Java JDK & JRE

- To achieve the idea of “compile the code once, running everywhere”, a Java Runtime Environment (JRE) is required in each system. JRE is required for any Java code and application

- A JDK is a Java Development Kit, targeting for developer. It not only contains a JRE, but also contains some common library, source code and tools for development and debugging the program. JDK contains a JRE.

IDE

- IDE stands for Integrated development environment. This usually refers to software that helps developer to do the development easier and faster.
- A good IDE should:
  - Check and correct the target language
  - Auto pop the content
  - Easy to do the searching within the project
  - Integrated with popular frameworks

Git and Github

- Git is a protocol for managing code version control. A team of developer should use some form of code version control to work together and be able to integrate each others code fairly easily.

- Github is a web platform that build for open source projects. Developers can post their project to share with the community and the world. They can also reference and benefit from others open source project.

- For this AP course, we use Github as our lab coding and assignment repository

In case you need a reset up

JDK: Java SE Development Kit 8u181, click accept license, and download the windows x64 version: https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

IDE (the community version): https://www.jetbrains.com/idea/download/#section=windows

Text Editor
VS CODE: https://code.visualstudio.com/

Version Control
GitDeskTop: https://desktop.github.com/

Some intelIJ tricks

- Double tap SHIFT: for finding a file
- CTRL + SHIFT + F: find a keyword within project
- To be continued

Java Loop

Update til today’s content

For loop

One complete java statement

1

One conditional statement

+

One complete java statement
Or a java operation
Run after the loop operation

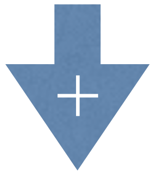
+

```
for ( [initial control variable declare]; [looping condition check] ; [condition change]) {

       / / do something
}
```

While loop

One conditional statement



```
while ( [changeable condition]) {

        / / keep doing something
}
```

Do while loop

```
do {

        // keep doing something

} while ( [changeable condition]);
```

One conditional statement

You can take the initiative and break the loop, but you need to know what you are doing

```
int myNumber = 1024;

while (myNumber > 0) {
        myNumber--;
        if (myNumber == 512) {
            break;
        }
}
System.out.println(myNumber);
```

Prints: 512

To be continued