run on all platforms that support Java without the need to recompile. (thanks to JVM?).

Released in May 1995. Several versions released until today (and still new ones are being developed). Most common one until recently has been version 8.

Basic Syntax

Comments

```
// one line comment  
  
/** Block comment  
    It can have several  
    lines  
*/
```

90% of all code comments:



Bad: https://github.com/pablopenna/flask_app/blob/master/src/beebotte_rnd.py

Good:

```
val Q3_2021 = Cycle.builder().year(2021).quarter(3).build()!!

@Suppress("UNUSED_PARAMETER") // Backed out change requiring context, but this logic v
fun WorkItem.getNumberOfNeededAssessors(context: ActivityContext? = null, featureFlags

    val targetJobLevel = getTargetJobLevelOrThrow()
    return when {
        targetJobLevel > 7 -> 2
        targetJobLevel == 7 -> if (featureFlags[FeatureFlag.DEFAULT_L7_TPAS_TO_ONE_ASS
        else -> 1
    }
}

fun getMinNumberOfRequiredAssessorsForLevel(level: Int, featureFlags: FeatureFlags): I
    return when {
        level <= 6 -> 1
        level == 7 -> if (featureFlags[FeatureFlag.DEFAULT_L7_TPAS_TO_ONE_ASSESSOR, fa
        else -> 2
    }
}
```

```

fun getMaxNumberOfRequiredAssessorsForLevel(level: Int): Int {
    return when {
        level <= 6 -> 1
        else -> 2
    }
}

fun WorkItem.getNeededAssessorTypes(context: ActivityContext? = null, featureFlags: Fe
    return when (type) {
        WorkItemType.TECH_PROMO_ASSESSMENT ->
            if (getNumberOfNeededAssessors(context, featureFlags) == 1) {
                listOf(AssessorReviewType.FIRST_ASSESSOR)
            } else {
                listOf(AssessorReviewType.FIRST_ASSESSOR, AssessorReviewType.SECOND_AS
            }
        WorkItemType.COMPLETED_WORK_REVIEW -> listOf(AssessorReviewType.CWR_REVIEWER)
        WorkItemType.PLANNED_WORK_REVIEW -> listOf(AssessorReviewType.PWR_REVIEWER)
        WorkItemType.technical_skills_review -> listOf(AssessorReviewType.TSR_REVIEWER)
    }
}

```

Variables

Used to store basic data in the program.

By convention they always start in lower case.

You can use camelCase or snake_case to separate words within a variable.

Good practices for naming a variable include giving it a meaningful name and not using abbreviations.

declaration:

```
int counter = 5;
```

After having a variable declared, you can just use it by using its name (in this case `counter`)

You can declare a variable without assigning a value. e.g.

```
int counter;
```

If it is used before assigning a value, the compiler will throw an error.

TYPES

The simple types are

https://www.w3schools.com/java/java_data_types.asp

byte	Stores whole numbers from -128 to 127
short	Stores whole numbers from -32,768 to 32,767
*int	Stores whole numbers from -2,147,483,648 to 2,147,483,647
long	Stores whole numbers from -9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
*float	Stores fractional numbers. Sufficient for storing 6 to 7 decimal digits
double	Stores fractional numbers. Sufficient for storing 15 decimal digits
*boolean	Stores true or false values
char	Stores a single character/letter or ASCII values
String	Stores several characters/letter

Wrapper classes (I should leave it for a future class)

Primitive Data Type	Wrapper Class
char	Character
byte	Byte
short	Short
long	Integer
float	Float
double	Double
boolean	Boolean

Operators

https://www.w3schools.com/java/java_operators.asp

Arithmetic:

- + Addition Adds together two values $x + y$
- Subtraction Subtracts one value from another $x - y$
- * Multiplication Multiplies two values $x * y$
- / Division Divides one value by another x / y
- % Modulus Returns the division remainder $x \% y$ $3\%3 = 0, 4\%3 = 1, 8\%3 = 2$
- ++ Increment Increases the value of a variable by 1 ++x Equivalent to $x = x + 1$. Usually used within loops.
- Decrement Decreases the value of a variable by 1 --x Equivalent to $x = x - 1$. Usually used within loops.

Condition:

- | | | |
|----|--------------------------|----------|
| == | Equal to | $x == y$ |
| != | Not equal | $x != y$ |
| > | Greater than | $x > y$ |
| < | Less than | $x < y$ |
| >= | Greater than or equal to | $x >= y$ |
| <= | Less than or equal to | $x <= y$ |

Logical: concatenating conditions

- | | | | |
|----|-------------|---|---|
| && | Logical and | Returns true if both statements are true | <code>name=="Pablo" && surname=="Pena"</code> |
| | Logical or | Returns true if one of the statements is true | <code>x < 5 x < 4</code> |
| ! | Logical not | Reverse the result, returns false if the result is true | <code>!isValid</code> |

Clauses using condition and/or logical resolve to either a boolean value of true or false

Short-circuiting

AND: reads from left to right. If one of the conditions is false, the whole AND evaluates as false and no further conditions are checked.

e.g.

```
public boolean isBookAvailableForAndroid(String bookId) {  
    return isAndroidAllowed && checkAvailabilityInDatabaseForPlatform(ANDROIDID, bookId);  
  
    // vs  
  
    return checkAvailabilityInDatabaseForPlatform(ANDROIDID, bookId) && isAndroidAllowed;  
}
```

OR: returns true as soon as one of the conditions is true

```
public boolean isUserAllowedTo(Operation operation, User user) {  
    return user.isAdmin() || operation.getAllowedUsers().include(user)  
}
```

If-else

They can be used to control the flow of the program.

The if-else clause: if the condition in the if is met (condition equals boolean true), the code block in it will execute. If not, the else code block will do.

```
int index = 2;  
if (index > 0) {  
    System.out.println("Reading started");  
} else {  
    System.out.println("Pending start")  
}
```


The output of executing this code would be “Reading started”. On the other hand

```
int index = -1;
if (index > 0) {
    System.out.println("Reading started");
} else {
    System.out.println("Pending start")
}
```

Would output “Pending start”

A more “complex” one

```
if (index > 0 || (isStarted && !errorOccurred)) {
    System.out.println("Reading started");
} else {
    System.out.println("Pending start")
}
```

if vs if-else

What will the output be for 5, 6 and -1?

```
if(index % 5 == 0) {
    System.out.println("A")
} else {
    System.out.println("B")
}
```

index

```
if(index % 5 == 0) {
    System.out.println("A")
} else if(index > 0){
    System.out.println("B")
}
```

index

```
if(index % 5 == 0) {
    System.out.println("A")
}
if(index > 0)
    System.out.println("B")
}
```

Switch statement

<https://docs.oracle.com/javase/tutorial/java/nutsandbolts/switch.html>

Usually used instead of an if-else when there are a lot of paths depending on the value of a single variable.

```
int month = 8;
String monthString;
switch (month) {
    case 1: monthString = "January";
            break;
    case 2: monthString = "February";
            break;
    case 3: monthString = "March";
            break;
    case 4: monthString = "April";
            break;
    case 5: monthString = "May";
            break;
    case 6: monthString = "June";
            break;
    case 7: monthString = "July";
            break;
    case 8: monthString = "August";
            break;
    case 9: monthString = "September";
            break;
    case 10: monthString = "October";
            break;
    case 11: monthString = "November";
            break;
    case 12: monthString = "December";
            break;
    default: monthString = "Invalid month";
            break;
}
System.out.println(monthString);
```

IntelliJ

Integrated Development Environment (IDE) for developing in Java. There are lots of them: BlueJ, eclipse, netbeans, VSCode, ...

I will be using IntelliJ but you can use whichever you want. They all provide the same basic functionality that we will be using.

<https://www.jetbrains.com/idea/download>

<https://www.jetbrains.com/help/idea/creating-and-running-your-first-java-application.html>

Exercises:

- Hello World
- Ask for your name and Print your name
- <https://www.w3resource.com/c-programming-exercises/conditional-statement/index.php>