Chapter 1

Introduction to Computers and Java

Objectives of Chap 1

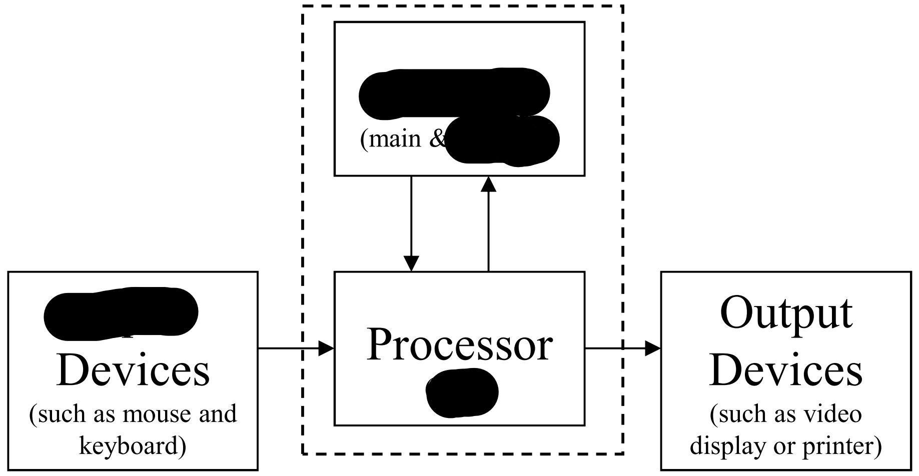
- 1) a brief overview of computer hardware and software
- 2) basic techniques of program design In general and object oriented programming in particular.
- 3) an overview of the java programming language.
- 4) introduce to applets and some graphics basics.

1.1 Computer Basics

- Computer system: hardware + software
- Hardware: the [redacted] components(machine)
- Program : a set of [redacted] for computer
- Software: All the [redacted] kinds of programs used to give instructions to the computer

Common Hardware Components

Standard Hardware Organization



- Processor (CPU)
  - » Central [redacted] Unit
  - » [redacted] and [redacted] the instructions
- Memory
  - » main & auxiliary
  - » holds [redacted] and [redacted]
- Input device(s)
  - » mouse, keyboard, etc.
- Output device(s)
  - » video display, printer, etc.
- CPU and memory are physically [redacted] together

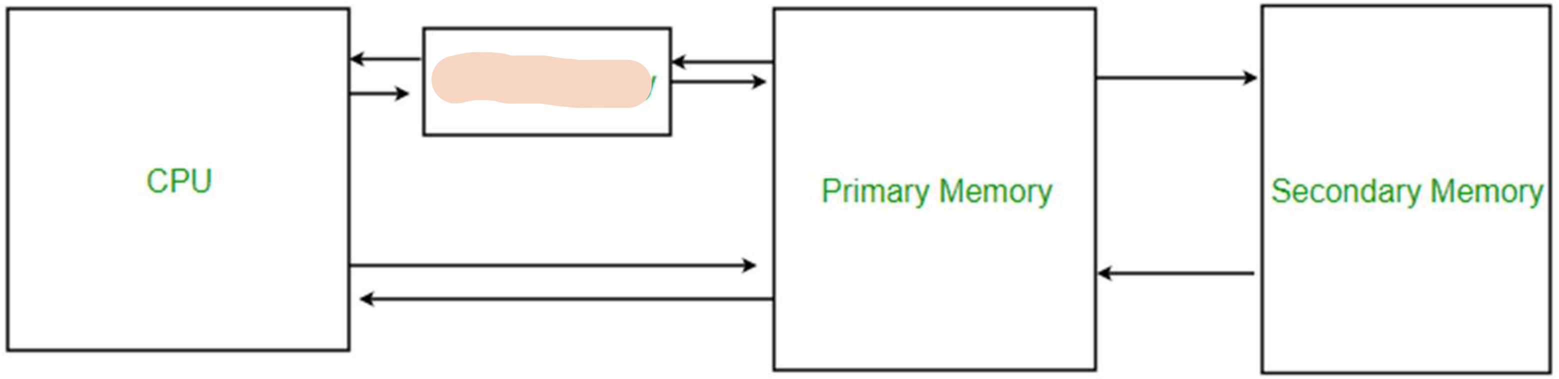
Physical Organization

- Keyboard
- Monitor
- [redacted]
  - » CPU
  - » memory
  - » disk drives
  - » I/O connectors
  - » etc.

Two Kinds of Memory

- Main
  - » working area
  - » [redacted] stores program and data (while program is [redacted])
- [redacted]
  - » [redacted] (more or less) memory
  - » Secondary memory
  - » saves program and results
  - » includes floppy & hard disk drives, CDs, tape, etc.
- (Cache Memory??)—Program Code [redacted]

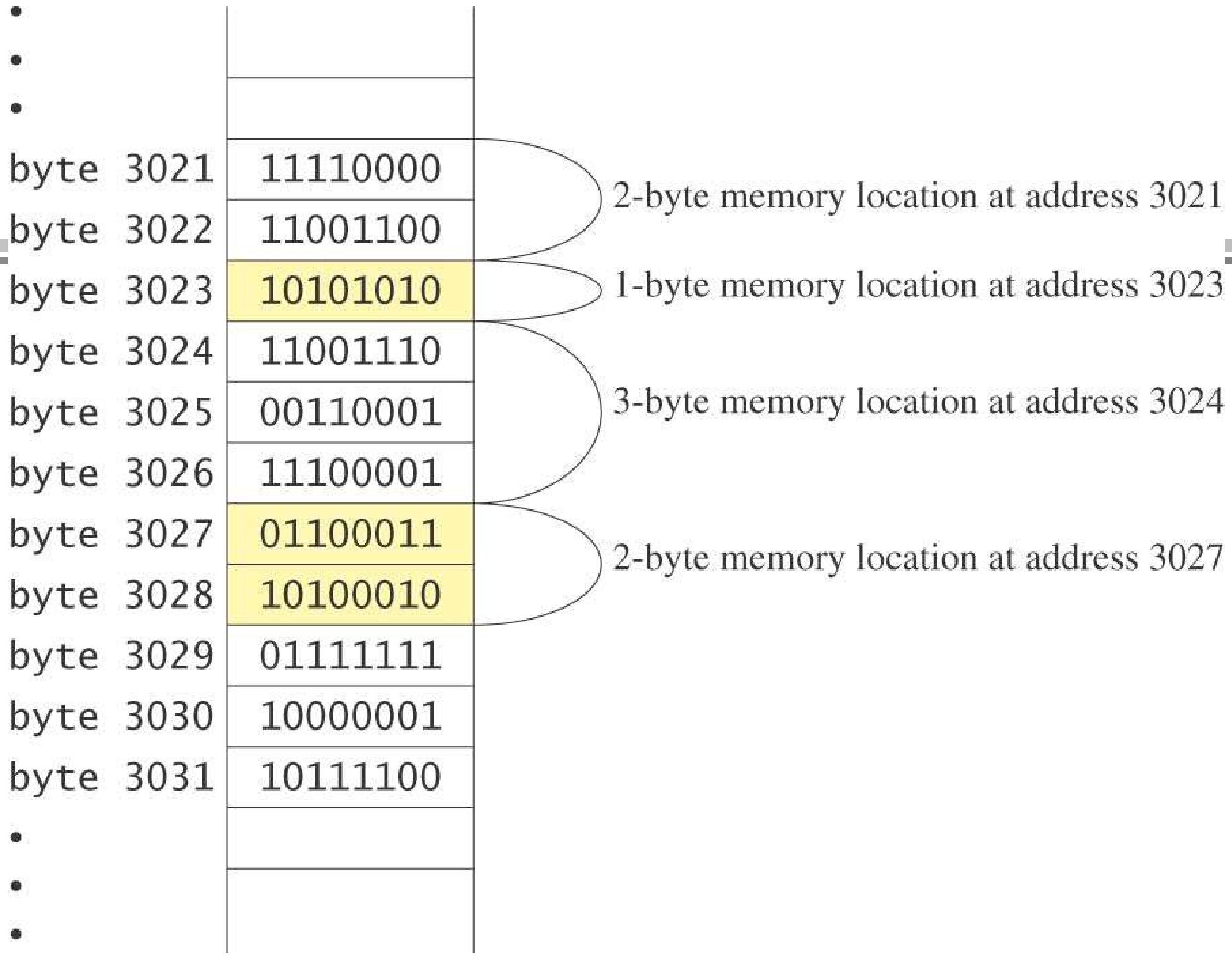
Cache memory



Main Memory Organization

- Bit = one binary digit
  - » Binary digit can have only one of two values, 0 or 1
- Byte = 8 bits
- "Byte Addressable"
  - » Main memory is a list of numbered locations that contain one [redacted] of data in each location
- Number of bytes per data item may vary

Address	Data Byte	
3021	1111 0000	Item 1: 2 bytes stored
3022	1100 1100	
3023	1010 1010	Item 2: 1 byte stored
3024	1100 1110	Item 3: 3 bytes stored
3025	0011 0001	
3026	1110 0001	
3027	0110 0011	Item 4: 2 bytes stored
3028	1010 0010	
3029	…	Next Item, etc.



Display 1.1

Main Memory

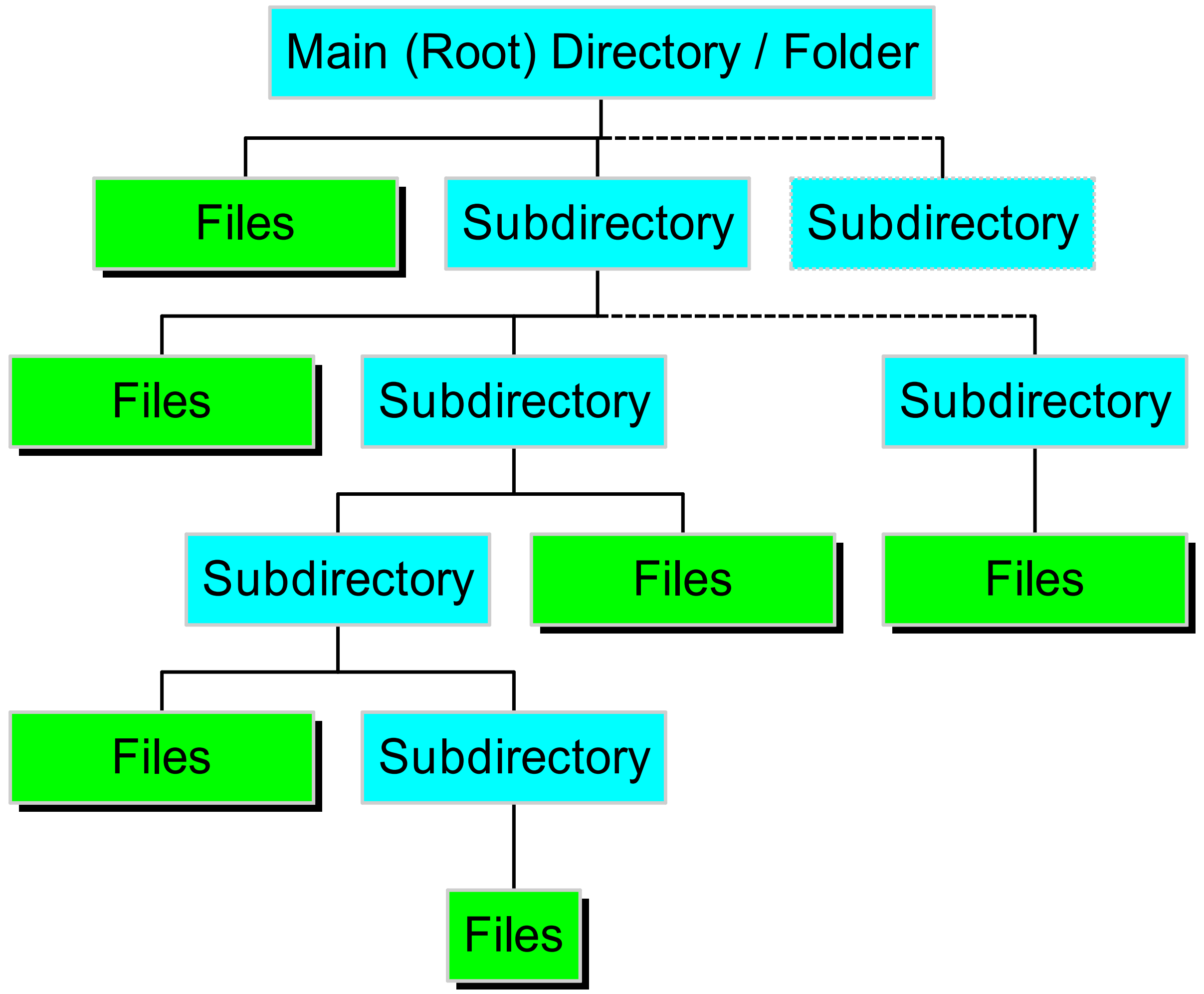
Why Just Zero and Ones?

- It is easy to make a physical device that has only two stable states.

- 전산전자가 제일 좋아하는 노래
  - »
- 전산전자가 제일 좋아하는 친구
  - »

낭만에 대하여

Memory Organization

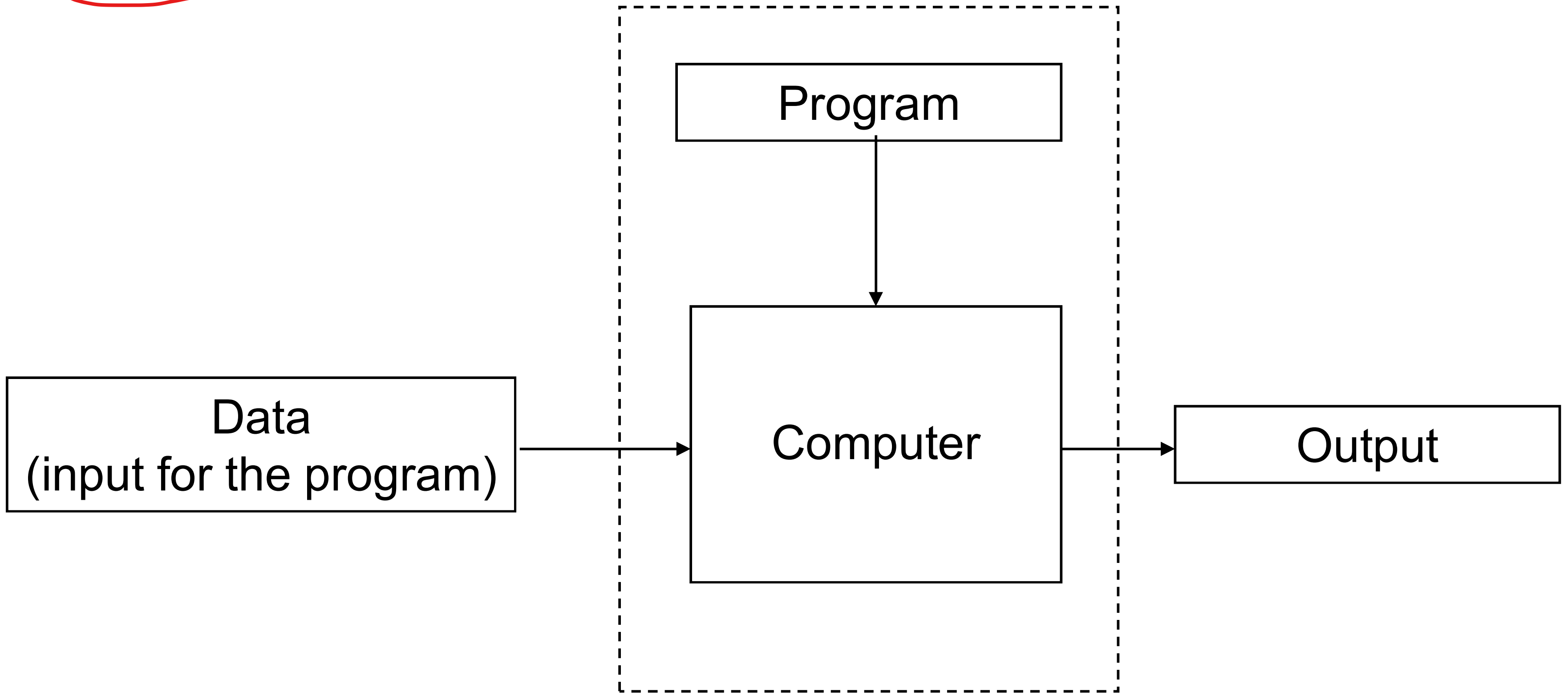


Questions

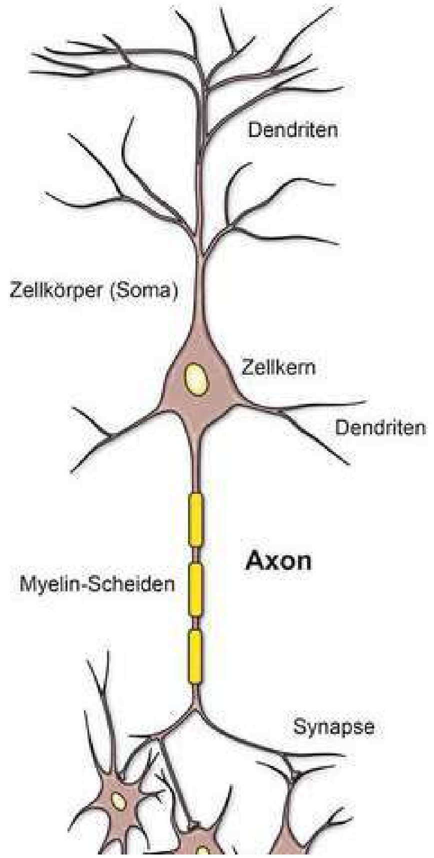
- Von-neumann architecture ?
  - » merits and demerits ?
- Non Von neumann architecture ?
- Brain-like architecture ?
- Stored Program ?

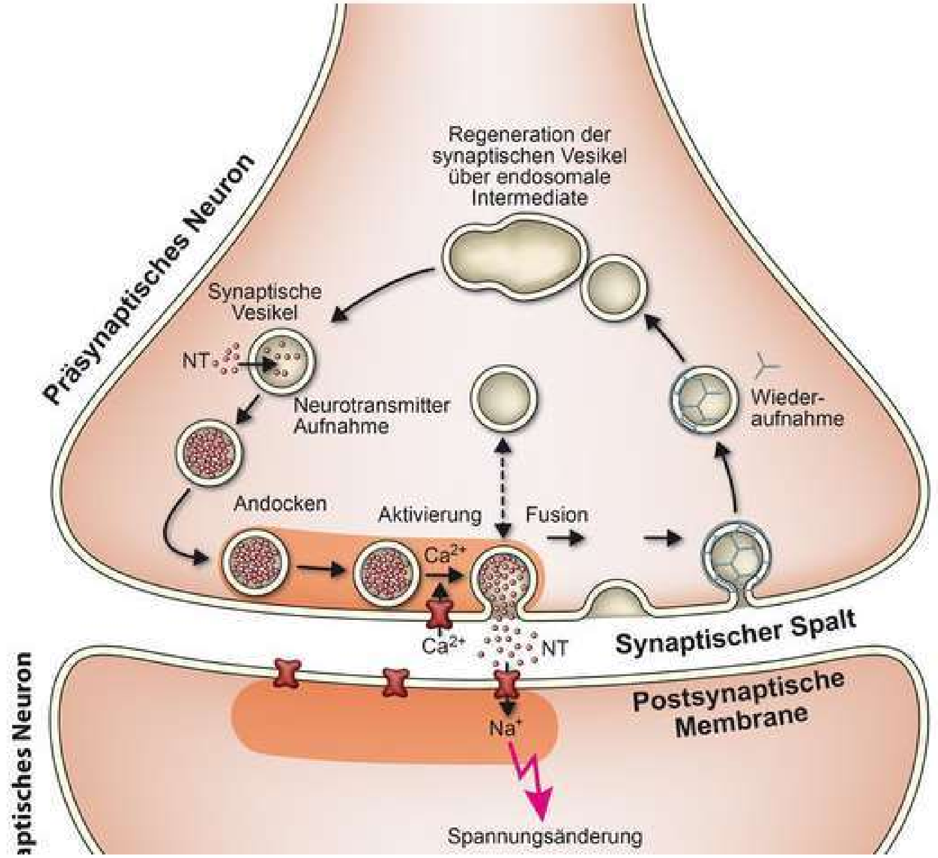
Display 1.2 Running a Program

[redacted]—a set of instructions for a computer to follow



Non-von Neumann architecture brain like computer



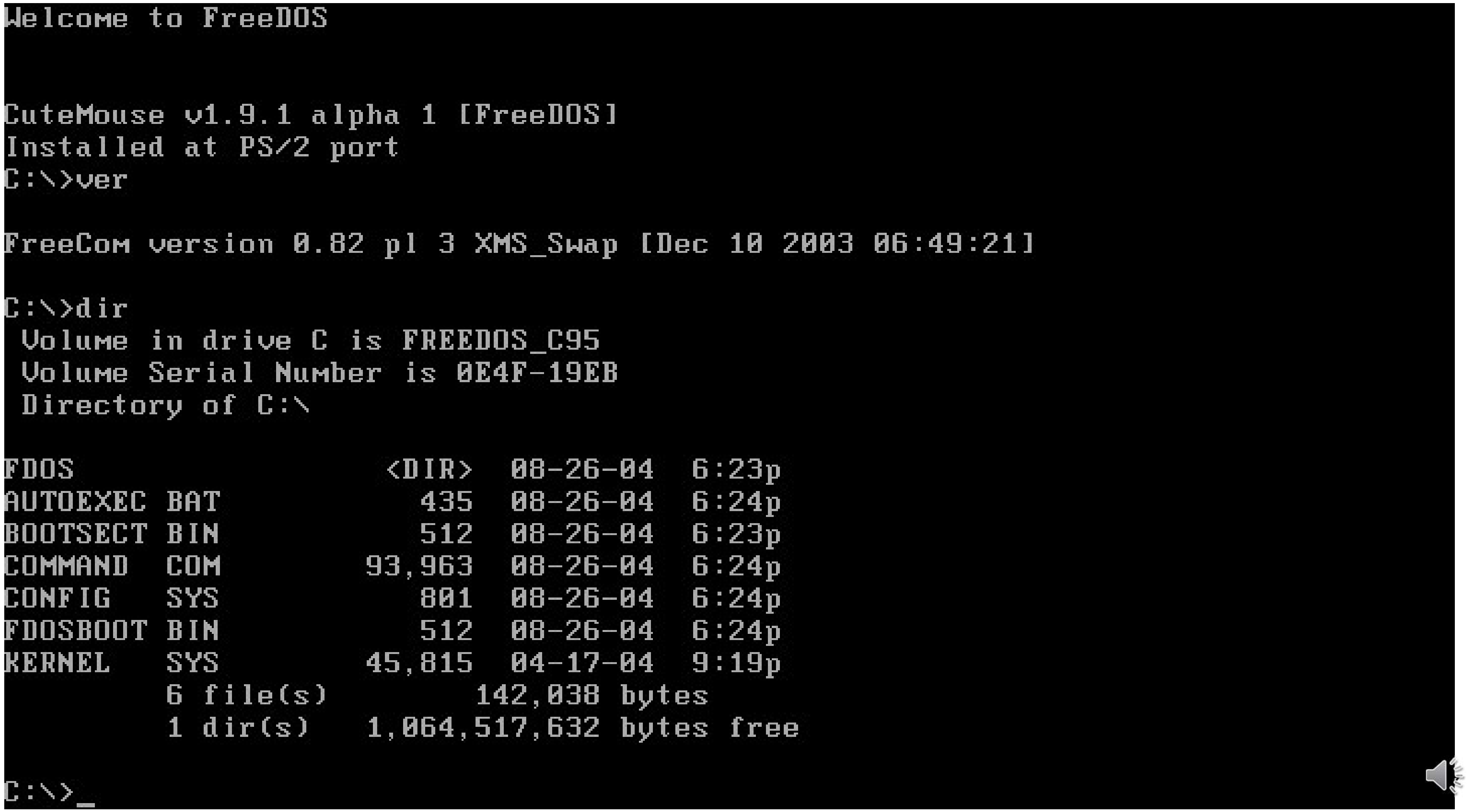


Many Types of [redacted]

- User-created applications
- Existing applications
  - » word-processor/editor
  - » web browser
  - » compiler or assembler , etc.
- Operating System
  - » DOS, Microsoft Windows, MacOS, Linux, UNIX, etc.
  - » MacOS ??

Various Types of [redacted]

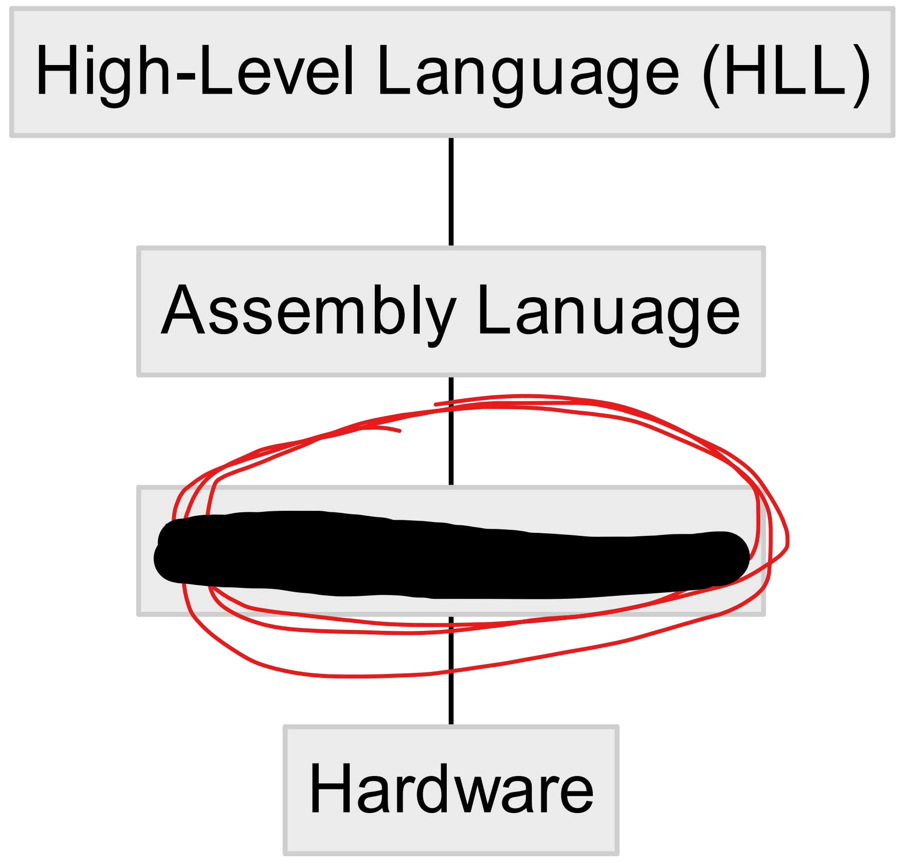
- Command-line
  - » type in key words and letters
  - » DOS and UNIX
- Menu
  - » parts of DOS and Windows
- [redacted] (Graphical User Interface)
  - » click on icon
  - » also called [redacted]-driven”
  - » MacOS, Win[redacted]ows



```
Welcome to FreeDOS


CuteMouse v1.9.1 alpha 1 [FreeDOS]
Installed at PS/2 port
C:\>ver

FreeCom version 0.82 pl 3 XMS_Swap [Dec 10 2003 06:49:21]

C:\>dir
 Volume in drive C is FREEDOS_C95
 Volume Serial Number is 0E4F-19EB
 Directory of C:\

FDOS                 <DIR>  08-26-04  6:23p
AUTOEXEC BAT           435  08-26-04  6:24p
BOOTSECT BIN           512  08-26-04  6:23p
COMMAND  COM        93,963  08-26-04  6:24p
CONFIG   SYS           801  08-26-04  6:24p
FDOSBOOT BIN           512  08-26-04  6:24p
KERNEL   SYS        45,815  04-17-04  9:19p
         6 file(s)         142,038 bytes
         1 dir(s)   1,064,517,632 bytes free

C:\>_
```

EC5274 SS/SD
ИЗОТ
СДЕЛАНО В БОЛГАРИИ
boedem
FLOPPY DISK
5,25" HD,DS
WRITE ENABLE
WRITE PROTECT

Programming Language Hierarchy



The highs and lows of programming languages ...

High-Level Language (HLL)

» closest to natural language
» words, numbers, and math symbols
» not directly understood by hardware
» “portable” source code (hardware [redacted])
» Java, C, C++, COBOL, FORTRAN, BASIC, Lisp, Ada, etc.

Machine Language
(lowest level)

» least natural language for humans, most natural language for hardware
» just 0s and 1s
» directly understood by hardware
» not portable (hardware [redacted])

[redacted] (middle level)

- a more or less human readable version of machine language
- words, abbreviations, letters and numbers replace 0s and 1s
- easily translated from human readable to machine executable code
- like machine code, not portable (hardware [redacted])

Examples & Questions ??

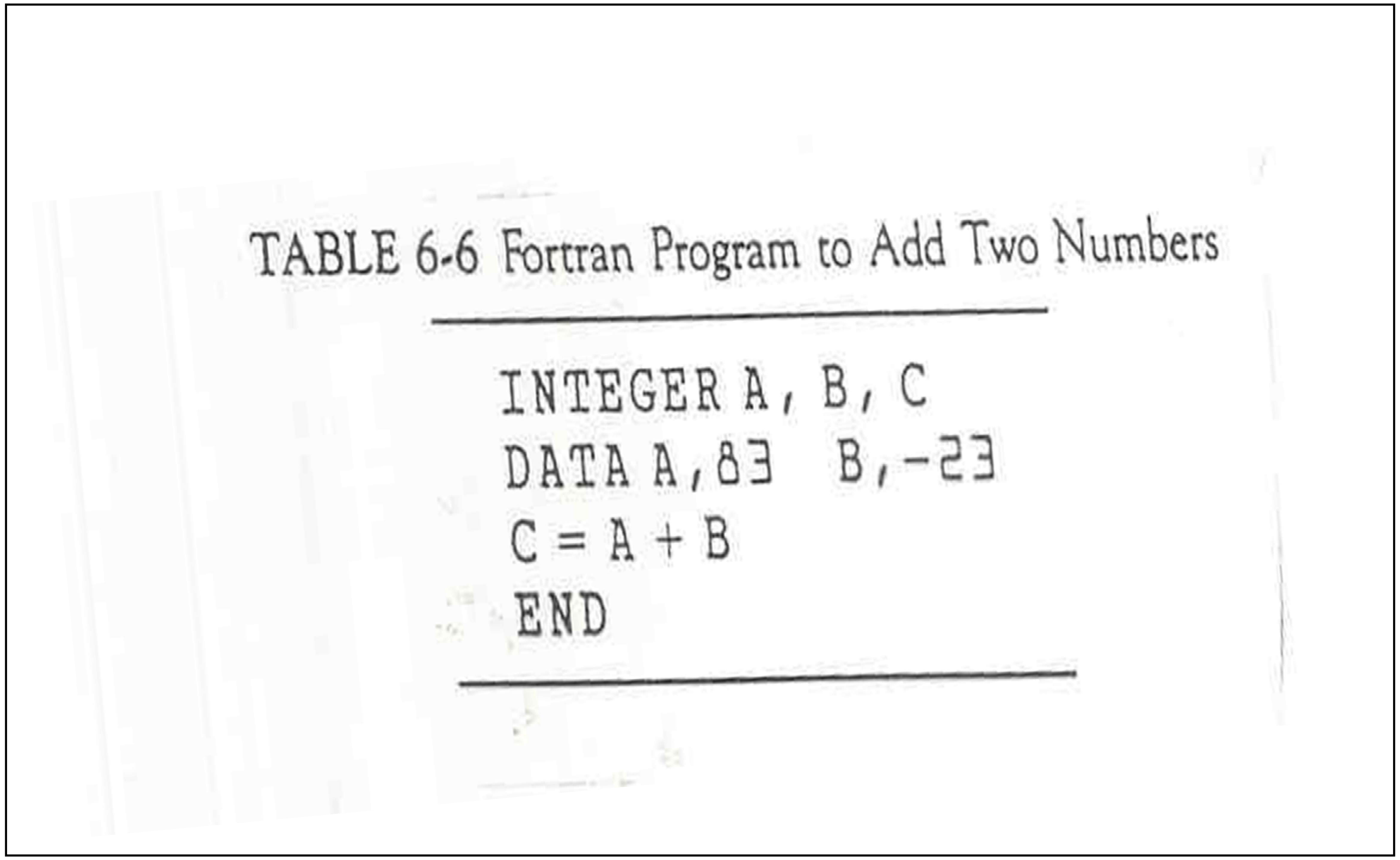


TABLE 6-6 Fortran Program to Add Two Numbers

```
INTEGER A, B, C
DATA A,83  B,-23
C = A + B
END
```

TABLE 6-2 Binary Program to Add Two Numbers

Location	Instruction code
0	0010 0000 0000 0100
1	0001 0000 0000 0101
10	0011 0000 0000 0110
11	0111 0000 0000 0001
100	0000 0000 0101 0011
101	1111 1111 1110 1001
110	0000 0000 0000 0000

TABLE 6-3 Hexadecimal Program to Add Two Numbers

Location	Instruction
000	2004
001	1005
002	3006
003	7001
004	0053
005	FFE9
006	0000

TABLE 6-4 Program with Symbolic Operation Codes

Location	Instruction	Comments
000	LDA 004	Load first operand into *AC*
001	ADD 005	Add second operand to *AC*
002	STA 006	Store sum in location 006
003	HLT	Halt computer
004	0053	First operand
005	FFE9	Second operand (negative)
006	0000	Store sum here

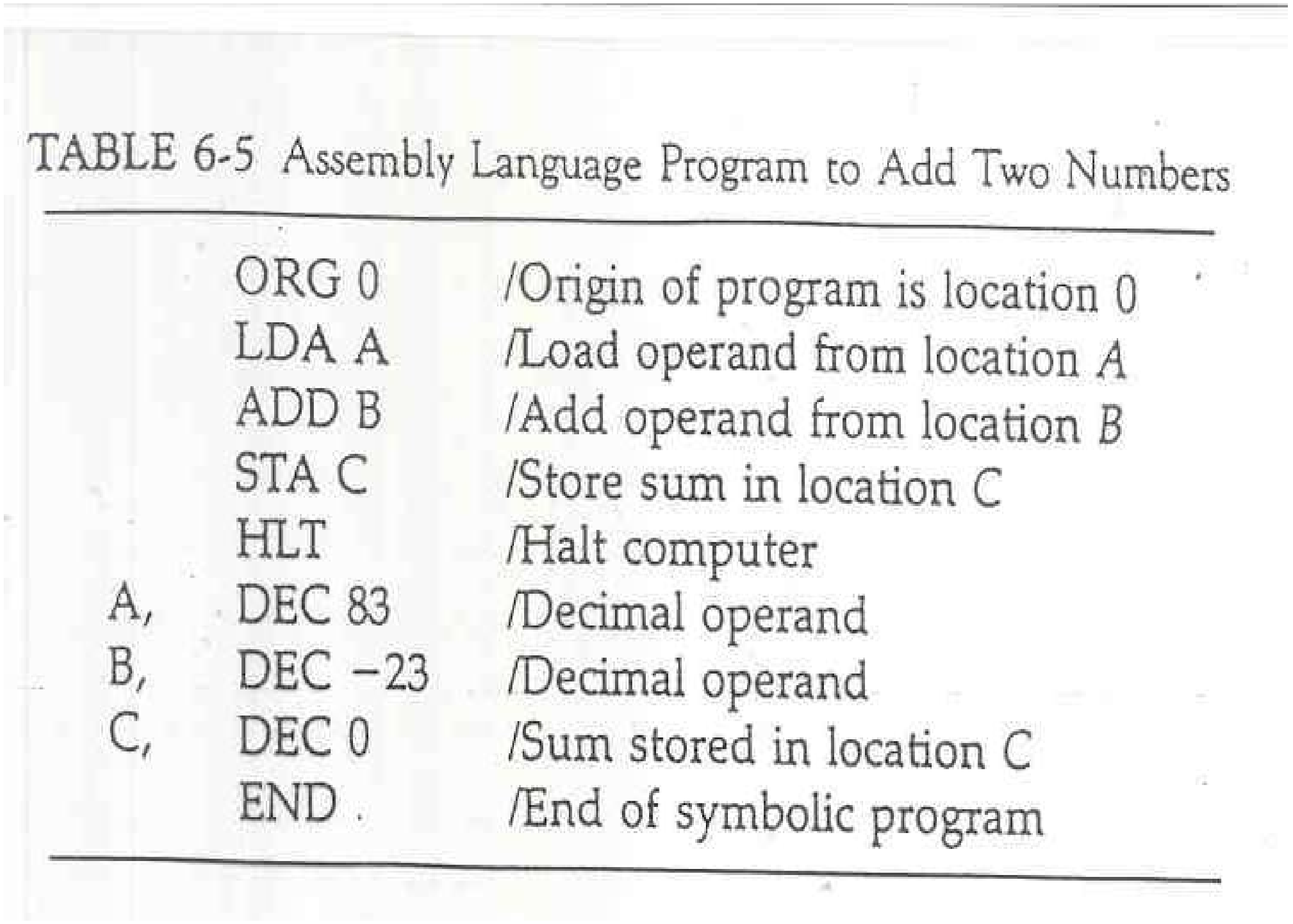
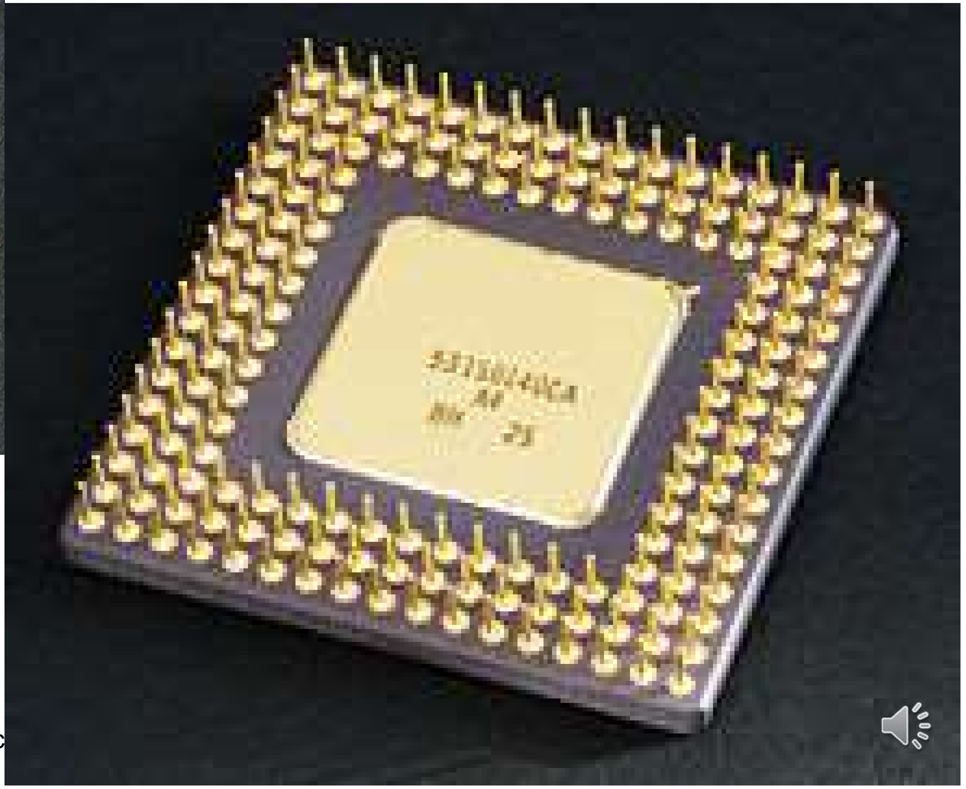


TABLE 6-5 Assembly Language Program to Add Two Numbers

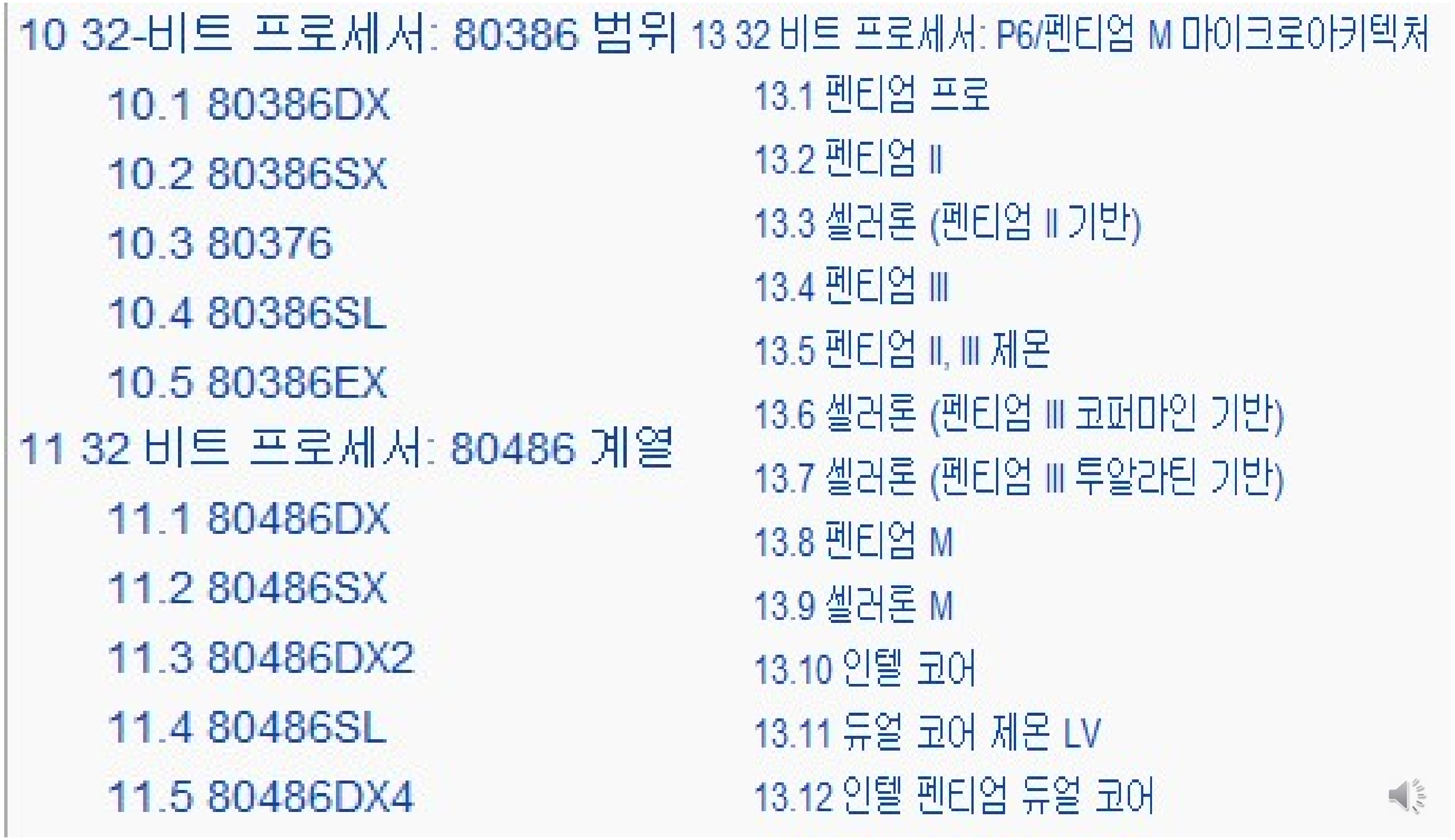
```
        ORG 0       /Origin of program is location 0
        LDA A       /Load operand from location A
        ADD B       /Add operand from location B
        STA C       /Store sum in location C
        HLT         /Halt computer
A,      DEC 83      /Decimal operand
B,      DEC -23     /Decimal operand
C,      DEC 0       /Sum stored in location C
        END         /End of symbolic program
```

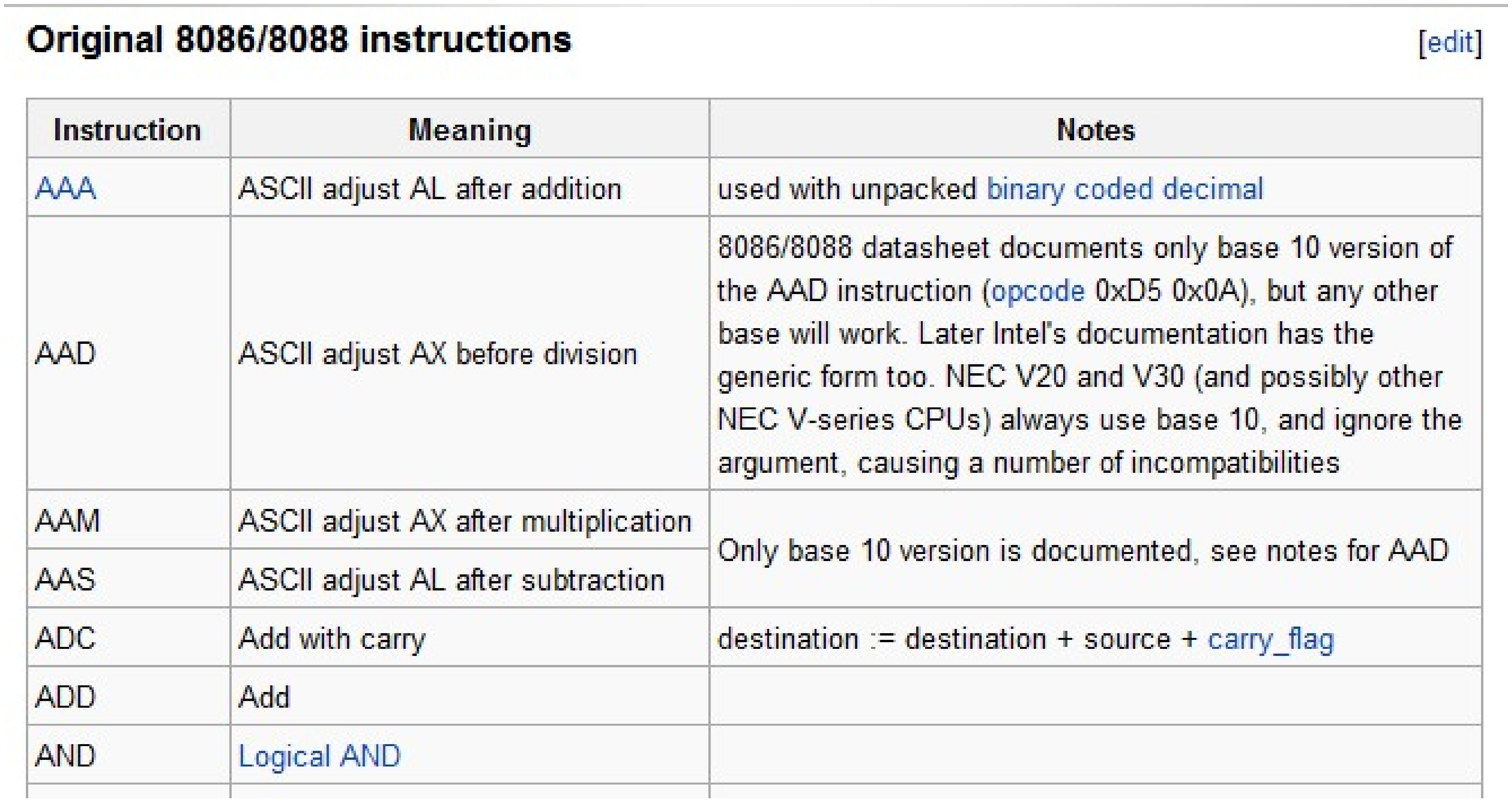
CPU & instruction Set



갤럭시 S/S2의 CPU

- 갤럭시S
  - » 삼성전자가 자체 개발한 1㎓급 CPU인 `S5PC111'이 탑재돼 있음
- 갤럭시 S2
  - » 엑시노스 **(EXYNOS 4210, 4212)**
  - » 삼성의 ARM Coretex-A9 계열 프로세서의 SoC (System on Chip)

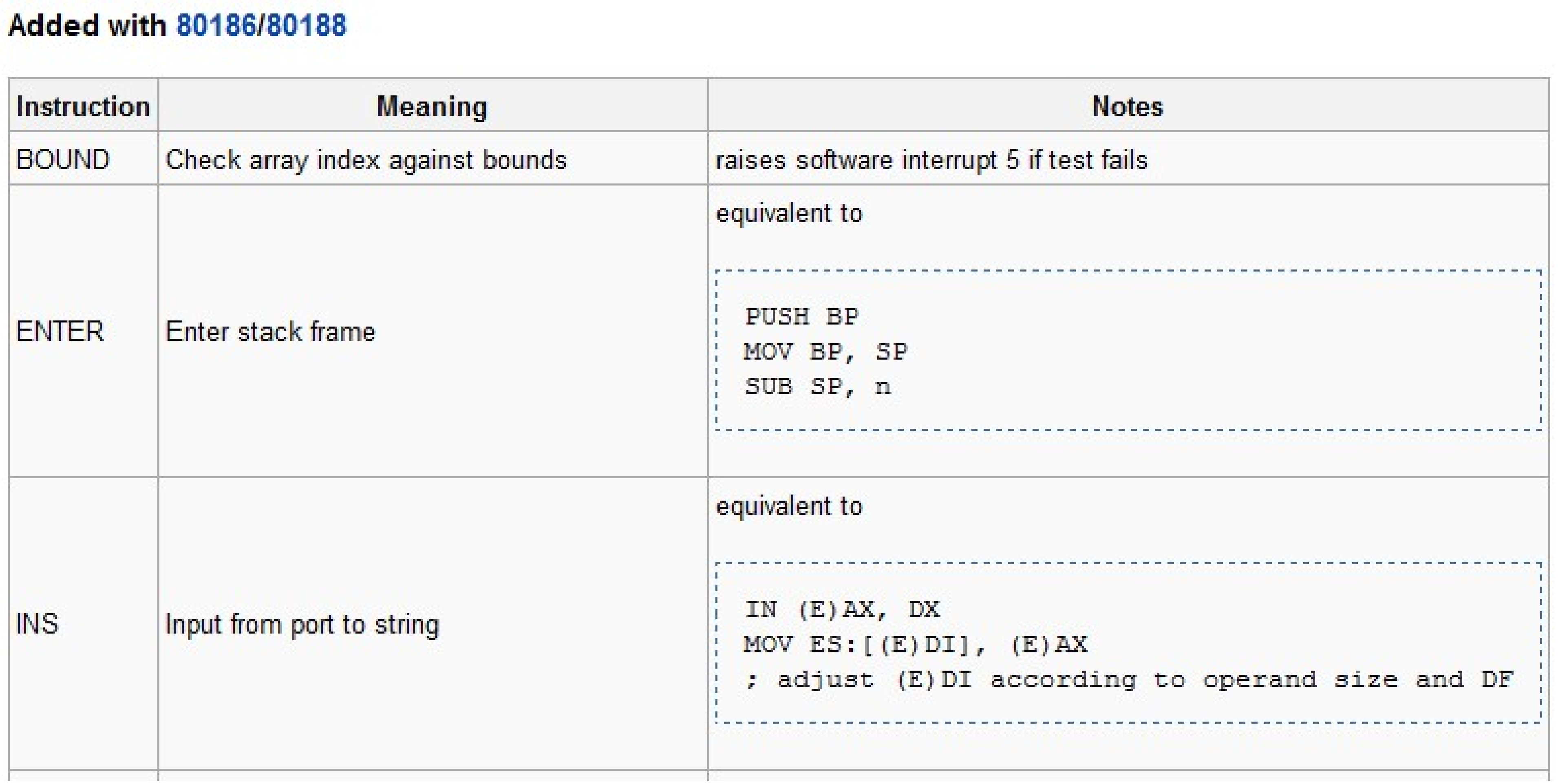




Original 8086/8088 instructions

[edit]

Instruction	Meaning	Notes
AAA	ASCII adjust AL after addition	used with unpacked binary coded decimal
AAD	ASCII adjust AX before division	8086/8088 datasheet documents only base 10 version of the AAD instruction (opcode 0xD5 0x0A), but any other base will work. Later Intel's documentation has the generic form too. NEC V20 and V30 (and possibly other NEC V-series CPUs) always use base 10, and ignore the argument, causing a number of incompatibilities
AAM	ASCII adjust AX after multiplication	Only base 10 version is documented, see notes for AAD
AAS	ASCII adjust AL after subtraction	
ADC	Add with carry	destination := destination + source + carry_flag
ADD	Add	
AND	Logical AND	



Added with 80186/80188

Instruction	Meaning	Notes
BOUND	Check array index against bounds	raises software interrupt 5 if test fails

ENTER	Enter stack frame	equivalent to
		`PUSH BP`
		`MOV BP, SP`
		`SUB SP, n`

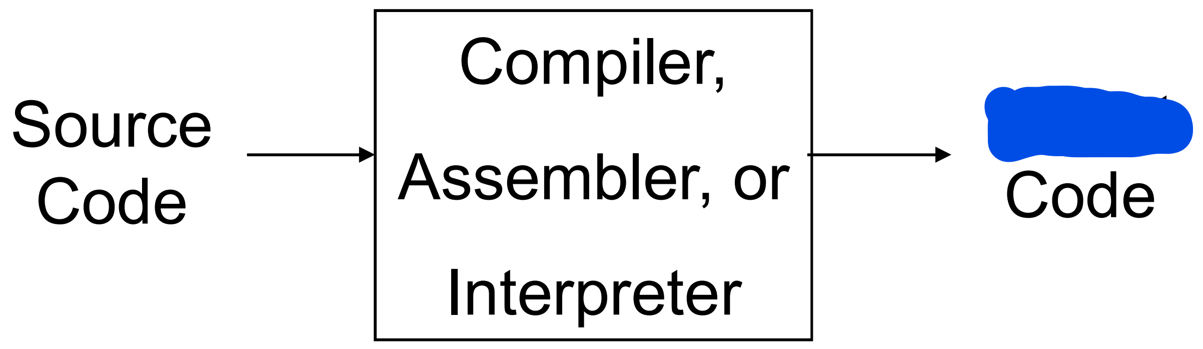
INS	Input from port to string	equivalent to
		`IN (E)AX, DX`
		`MOV ES:[(E)DI], (E)AX`
		`; adjust (E)DI according to operand size and DF`

Getting from Source to Machine Code

- “[redacted] *a program”*
  translating from a high-level language source code to machine (object, or executable) code.
- [redacted]
  a program that translates HLL source code to machine (object, or executable) code.
- [redacted]”
  translating from assemble language source code to machine (object, or executable) code.
- [redacted]
  a program that translates assembly source code to machine (object, or executable) code.
- Compilers need to know the specific target hardware

Compilers vs. Assemblers vs. Interpreters

- Compilers and Assemblers
  - » translation is a [redacted] user step
  - » translation i[redacted].e. not at run time
- Interpreters - another way to translate source to object code
  - » interpretation (from source to object code) is not a separate user step
  - » translation is[redacted]e. at run time

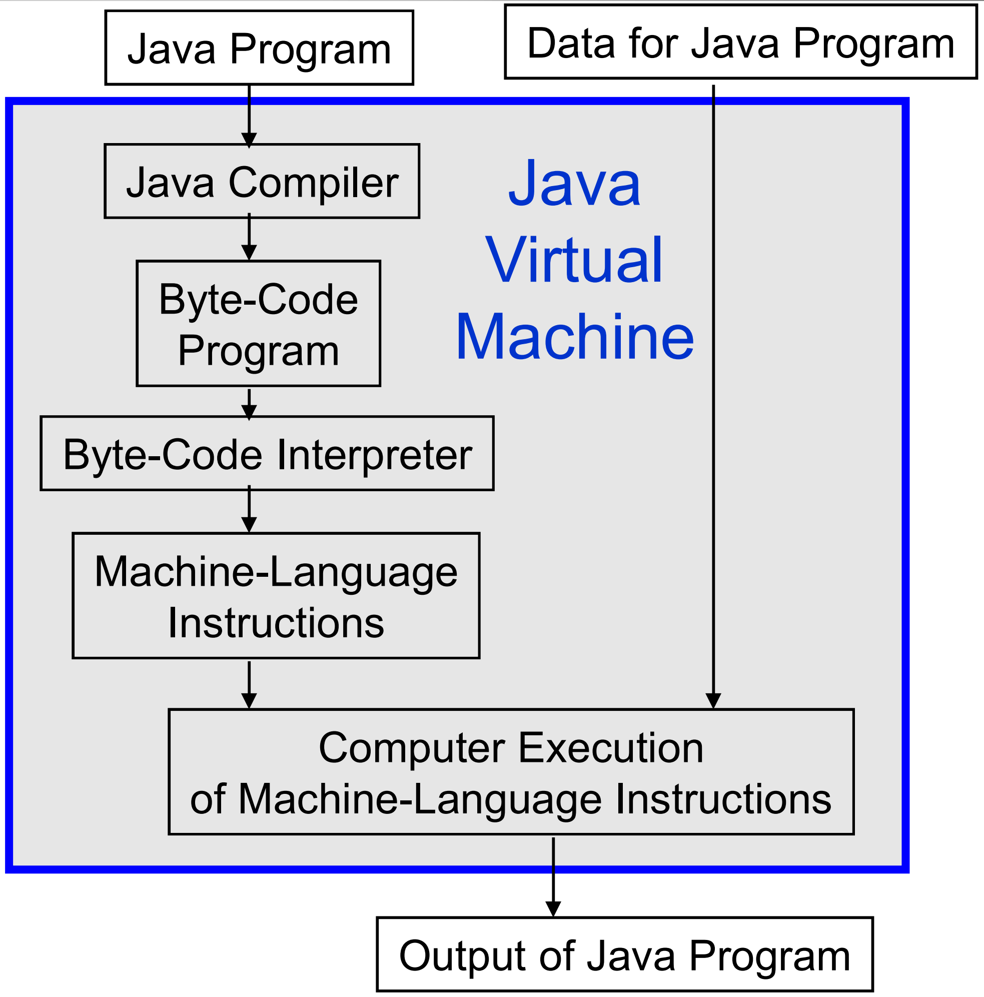


Disadvantage of translation

- 1) need a for each type of computer and each operating system
- 2) if a manufacturer comes out with a new type of computer, a team of programmers must write for that computer.

Java Program Translation

- Both Compilation and Interpretation
- Intermediate Code: [redacted]
  - » similar to assembly code, but hardware [redacted]
- Interpreter translates from generic [redacted] code to hardware-specific [redacted] code



Questions ??

- Java Virtual Machine?
- Virtual ?
- Virtual Machine ?
- Java Byte Code ?
  - » machine language
    for a [redacted] computer
    that is something like
    the [redacted] of all computers

Java Byte Code

- generated by [redacted]
  - » Instead of generating machine language as most compilers do, the Java compiler generates byte code.
- easily [redacted] machine language of various kinds of computers
- executed by [redacted]
- [redacted] to programmer
  - » You don't have to know anything about how byte code works to write a Java program.

Why Use Byte Code?

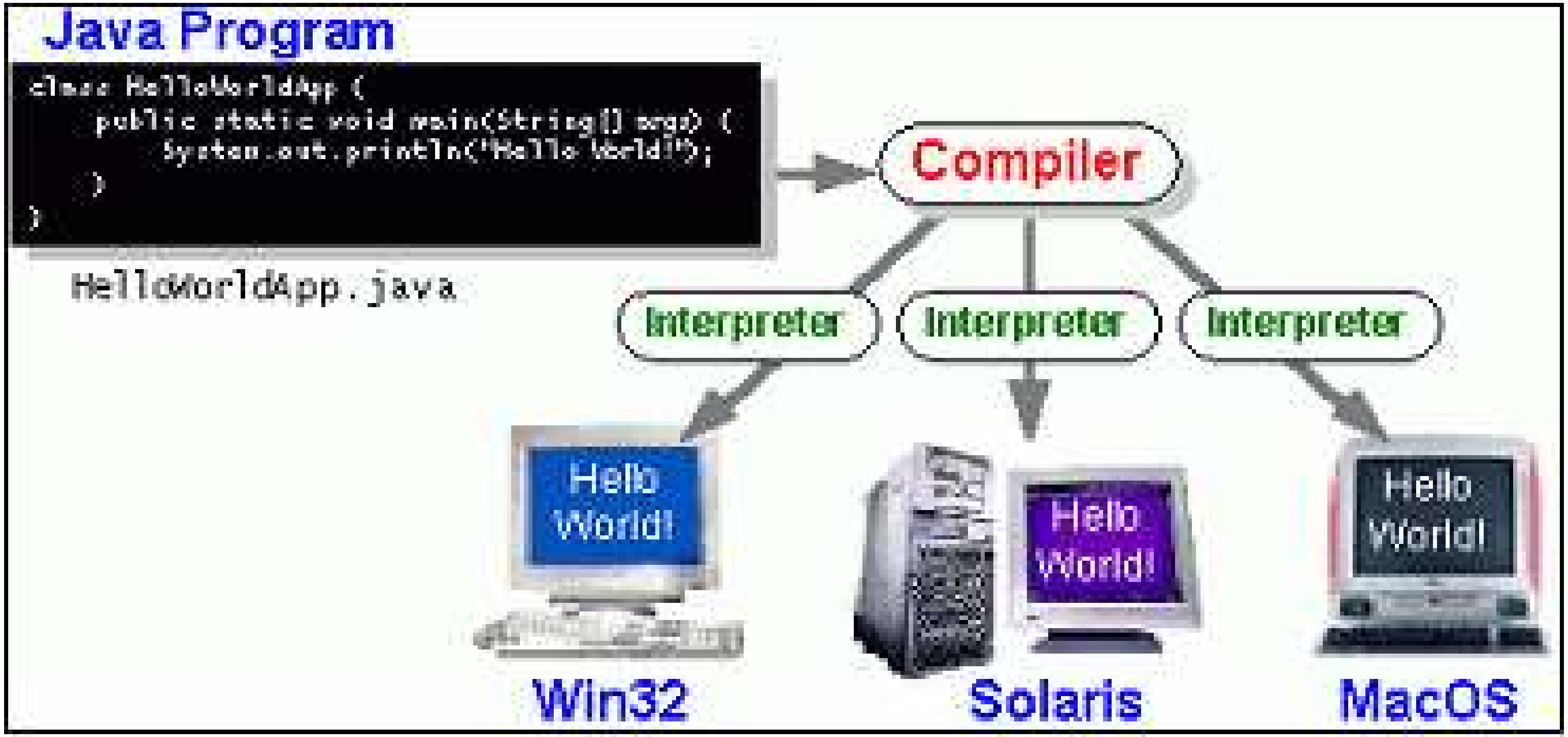
Disadvantages:

- requires both [redacted]
- [redacted] program execution

Advantages:

- [redacted]
  - » very important
  - » same program can run on computers of different types (useful with the Internet)
  - » Java compiler for new types of computers can be made quickly ➔ Java is good for internet applications.

portability



```
class HelloWorldApp {
    public static void main(String[] args) {
        System.out.println("Hello World!");
    }
}
```

HelloWorldApp.java

portability

- When new type of computer
  - » do not have to design a new java [redacted]
  - » every type of computer must have its own byte-code [redacted]
  - » but these interpreter are [redacted] programs compared to a compiler

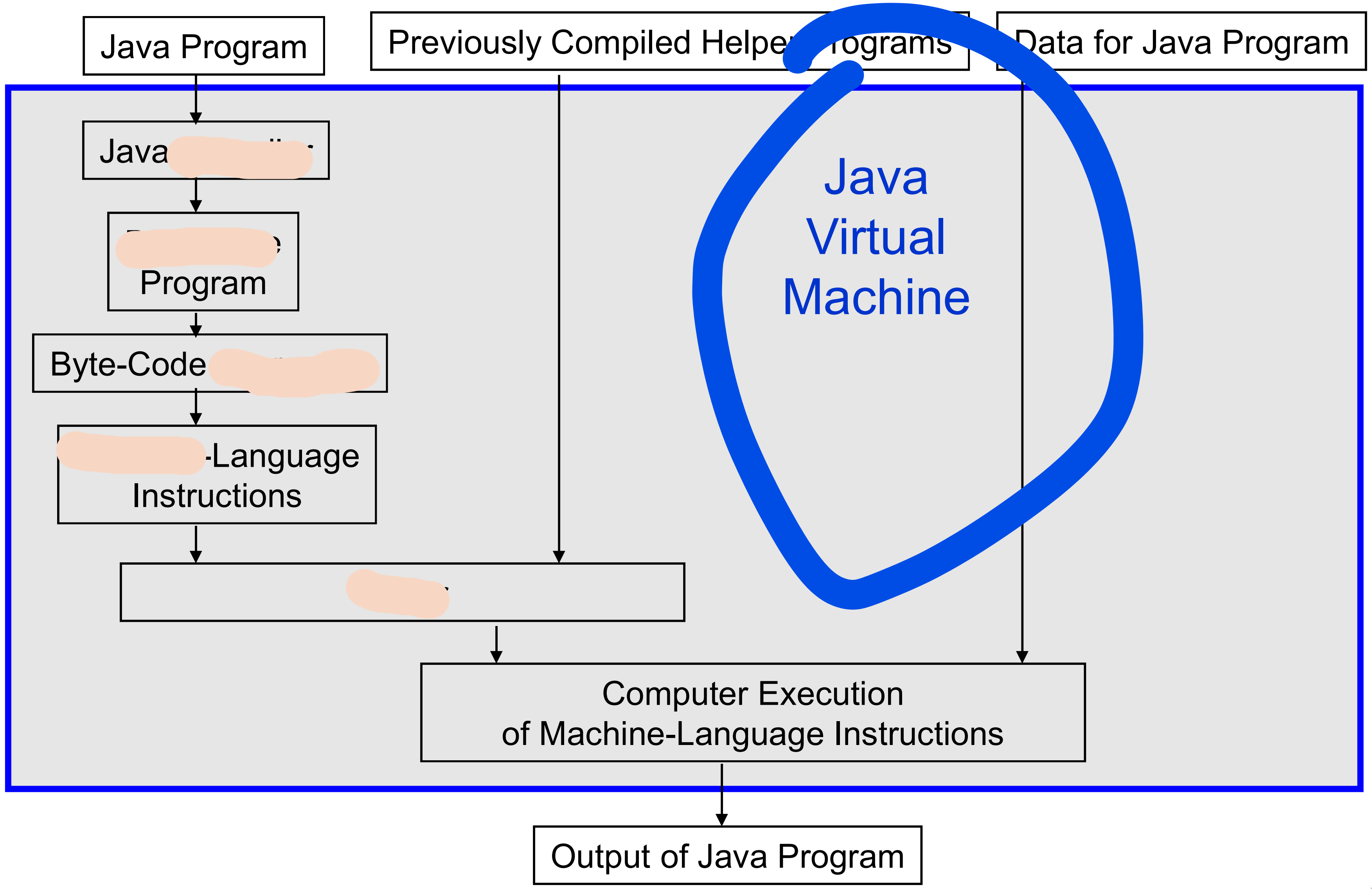
- new type of computer
  - » originally a language for programming home appliances
  - » ➔ Web Browser

Class Loader

- [redacted]
  - » the process of connecting them
  - » need connect several pieces of program

  - » ➔ Class loader
  - » ➔ Linker

Java Program Translation Including Linker



Questions ??

- [redacted] linking ??
  - » 실행 class가 [redacted] 중에 필요한 외부 class와 결합(linking)되도록 하는 방식
- 장점
  - » 여러 클래스들이 한 프로그램을 구성하는 경우에, 한 클래스를 수정해야 할 일이 발생할 경우, [redacted] 필요가 없이, 단지 변경된 클래스가 속한 파일만을 컴파일함

끝