

Introduction to Computers and Java

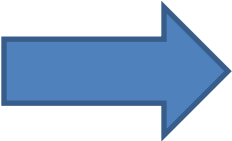




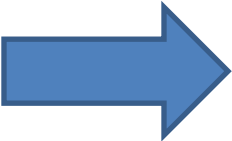




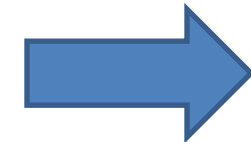
INPUT
(Data)



INPUT
(Data)

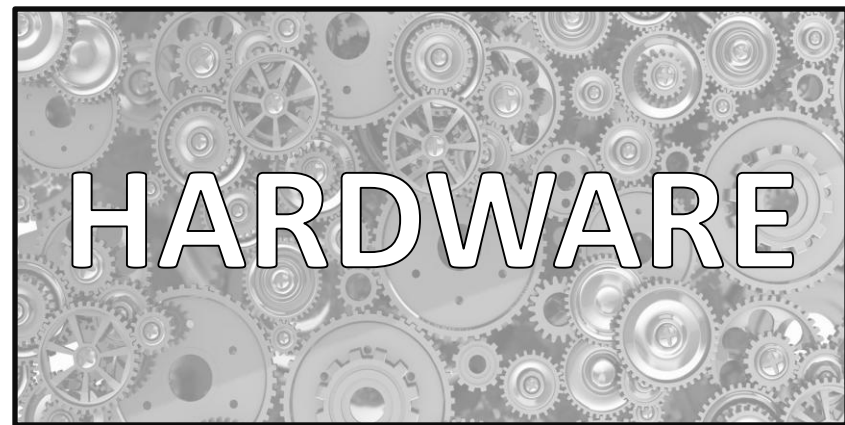


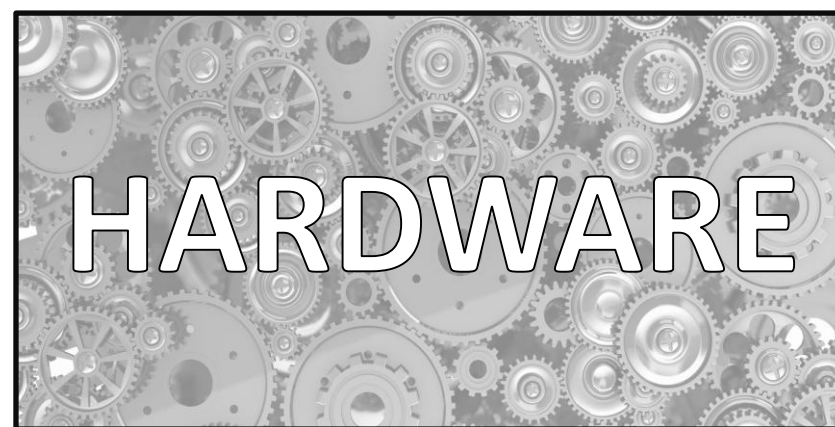
INPUT
(Data)



Output
(Information)







HARDWARE

CPU



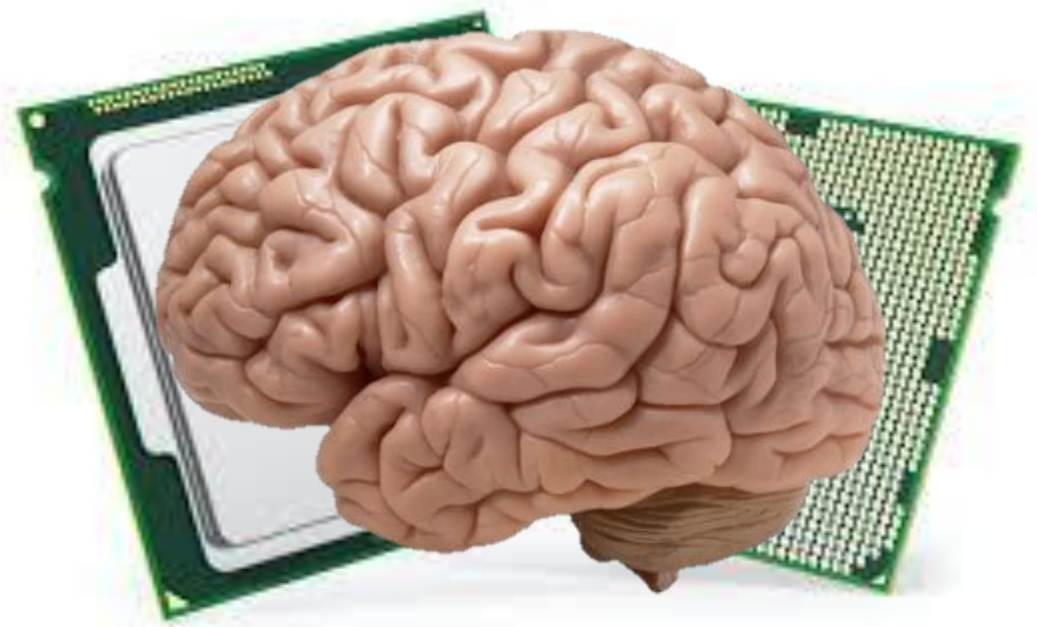
Memory



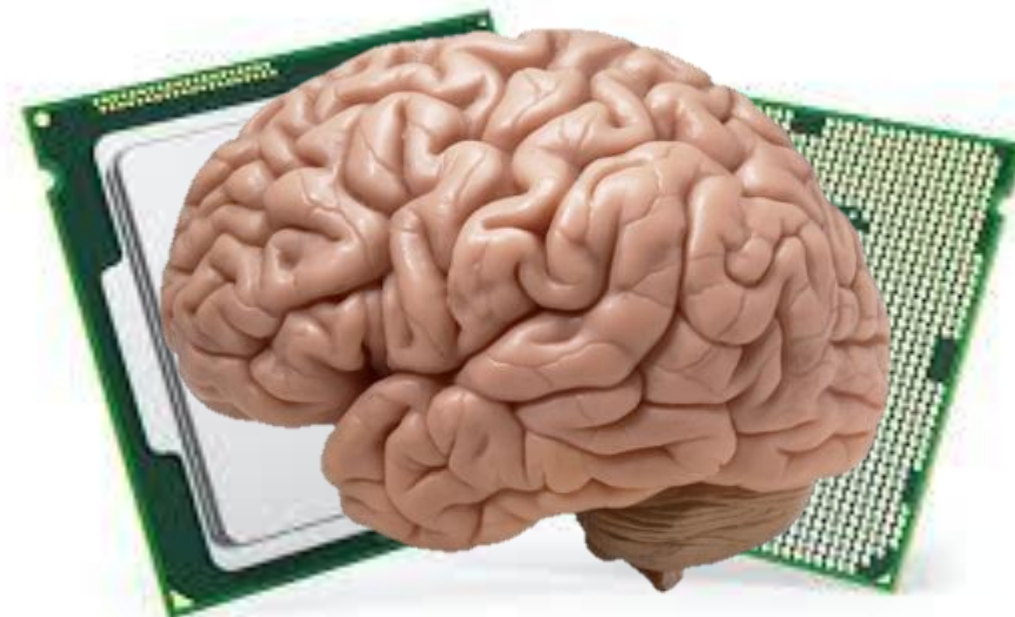
HARDWARE



HARDWARE

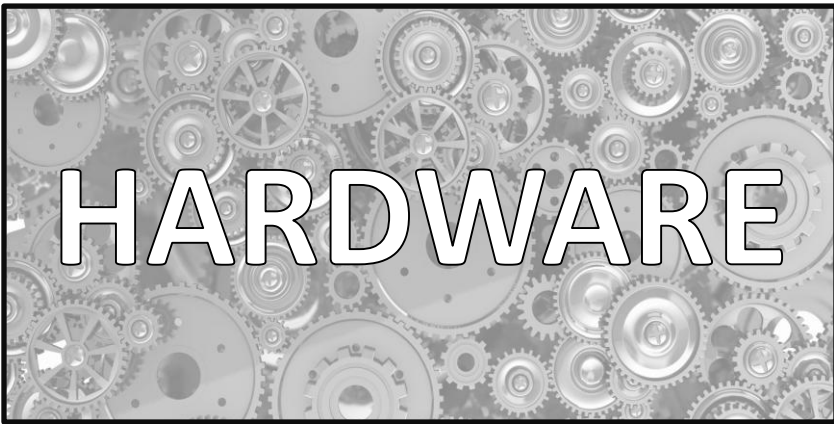


HARDWARE



HARDWARE





RAM

HARDWARE

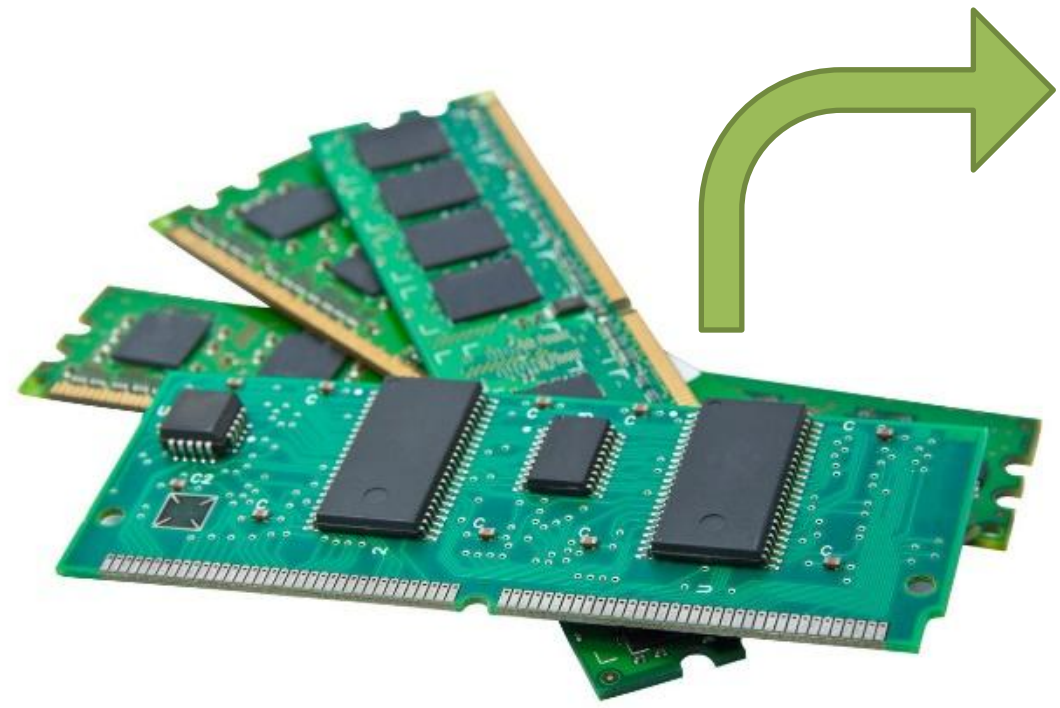


Secondary

HARDWARE



HARDWARE



Memory	
Addresses	Values
...	
256	01000001
260	01000010
264	01000010
268	01000001
...	

Bits and Bytes

- Bit – a digit with a value of 0 or 1
- Byte – Consists of 8 bits
- Address – The numbered location where each byte resides
- All data is encoded as a 0 or 1
 - Everything is a number
- When more than 1 bytes is needed then several adjacent addresses are used

Memory	
Addresses	Values
...	
256	01000001
260	01000010
264	01000010
268	01000001
...	

Files and Folders

- File – large group of bytes stored in secondary (auxiliary) memory
 - Files have names
 - Most files have extensions
- Folder (Directory) – group together multiple files
- Java programs are stored in files
 - Source code have the extension “.java”
 - Byte-Code have the extension “.class”

Memory	
Addresses	Values
...	
256	01000001
260	01000010
264	01000010
268	01000001
...	

Running Software

Secondary



Main



CPU



Running Software

Secondary



Load →

Main



CPU



Running Software

Secondary

Main

CPU



Load



Run Code



Running Software

Secondary

Main

CPU



Load



Run Code

Store Info



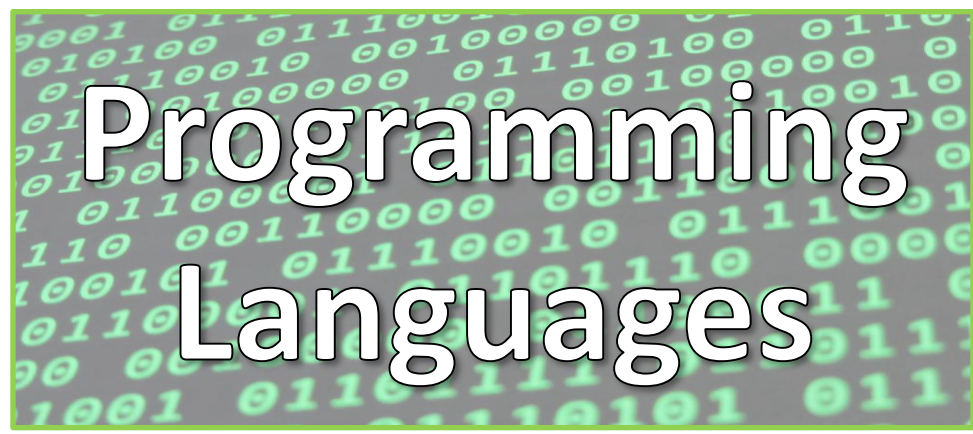




Algorithm ('algə_rɪt_həm) noun
A set of instructions to solve a problem

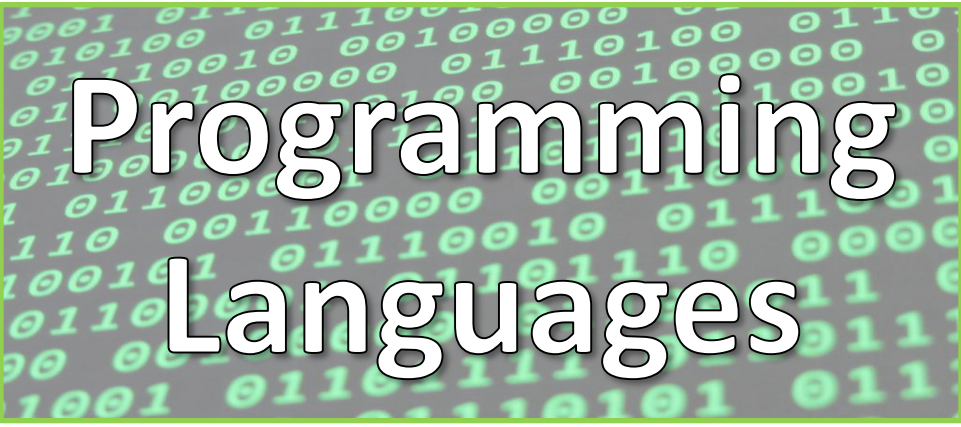


Program (prō, gram) *noun*
A set of instructions for a computer to follow.



Programming Languages





LOW LEVEL

Programming Languages

LOW

LEVEL

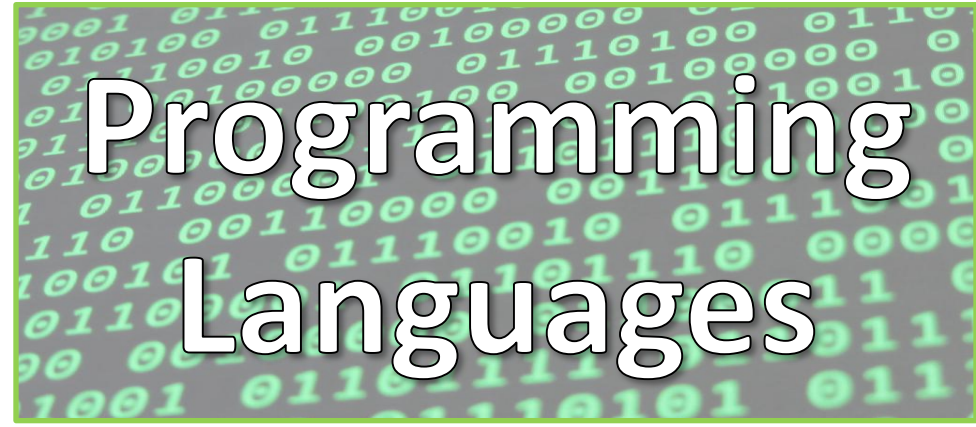
Machine Code

```

*****
* FUNCTION: INCH - Input character
* INPUT: none
* OUTPUT: char in acc A
* DESTROYS: acc A
* CALLS: none
* DESCRIPTION: Gets 1 character from terminal

```

C010	B6	80	04	INCH	LDA A	ACIA	GET STATUS
C013	47				ASR A		SHIFT RDRF FLAG INTO CARRY
C014	24	FA			BCC	INCH	RECIEVE NOT READY
C016	B6	80	05		LDA A	ACIA+1	GET CHAR
C019	84	7F			AND A	#\$7F	MASK PARITY
C01B	7E	C0	79		JMP	OUTCH	ECHO & RTS



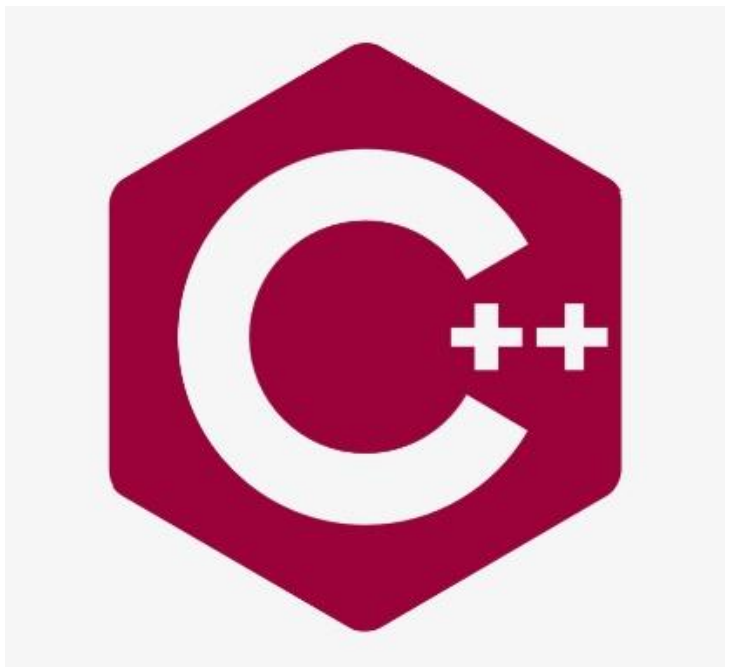
High Level

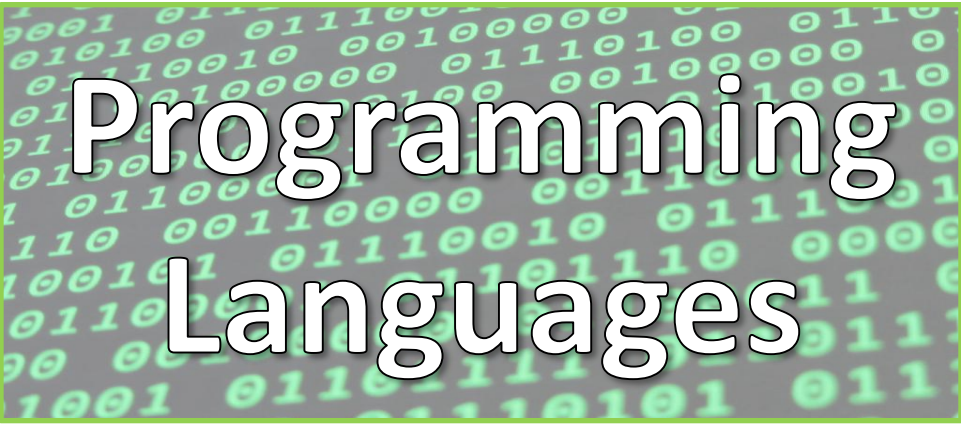


Programming Languages

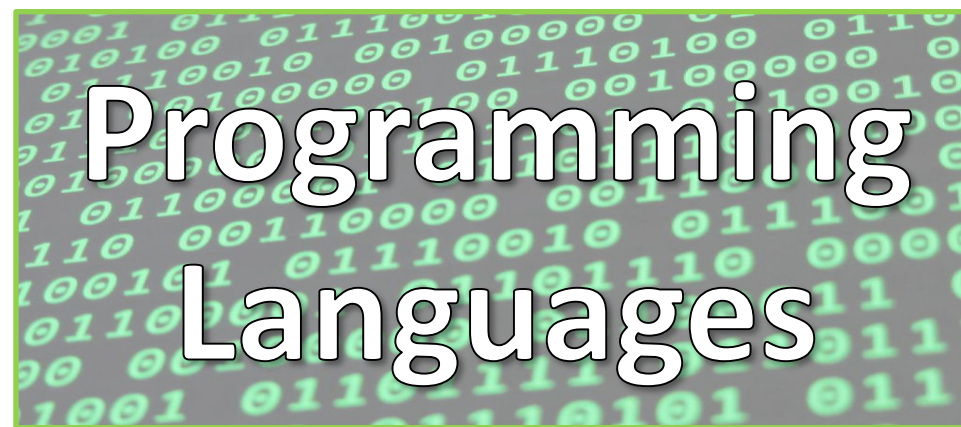


High Level



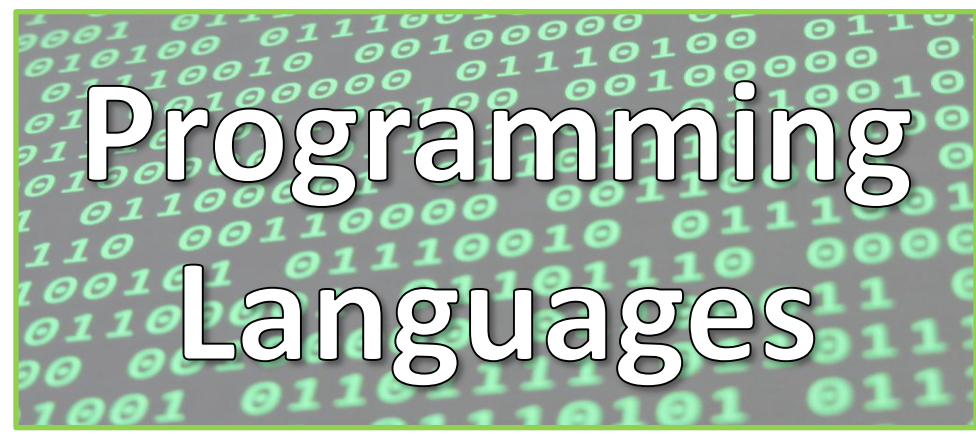


High Level



High Level

Nouns and Verbs



High Level

Syntax

Programming Languages



Programming Languages



Compiler



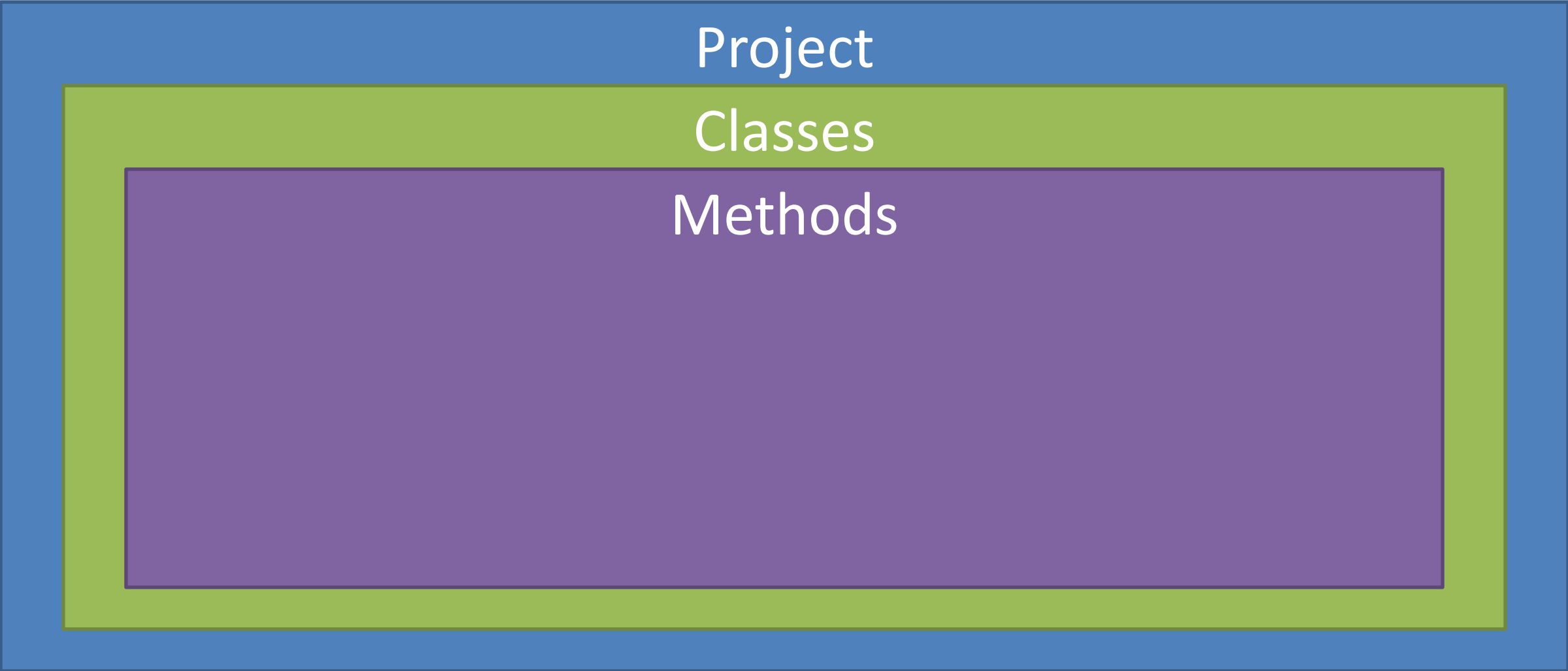




Project



Project
Classes





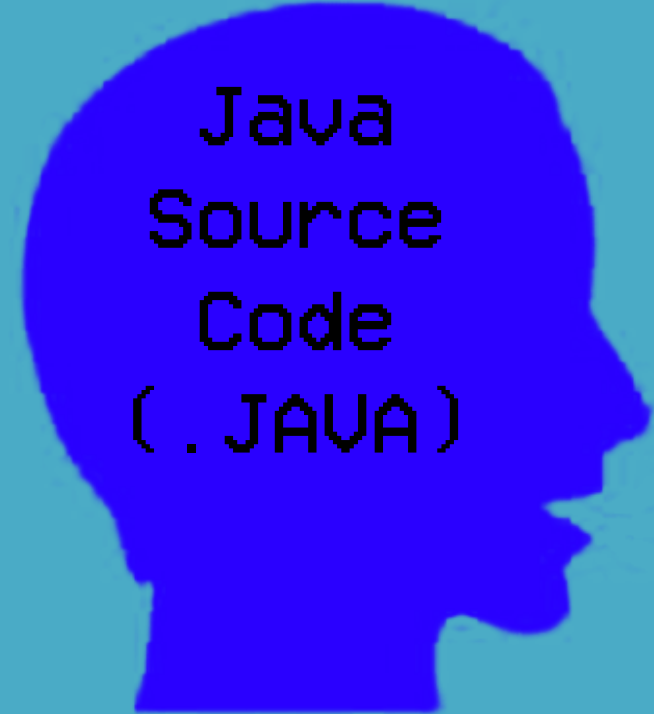
Classes

Methods

- Source Code in files with “.JAVA” extension
- The filename must MATCH the name of the class
- Everything is an “Object”



Compilation

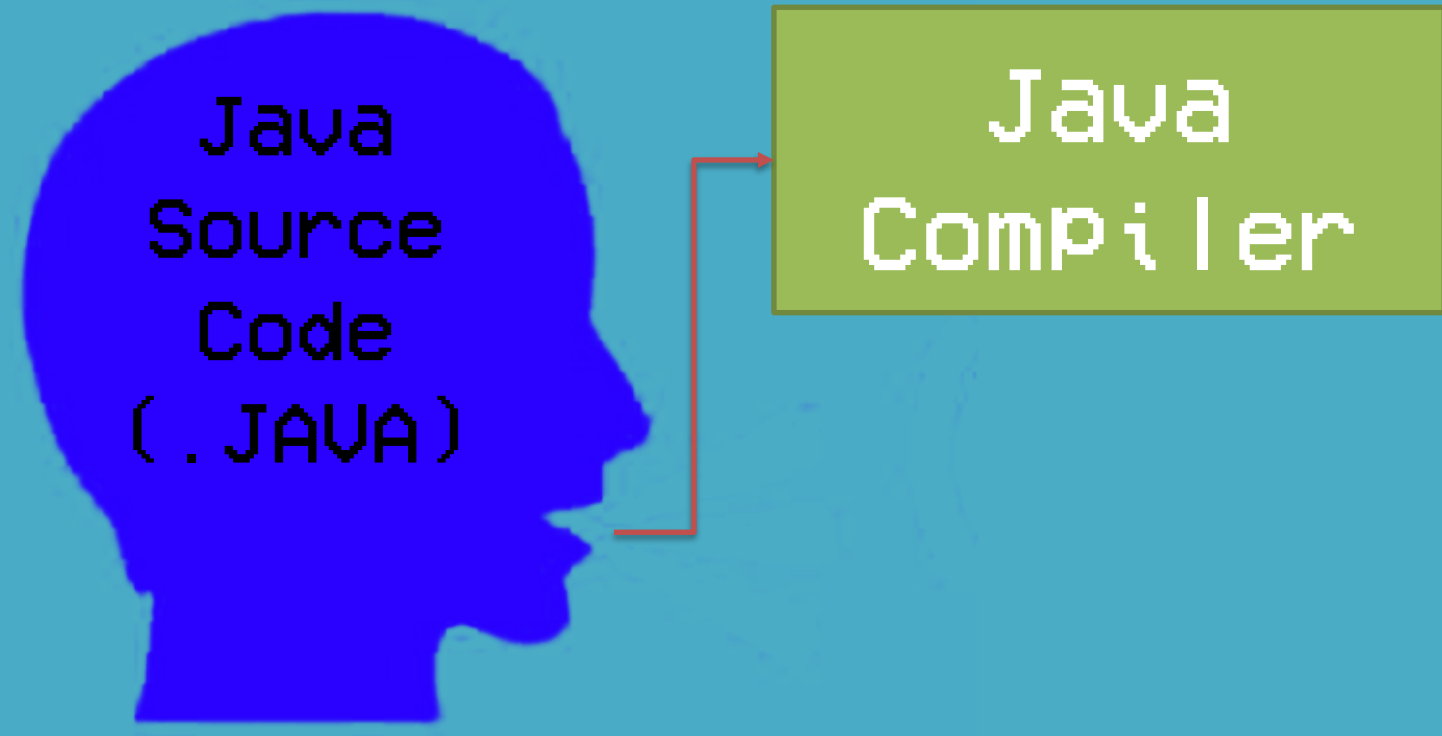


Running





Compilation

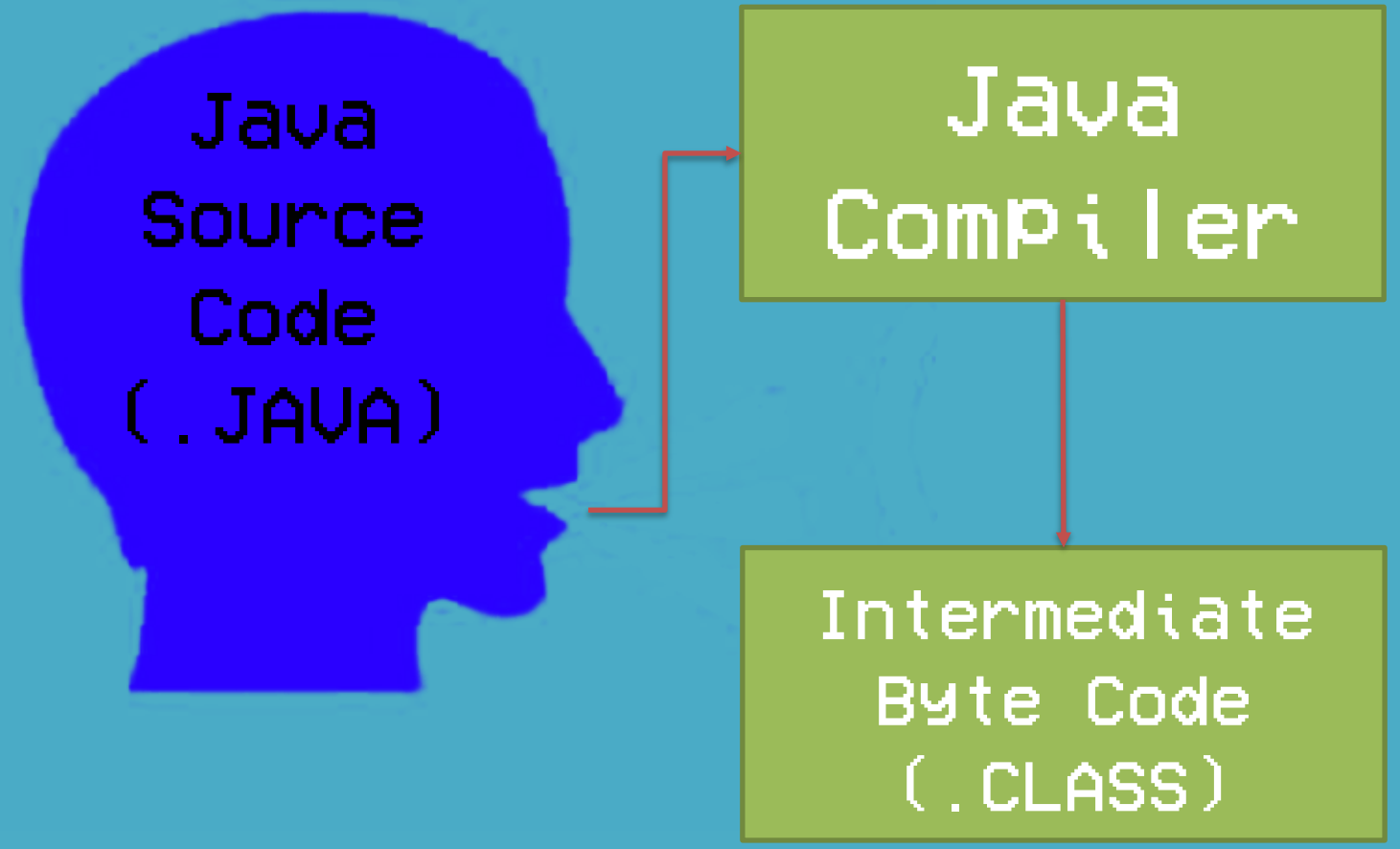


Running





Compilation

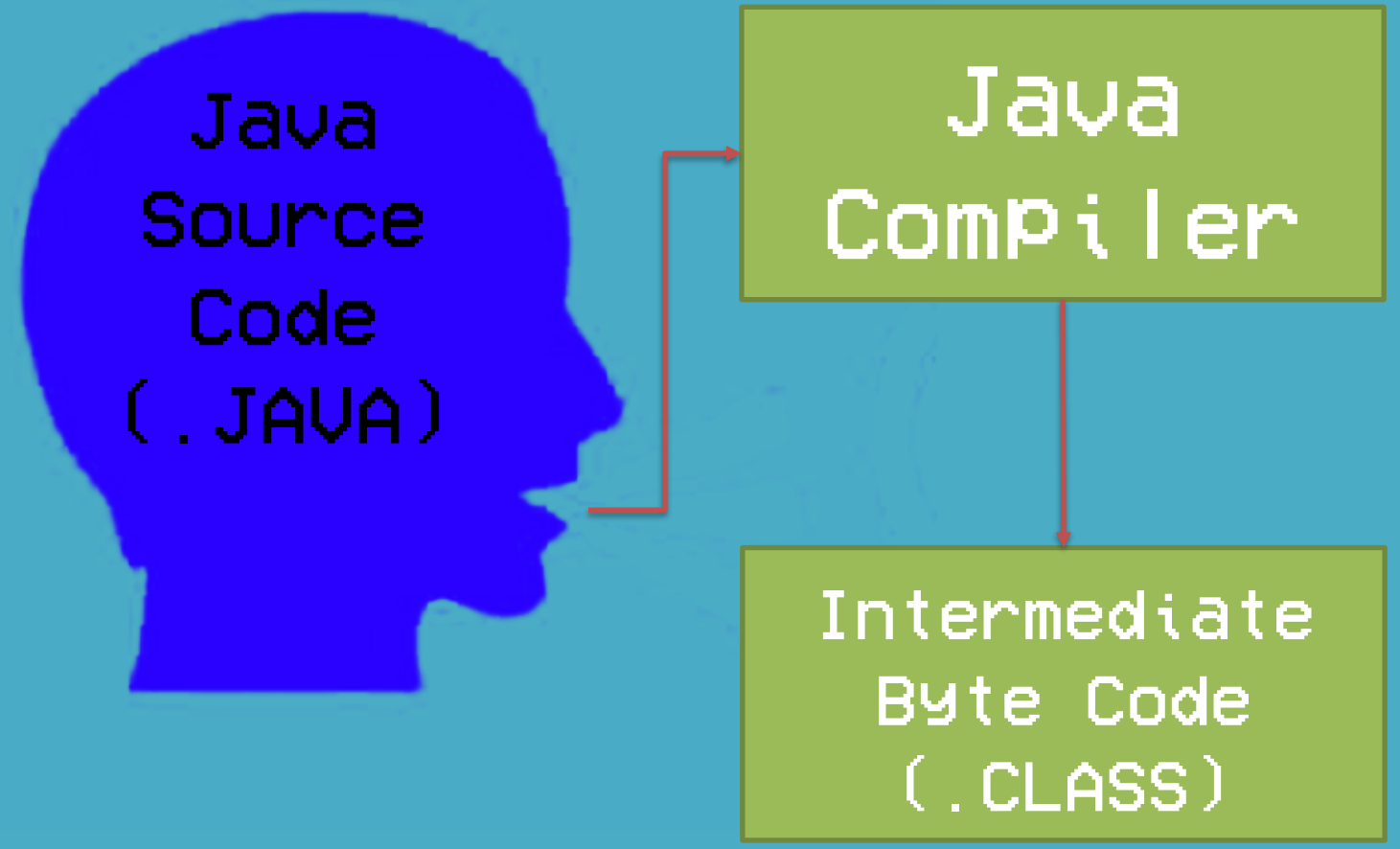


Running





Compilation



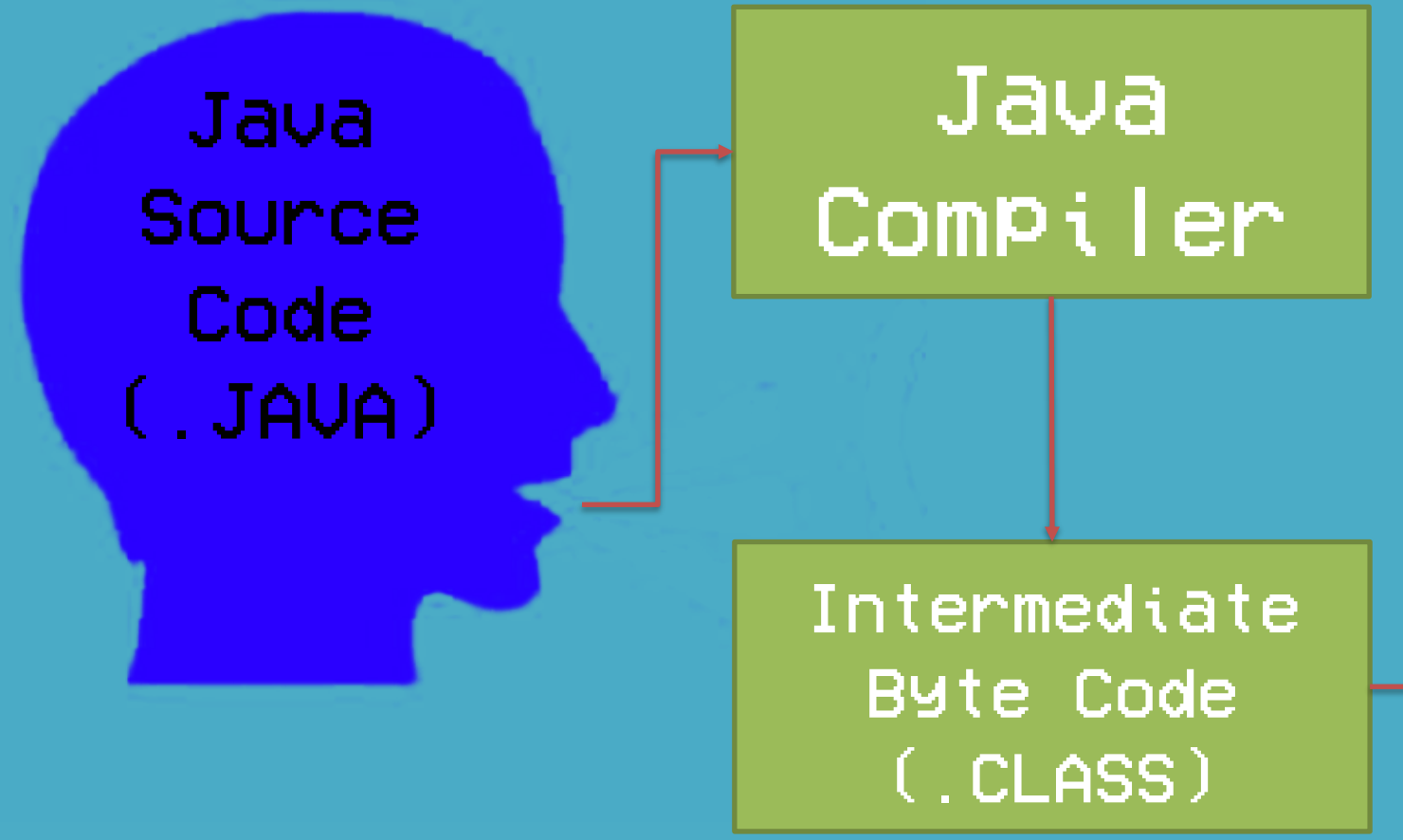
Running

Java Virtual Machine (JVM)

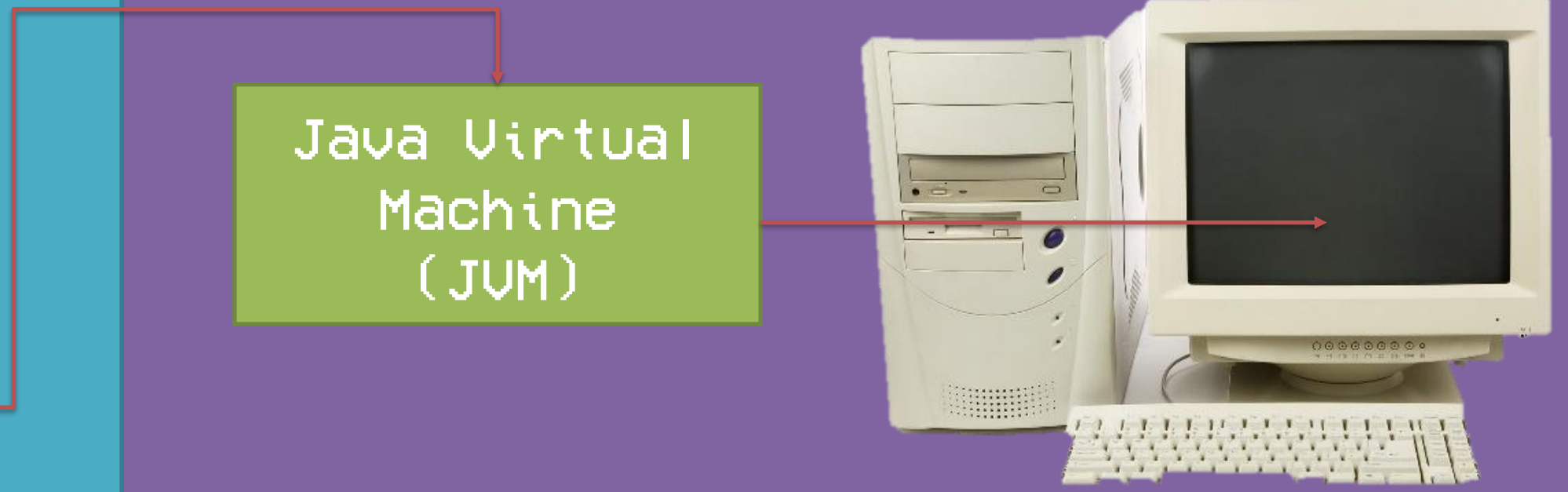




Compilation



Running



Other Terminology

- Running (Executing) – Is when the computer is following the instructions in a program
- Statement – An instruction to the computer. Most end with a semicolon “;”
 - int i;
 - double j;
- Syntax – The grammar rules for a programming language
- Comments – Code ignored by the compiler that is generally used to explain the code.
 - Single line comments use the “//”
 - Multiline comments use “/* */”
- Arguments – Information found inside of parenthesis “()” that provide information for methods or other statements
 - if(<<ARGUMENT>>)
 - System.out.println(<<ARGUMENT>>)
- Bug – an error in a program
- Debugging – the process of removing errors
- There are 3 major classes of errors
 - Syntax
 - Runtime
 - Logic

Types of Errors

- Syntax Errors– Grammatical mistakes in a program
 - Very common at first
 - These prevent a program from compiling and running
 - Common
 - Missing a Semicolon at the end of a statement
 - Using the wrong, misspelled, repeated, or incorrectly capitalized identifier
 - Mismatched curly braces “{}”, parenthesis “()”, single quotes “'””, double quotes “””, etc.

```
int i //Missing a semicolon
double j = 0.0;
J = 1.0; //Wrong identifier
System.out.println(i; //Missing parenthesis
```

Types of Errors

- Runtime Errors – Errors detected when the program is running but not during compilation
 - The code will compile but crashes at some point
 - When this happens the computer detects the error and terminates the program
 - Common
 - Divide by 0
 - Calling a method from a NULL object
 - An index going outside the bounds of an Array

```
double j = 1.0 / 0.0; // Divide by 0
Scanner keyboard; // This has not been constructed so it is NULL
keyboard.hasNextLine(); // Calling method from NULL object
int[] a = {5,4,3,2,1}; // An array
a[5] = 2; // Index 5 is out of bounds
```

Types of Errors

- Logic Errors – despite the program compiling and running, it produces incorrect results
 - Arguably the hardest to fix
 - Common:
 - Order of operations error
 - Round off mistakes
 - Incorrectly using types or methods

```
double f = 72.0; // 72 degrees fahrenheit
double c = f - 32.0 * 5.0/9.0; // Order of operations error
double c2 = (f-32.0)*(5/9); // 5/9 = 0 so it will always be 0
double c3 = (f-32.0)*(5.0/9.0); // Correct
```

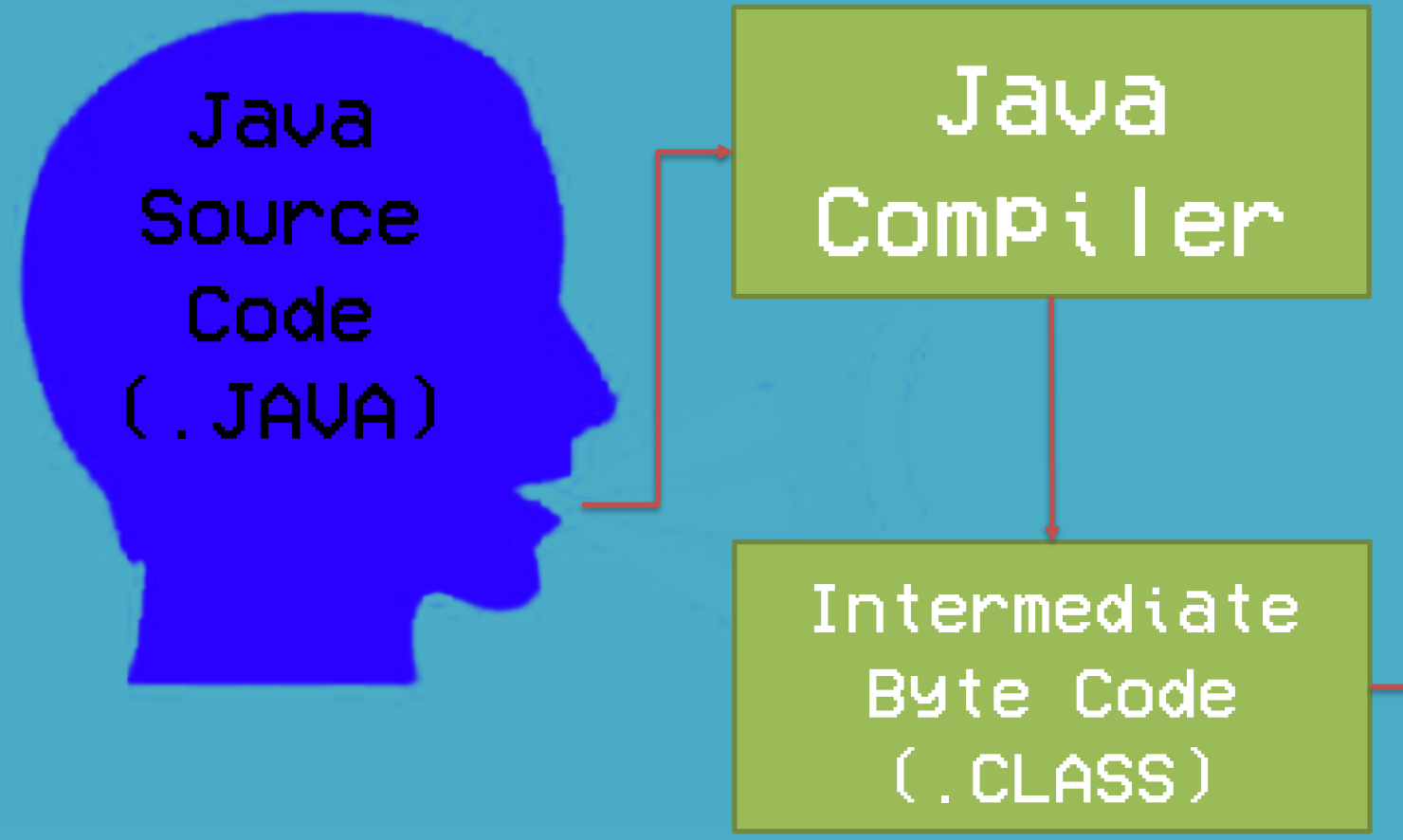
Writing and Running
a Program



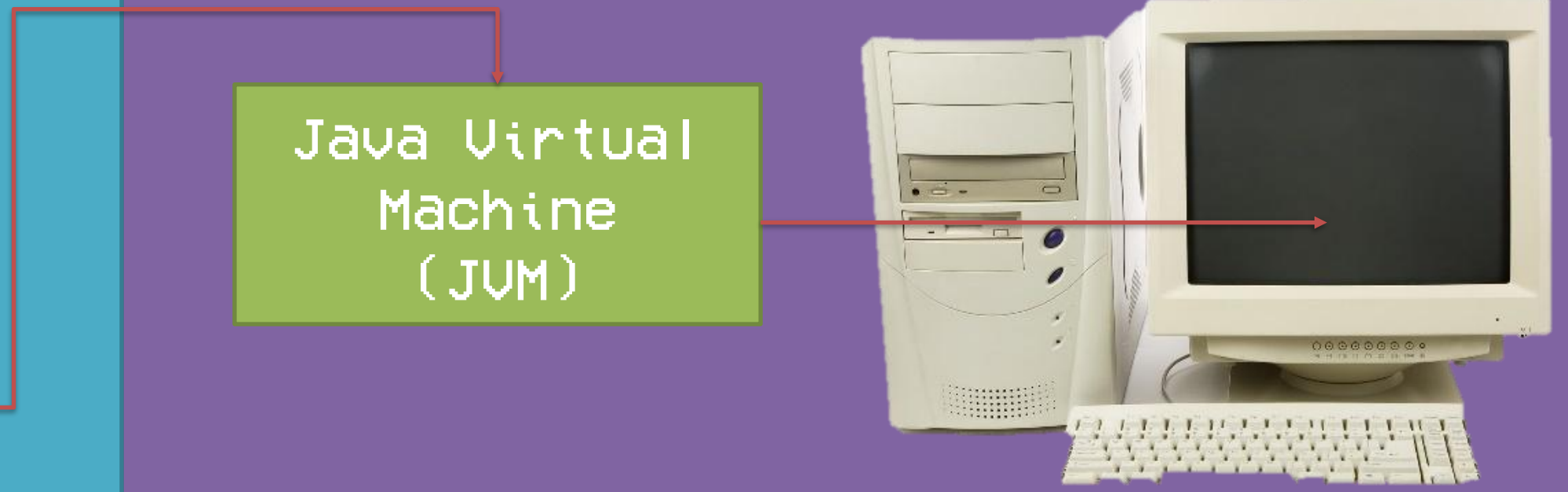
Integrated Development Environment (IDE)

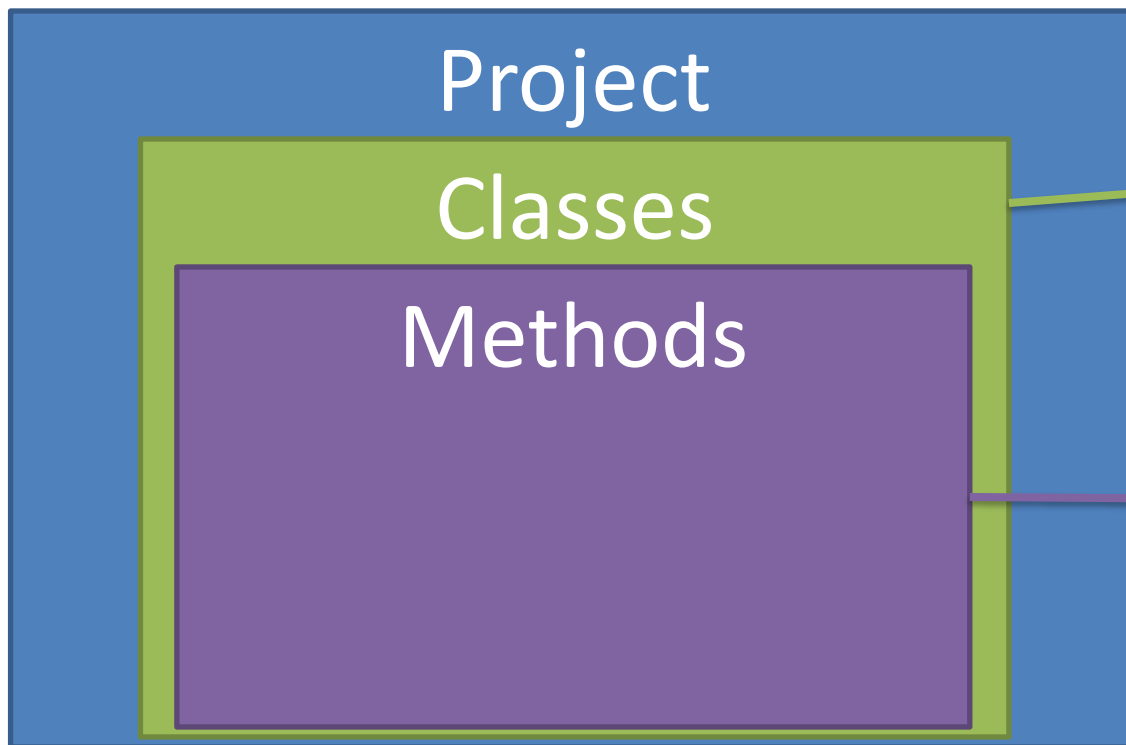
- Software for writing and testing Software.
- Not necessary but speeds up development.
- Eclipse, IntelliJ, VSCode, NetBeans, etc.

Compilation



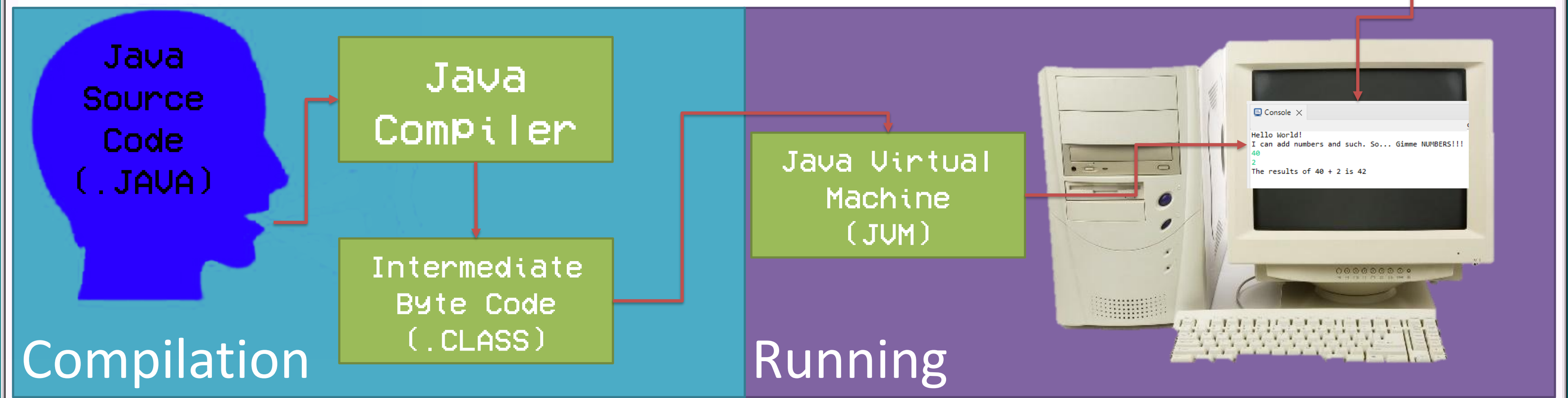
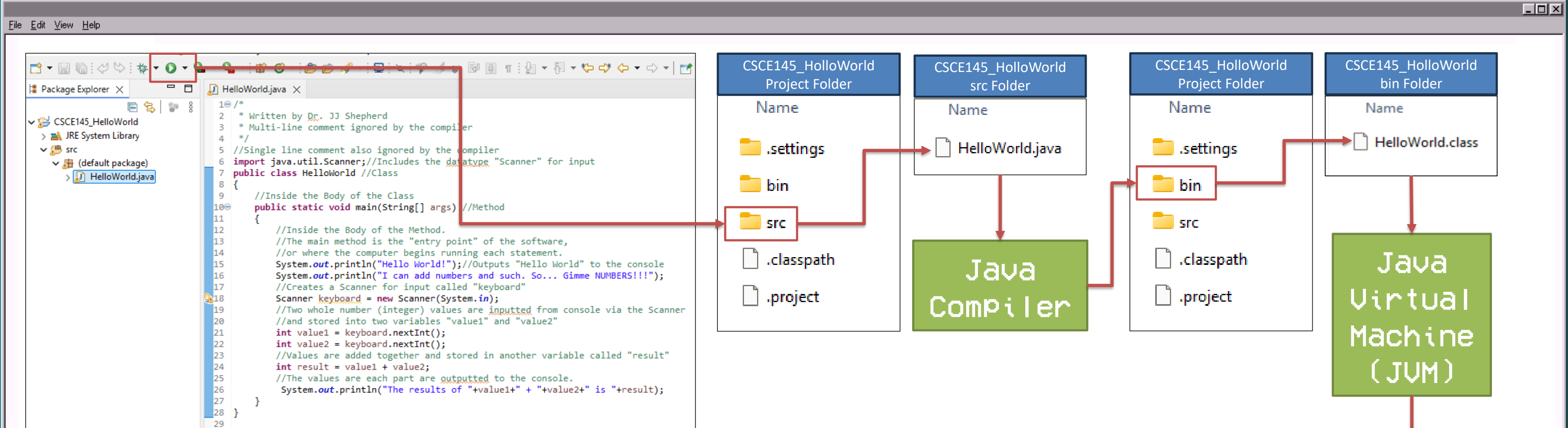
Running





The screenshot shows an IDE window with two panes. The left pane is the "Package Explorer" showing a project named "CSCE145_HelloWorld" with a "src" folder containing "HelloWorld.java". The right pane is the "HelloWorld.java" editor showing the following code:

```
1  /*
2  * Written by Dr. JJ Shepherd
3  * Multi-line comment ignored by the compiler
4  */
5  //Single line comment also ignored by the compiler
6  import java.util.Scanner; //Includes the datatype "Scanner" for input
7  public class HelloWorld //Class
8  {
9      //Inside the Body of the Class
10     public static void main(String[] args) //Method
11     {
12         //Inside the Body of the Method.
13         //The main method is the "entry point" of the software,
14         //or where the computer begins running each statement.
15         System.out.println("Hello World!"); //Outputs "Hello World" to the console
16         System.out.println("I can add numbers and such. So... Gimme NUMBERS!!!");
17         //Creates a Scanner for input called "keyboard"
18         Scanner keyboard = new Scanner(System.in);
19         //Two whole number (integer) values are inputted from console via the Scanner
20         //and stored into two variables "value1" and "value2"
21         int value1 = keyboard.nextInt();
22         int value2 = keyboard.nextInt();
23         //Values are added together and stored in another variable called "result"
24         int result = value1 + value2;
25         //The values are each part are outputted to the console.
26         System.out.println("The results of "+value1+ " + "+value2+ " is "+result);
27     }
28 }
29
```



HELLO WORLD!