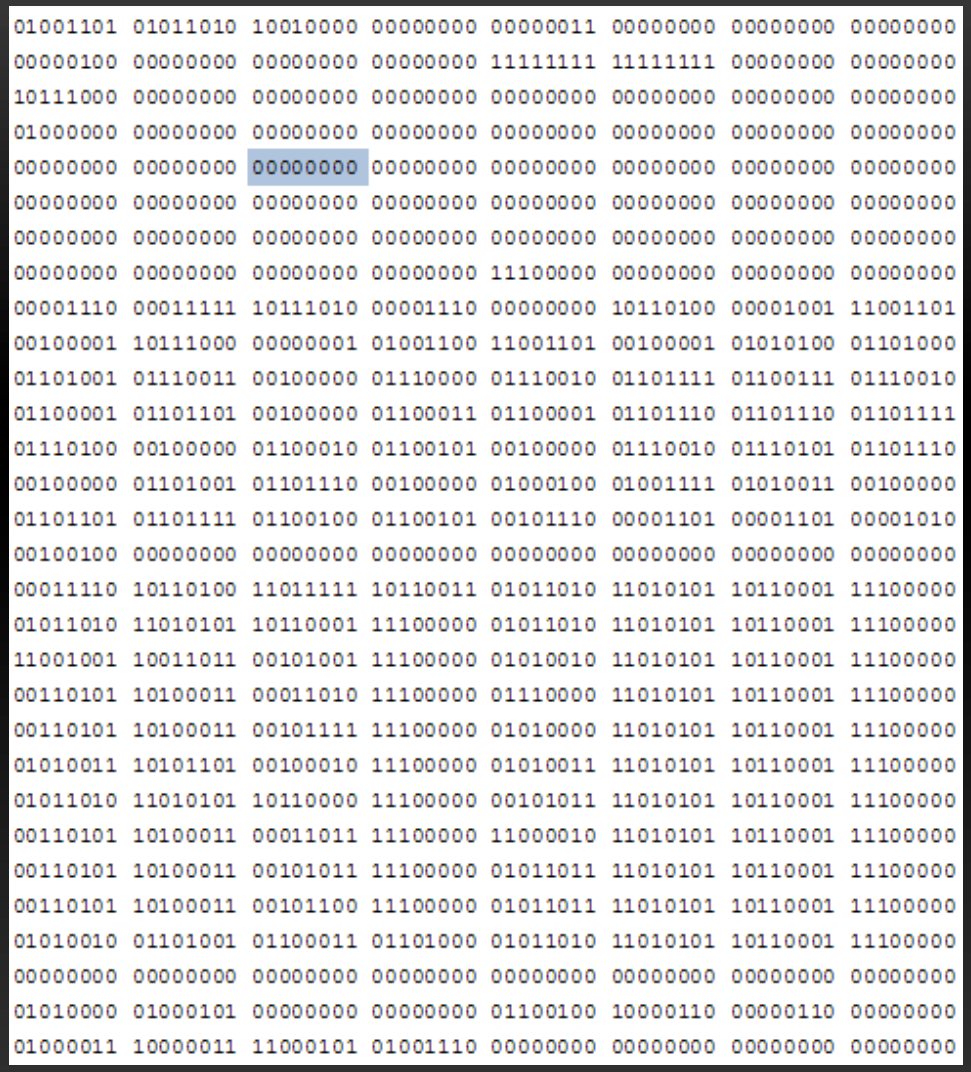
Learning Outcomes

1. Describe the history and evolution of programming languages
2. Categorize programming languages based on principal characteristics
3. Describe the fundamental elements of a programming language
4. Compare and contrast the 4 generations of programming languages
5. Identify a basic programming elements in simple computer programs

1st Generation

- The first programming language was the machine code itself, combinations of 1s and 0s
- As close to the machine as it gets, the machine's own language.
  - Specific to particular CPU architectures, a program written for one type of processor can only work on that type and nothing else
  - Hard to do, impossible for large programs
  - Not for humans… well not for most humans
  - This is how the first programs were programmed on the first computers
- The source code and the executable code are the same
- No compiler needed because no translation is required
- We don't do this anymore, not even in micro-controller programming



The beginning
of
"java.exe"
in binary

```
01010100 01101000
01101001 01110011 00100000
01110000 01110010 01101111
01100111 01110010 01100001
01101101 00100000 01100011
01100001 01101110 01101110
01101111 01110100 00100000
01100010 01100101 00100000
01110010 01110101 01101110
00100000 01101001 01101110
00100000 01000100 01001111
01010011 00100000 01101101
01101111 01100100 01100101
```

Binary for “This program cannot be run in DOS mode”

2nd Generation

- 1st generation was hard, too hard to create large programs
- The solution was to "name" the binary instructions using short but easier to remember "mnemonics" like…
  - add, sub, mov, call, push, pop, jmp, cmp
- Such programming languages are called assembly languages
- The program that translates assembly language into machine code is called assembler
  - Executable programs can be converted back to assembly using disassemblers
- Barely above 1s and 0s but much easier
  - We still write programs in assembly, parts of highly optimized programs like games, real-time systems, micro-controller programs etc.

```
00401000                        SUB_L00401000:
00401000 55                                     push    ebp
00401001 8BEC                                   mov     ebp,esp
00401003 51                                     push    ecx
00401004 51                                     push    ecx
00401005 53                                     push    ebx
00401006 57                                     push    edi
00401007 8BF8                                   mov     edi,eax
00401009 8D45F8                                 lea     eax,[ebp-08h]
0040100C 33DB                                   xor     ebx,ebx
0040100E 50                                     push    eax
0040100F 895DF8                                 mov     [ebp-08h],ebx
00401012 895DFC                                 mov     [ebp-04h],ebx
00401015 E8872A0000                             call    SUB_L00403AA1
0040101A 50                                     push    eax
0040101B 57                                     push    edi
0040101C E8BA5F0000                             call    SUB_L00406FDB
00401021 83C40C                                 add     esp,0000000Ch
00401024 83F801                                 cmp     eax,00000001h
00401027 7410                                   jz      L00401039
00401029                        L00401029:
00401029 33C0                                   xor     eax,eax
0040102B E988000000                             jmp     L004010B8
00401030                        L00401030:
00401030 3C30                                   cmp     al,30h
00401032 7C0B                                   jl      L0040103F
00401034 3C39                                   cmp     al,39h
00401036 7F07                                   jg      L0040103F
00401038 47                                     inc     edi
00401039                        L00401039:
```

The beginning of **java.exe** in **assembly language**

3rd Generation

- As programs became more complicated the need for programming languages closer to "human" languages grew
- Software development processes evolved with new constructs, better ways of creating complex, robust, easier to maintain programs
- The program that translates source code of a 3rd generation language into machine language is called a compiler
  - Only the compiler designer needs to know machine language!
- 3rd generation programming languages were a huge leap in computer science
- Also called high-level languages

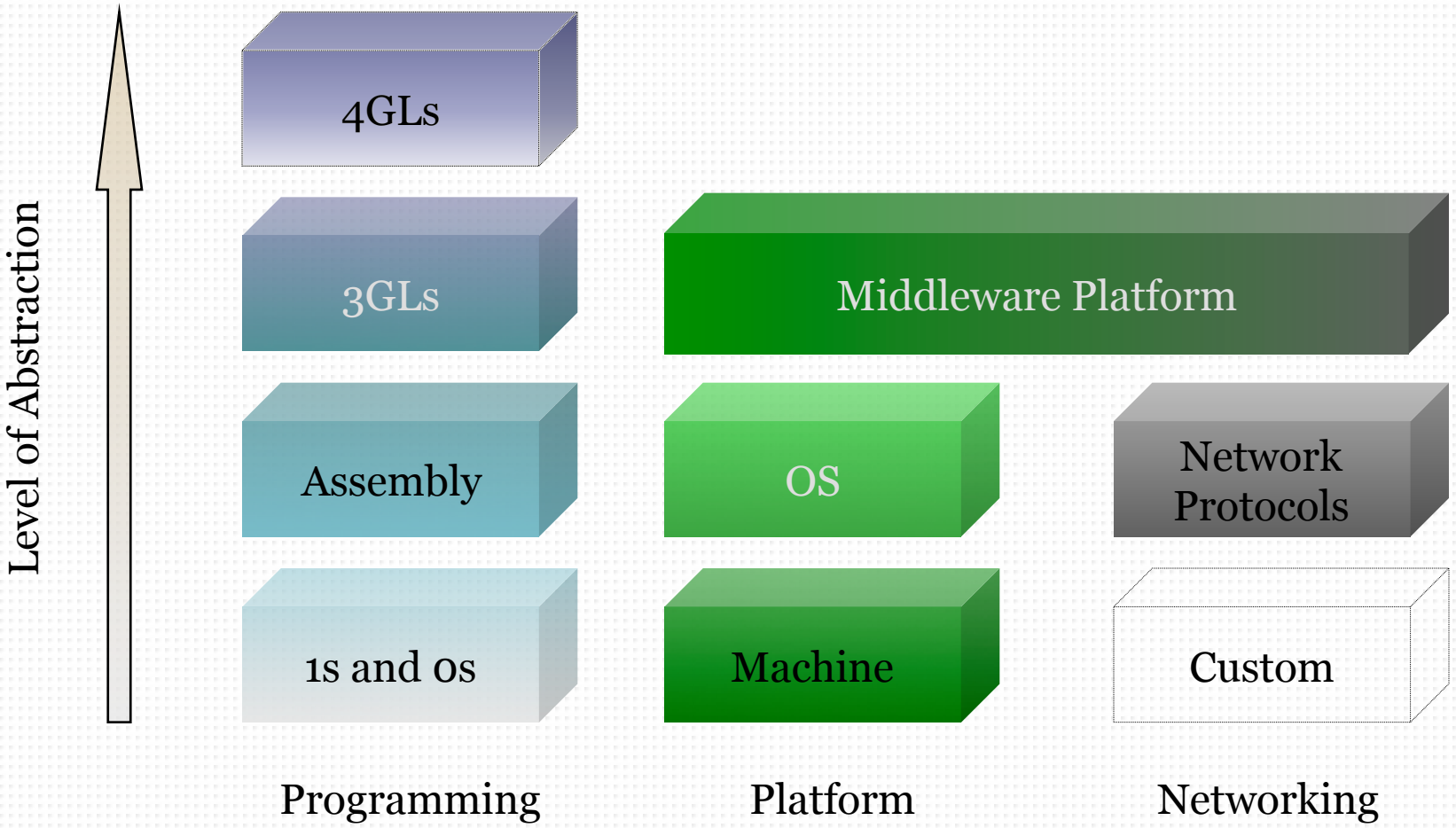
3rd Generation Programming Languages

- Modern languages
  - Java
  - C#, Visual Basic.NET
  - C++
  - C
  - Delphi
  - Visual Basic
  - Pascal
  - PHP
- Legacy languages, still in use today
  - COBOL
  - FORTRAN
  - Ada

4th and 5th Generation Languages

- In 3rd generation languages you still write commands telling the computer what to do.
- How can one improve that?
  - By telling the computer what you want instead
  - SQL (Structured Query Language) and FoxPro for database development
  - Programs that generate other programs, like auto code gen. in some IDEs
  - These can be considered 4th generation languages, but true, general purpose 4th gen languages are still a research area
- 5th generation languages – solve problems by specifying constraints, not by giving an algorithm, examples: Prolog, Datalog
  - Artifical intelligence, logic programming
  - Example applications: Medical diagnosis, robot planning, music composition

Evolution in Computing



So what exactly is a programming language?

It is just a language like any other…!

Example program written in Java

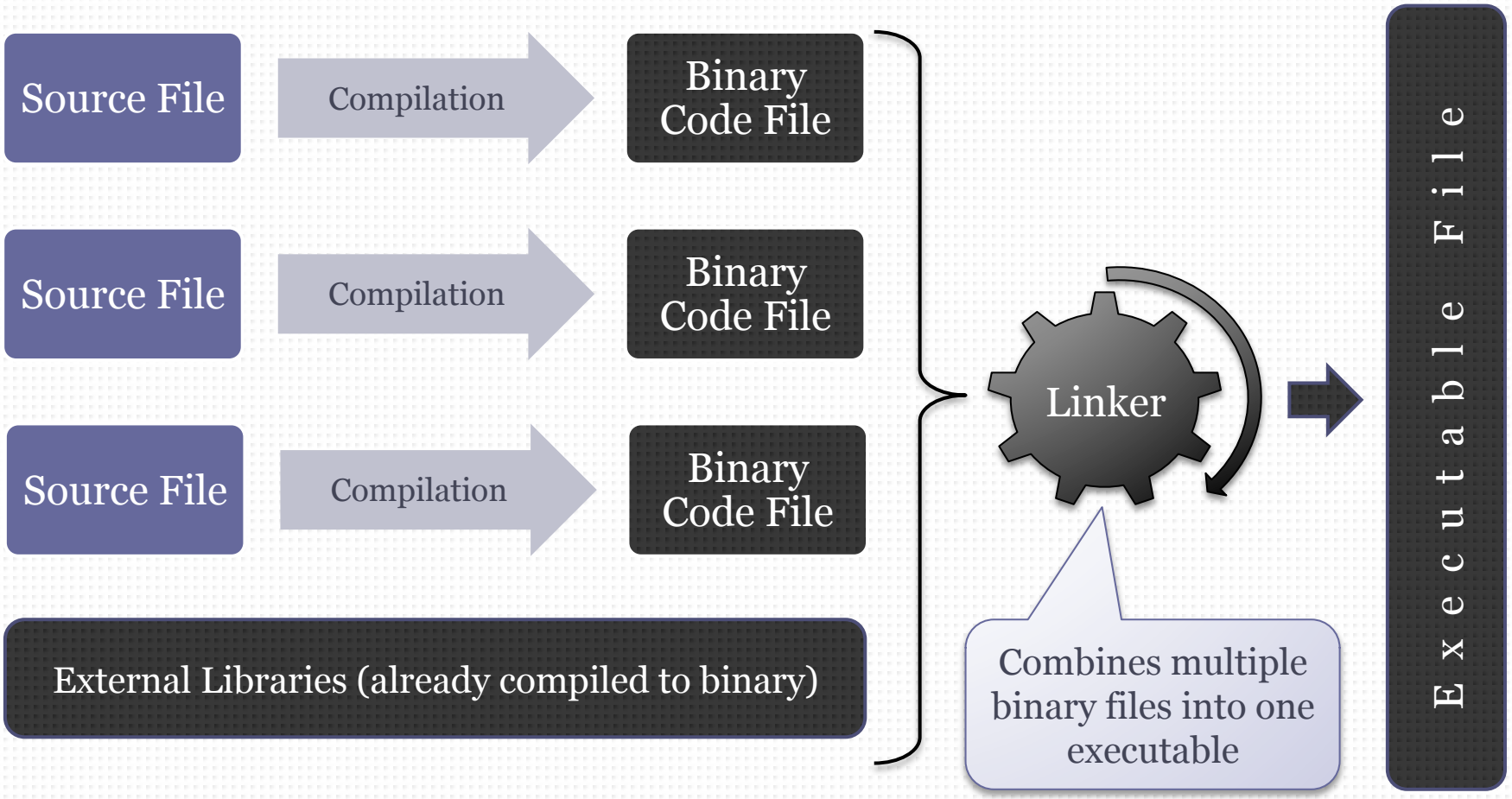
```
package sheridan;

public class Program {
    public static void main(String[] args) {
        // Write a message to the screen
        System.out.println("Hello PROG10082 class");
        int result = 2 + 2;
        System.out.println("The answer is " + result);
    }
}
```

Just a Language

- A programming language is a lot like any other language (e.g. English, Spanish, Punjabi) but simpler and more precise
- Alphabet: letters, numbers, math symbols, parenthesis and brackets, semi-colon, slashes, spaces, tabs
  - Largely the same as the English alphabet
- Vocabulary: the words that have a defined meaning.
  - A programming language has a MUCH SMALLER vocabulary
  - Words in a programming language vocabulary are called "reserved words" or "keywords"
  - We can invent new words in our programs but we must first declare them and define them. They cannot be the same as the reserved words.
- Syntax: the way words are put together to convey meaning
  - Very important to a programming language and very exact. The computer only understands exact rules

Many Source Files to One Executable



Review: Source Code

- What we write and what the compiler translates
- Source code is stored (saved) in source files
  - These are text files
- Source files are stored (saved) on storage devices like a hard drive
- Source code is a set of commands or instructions written in a programming language
- A compiler reads source files, processes all the commands and translates them into an executable file that contains machine code
  - The process of transforming a source file into an binary file is called compilation, or compiling
- A program is a collection of source files