CalPolyPomona

Introduction to Computers and Java

Lecture 1a

Topics

- Why Program?
- Computer Systems: Hardware and Software
- Programming Languages
- Object-Oriented Programming
- Software Engineering
- Java History
- What Is a Program Made Of?
- The Programming Process

Why Program? (1 of 4)

- Computers are tools that can be programmed to perform many functions, such as:
  - spreadsheets
  - databases
  - word processing
  - games
  - etc.
- They are machines specifically designed to follow instructions, doing whatever job their programs, or **software**, tell them to do.
- Computer Programmers implement **programs** that perform these functions.

Why Program? (2 of 4)

- Because of the computer's programmability, it doesn't belong to any single profession. It is perhaps the most versatile tool ever made

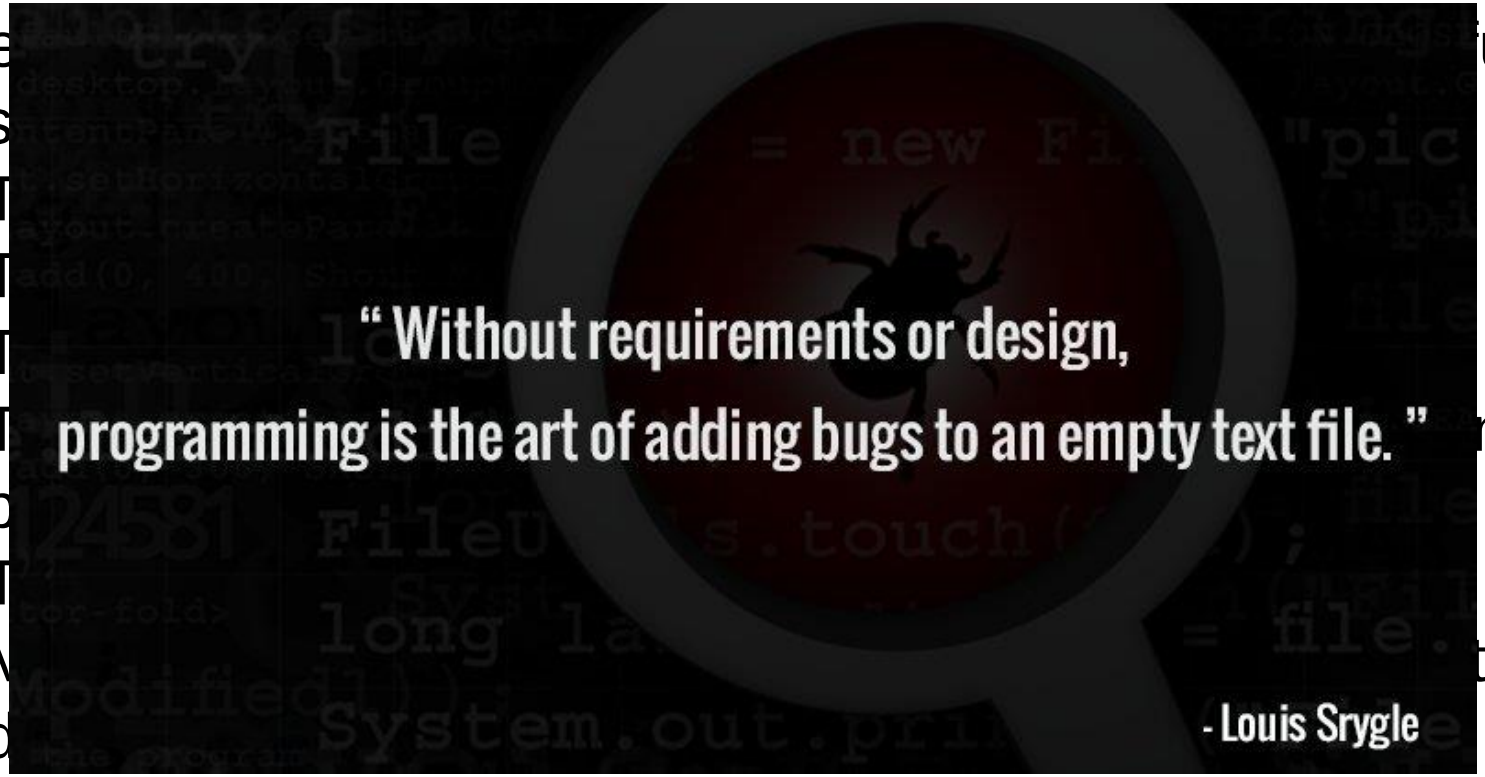
- Computer Programming is both and **art** and a **science!**

Why Program? (3 of 4)

- Point to ponder #1

  Why art?

  Eve ully
  des
  - T
  - T
  - T
  - T n is
    p
  - T
  - M tten
    d



Why Program? (4 of 4)

- Point to ponder #2

  Why science?

  - Programs must be analytically correct.
  - Programs rarely work the first time they are programmed.
  - Programmers must perform the following on a continual basis: analyze, experiment, correct, and redesign.
  - Programming languages have strict rules, known as *syntax*, that must be carefully followed.

Computer Systems: Hardware and Software

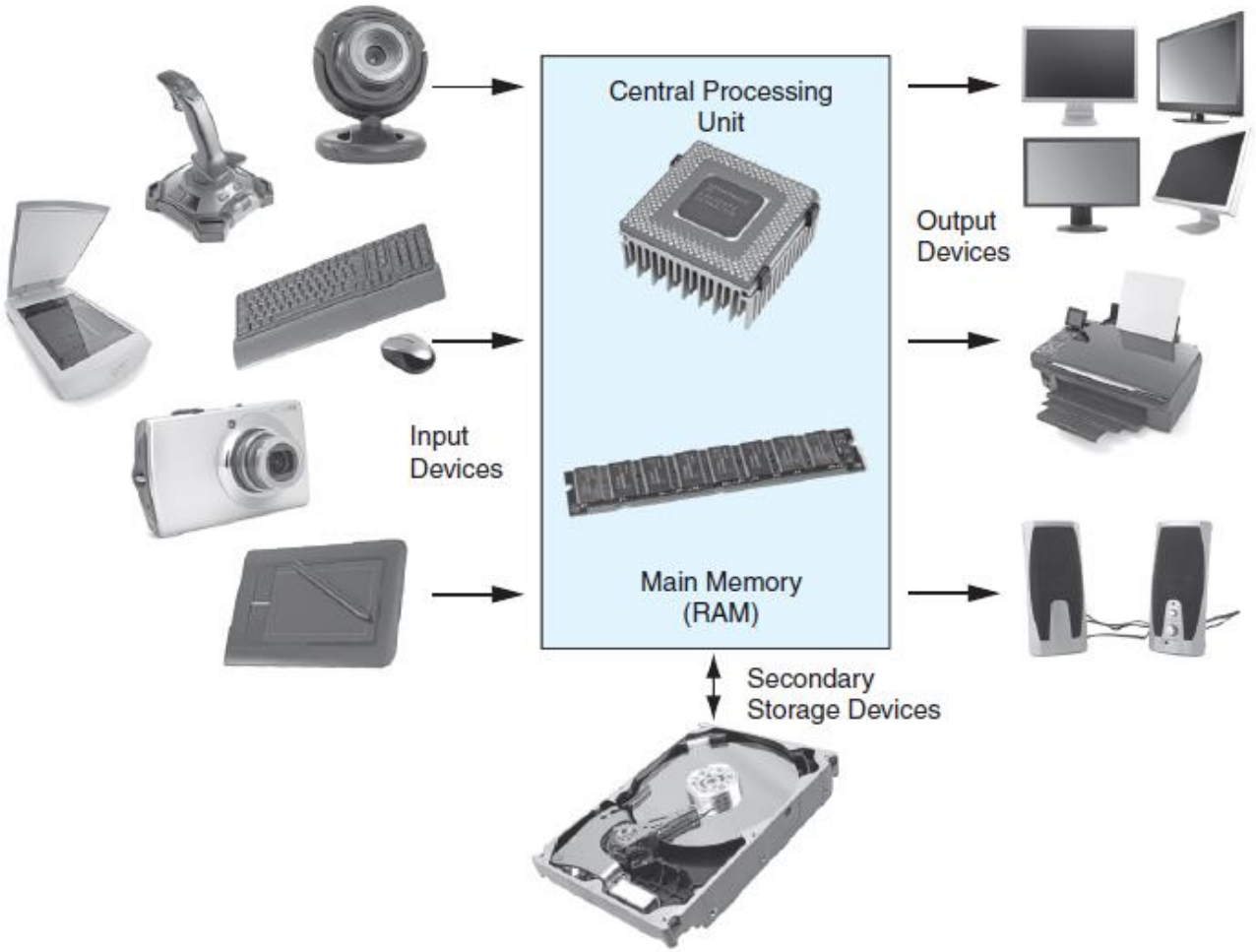
"Those parts of the system that you can hit with a hammer are called **hardware**; those program instructions that you can only curse at are called **software**".

-- Levitating Trains and Kamikaze Genes Technological Literacy for the 1990's

Computer Systems: Hardware (1 of 2)

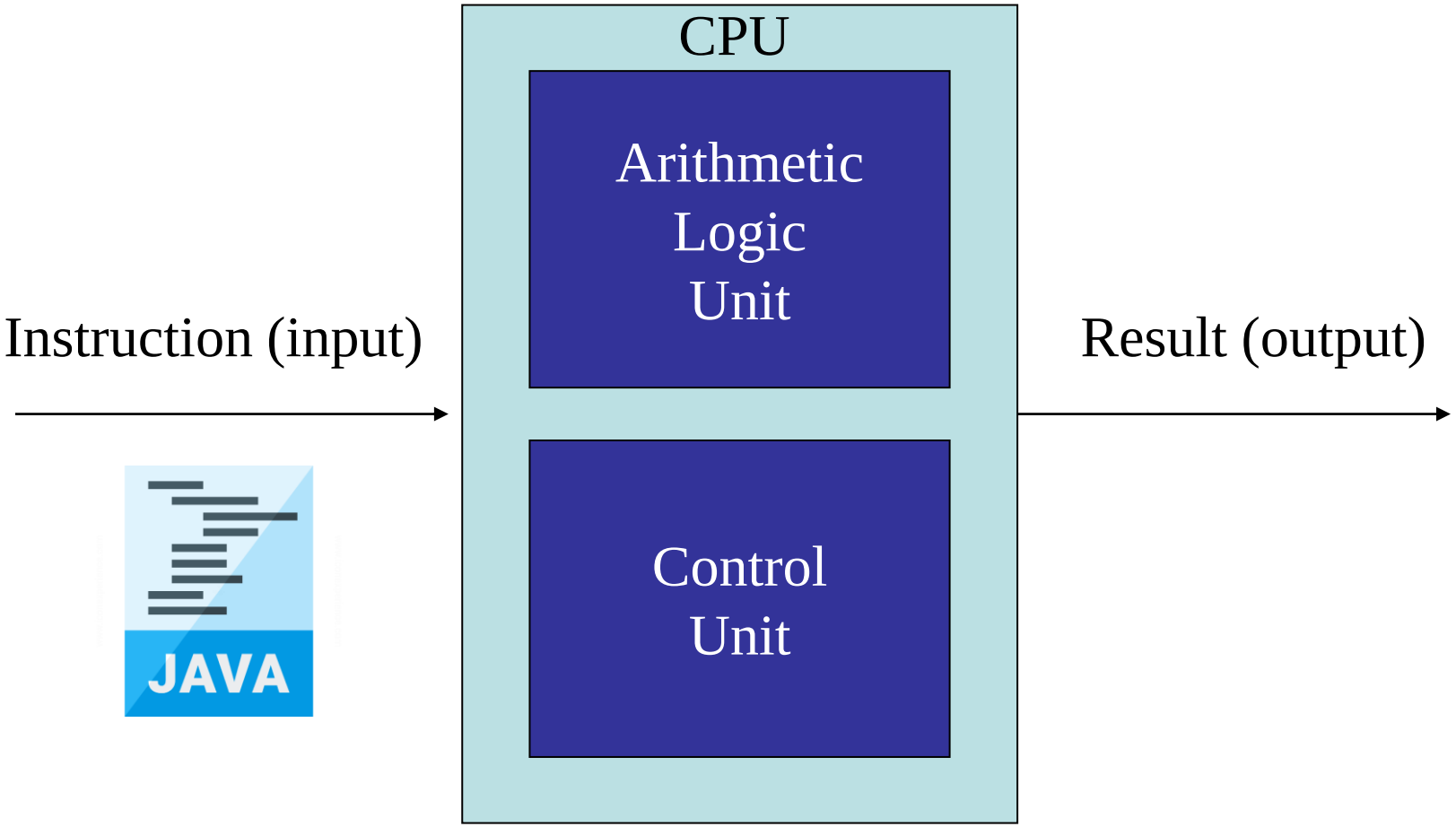
- Computer hardware components are the physical pieces of the computer.
- The major hardware components of a computer are:
  - The central processing unit (CPU)
  - Main memory
  - Secondary storage devices
  - Input and Output devices

Computer Systems: Hardware (2 of 2)



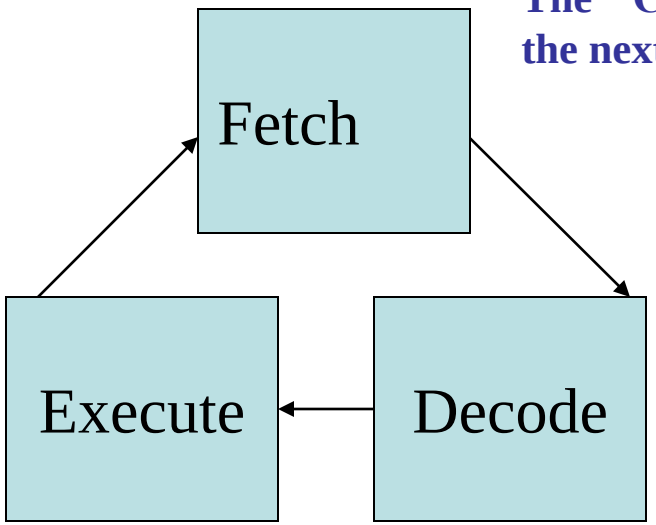
Computer Systems: Hardware

Central Processing Unit (1 of 2)



Computer Systems: Hardware
Central Processing Unit (2 of 2)

– To process program information, the CPU performs the fetch, decode, execute cycle.



The CPU's control unit fetches, from main memory, the next instruction in the sequence of program instructions.

The instruction is encoded in the form of a number. The control unit decodes the instruction and generates an electronic signal.

The signal is routed to the appropriate component of the computer (such as the ALU, a disk drive, or some other device). The signal causes the component to perform an operation.

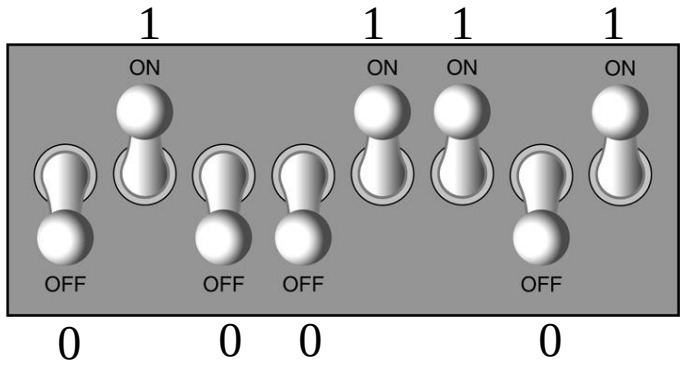
Computer Systems: Hardware

Main Memory (1 of 5)

- Commonly known as *random-access memory (RAM)*
- RAM contains:
  - Sequence of instructions of the **running** programs
  - data used by those programs.
- RAM is divided into units called *bytes*.
- A byte consists of eight *bits*.

Computer Systems: Hardware

Main Memory (2 of 5)



- A bit is either on or off:
  - 1 = on
  - 0 = off
- The bits form a pattern that represents a character or a number.
- Each byte in memory is assigned a unique number known as an *address*.
- RAM is *volatile*, which means that when the computer is turned off, the contents of RAM are erased.

Computer Systems: Hardware

Main Memory (3 of 5)

0	1	2	3	4	5	6	7	8	9
10	11	12	13	14	15	16 149	17	18	19
20	21	22	23 72	24	25	26	27	28	29

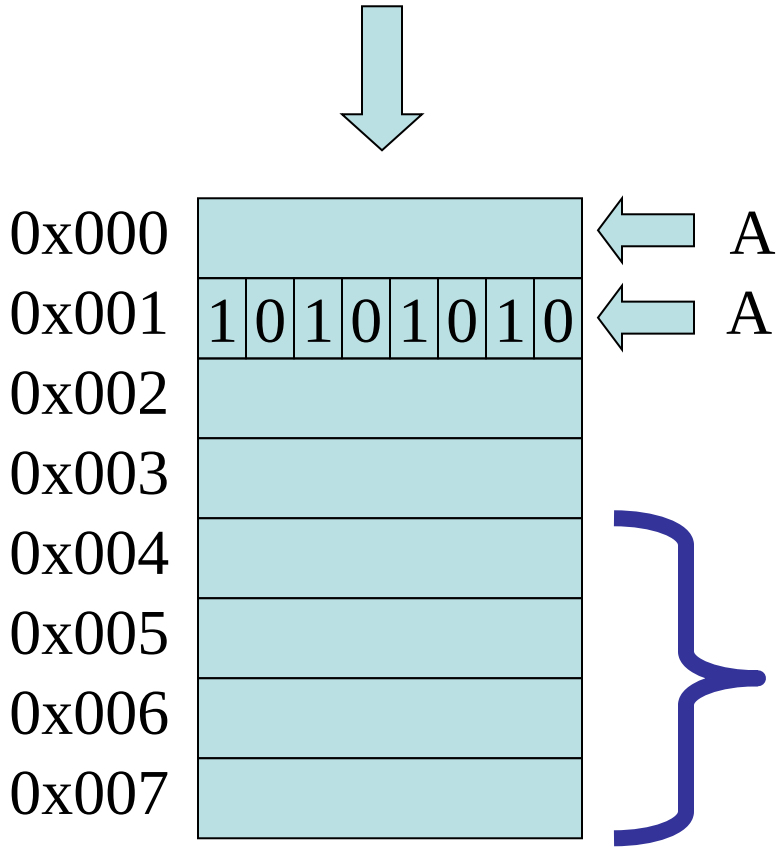
Series of bytes with their addresses. In the illustration, sample data is stored in memory. The number 149 is stored in the byte at address 16, and the number 72 is stored in the byte at address 23.

A byte is identified by its address in much the same way a post office box is identified by an address.

Computer Systems: Hardware

Main Memory (4 of 5)

Main memory is typically visualized as a column or row of cells.



A section of memory is called a *byte*.

A byte is made up of 8 *bits*.

A section of two or four bytes is often called a *word*.

Computer Systems: Hardware
Main Memory (5 of 5)

- Point to ponder #3

  Why does the definition of bit sound like that of atom?

- One is the **smallest unit** of measurement used to quantify computer data while the other is the **smallest unit** of ordinary matter that forms a chemical element.

Computer Systems: Hardware

Secondary Storage Devices

- Secondary storage devices can store information for longer periods of time (*non-volatile*).
- Common Secondary Storage devices:
  - Disk drive
  - External drive
  - CD drive
  - Solid state drive (SSDs)
  - USB drive (Flash drive)
  - DVD drive

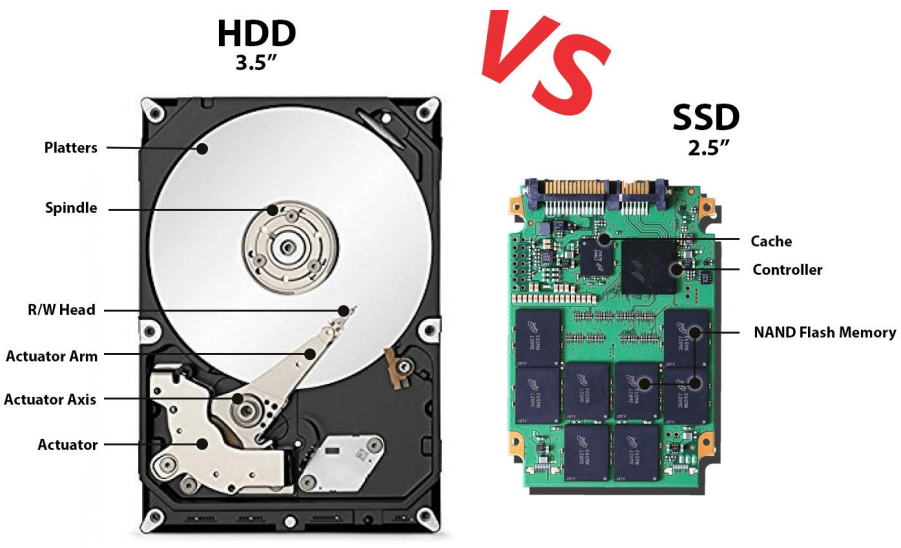
Computer Systems: Hardware

Secondary Storage Devices

- Point to ponder #4

  Why SSDs are faster than HDDs?

HDD are essentially metal platters with a magnetic coating that store data. A read/write head on an arm accesses the data while the platters are spinning. On the other hand, SSDs store and retrieve data using only flash-memory chips, **without any involvement of moving mechanical parts**.



Computer Systems: Hardware
Input Devices

- Input is any data the computer collects from the outside world.
- That data comes from devices known as *input devices*.
- Common input devices:
  - Keyboard
  - Mouse
  - Scanner
  - Digital camera

Computer Systems: Hardware
Output Devices

- Output is any data the computer sends to the outside world.
- That data is displayed on devices known as *output devices*.
- Common output devices:
  - Monitors
  - Printers
- Some devices such as disk drives perform input and output and are called *I/O devices* (input/output).

Computer Systems: Software

- Software refers to the programs that run on a computer.
- There are two classifications of software:
  - Operating Systems
  - Application Software

Computer Systems: Software

Operating Systems

- An operating system is a set of programs that manages the computer's hardware devices and controls their processes.
- Most all modern operating systems are multitasking:
  - UNIX
  - Linux
  - Mac OS
  - Windows

Computer Systems: Software
Operating Systems

- Point to ponder #5

  But what exactly means to be a multitasking operating system?

- It does not require parallel execution, but it does require that more than one task can be part-way through execution at the same time and that more than one task is advancing over a given period (time sharing).

Computer Systems: Software

Application Software

- *Application software* refers to programs that make the computer useful to the user.
- Application software provides a more specialized type of environment for the user to work in.
- Common application software:
  - Spreadsheets
  - Word processors
  - Accounting software
  - Tax software
  - Games

Computer Systems: Software

Application Software

- Point to ponder #6

  Which type of software are we supposed to build during this course?

  - Operating Systems?
  - Application Software?

Java Applications!

Programming Languages (1 of 8)

- A programming language is a special language used to write computer programs.
- A computer program is a set of instructions that enable the computer to solve a problem or perform a task.
- Collectively, these instructions form an *algorithm*
- An algorithm is a set of well-defined sequential steps to completing a task.

Programming Languages (2 of 8)

- Point to ponder #7

  So, algorithms and programs are the same thing?

  - Algorithm of linear search:

```
1. Start from the leftmost element of arr[] and
one by one compare x with each element of arr[].
2. If x matches with an element, return the index.
3. If x doesn’t match with any of elements, return -1.
```

  - Program of linear search:

```
int search(int[] arr, int x)
{
    int i;
    for (i = 0; i < arr.length; i++)
        if (arr[i] == x)
         return i;
    return -1;
}
```

Programming Languages (3 of 8)

- An algorithm is a sequence of steps that describes an idea for solving a problem meeting the criteria of correctness and terminability

- A program is a sequence of instructions written to run on a machine. It simply has to compile (or be interpreted)! It does not need to solve a problem.

  This is a valid python program:

```
1 while True:
2         print "dahsdkasd"
```

- Programs usually implement algorithms, going from an idea to a concrete thing. Often you need to fill in gaps that the algorithm does not specify but is intuitively understandable to humans.

Programming Languages (4 of 8)

- A computer needs the algorithm to be written in *machine language*.
- Machine language is written using *binary numbers (numbers consisting of only 1s and 0s)*.
- The binary numbers form machine language instructions, which the CPU interprets as commands, e.g., 1011010000000101.
- Each CPU has its own machine language.
  - Motorola 68000 series processors
  - Intel x86 series processors
  - ARM processors, etc.

Programming Languages (6 of 8)

- In the distant past, programmers wrote programs in machine language.
- Programmers developed higher level programming languages, which use words instead of numbers, to make things easier.
- Those programming languages translate the understandable words into machine language.
- The first programming language was *Assembly (or Assembler)*.
- Assembler made things easier but is also processor dependent.

Programming Languages (7 of 8)

- High level programming languages followed that were not processor dependent.
- Some common programming languages:

Java	C	Visual Basic
BASIC	C++	Python
COBOL	C#	Ruby
Pascal	PHP	JavaScript

Programming Languages (8 of 8)

```java
public class Adder {
  public static void main(String[] args) {
    for (int i = 0; i < 80000;i++) {
      additions();
    }
  }

  private static void additions() {
    int a = 1;
    int b = 2;
    int c = 3;
    a = b + c;
    b = a + c;
    c = a + b;
  }
}
```

Java

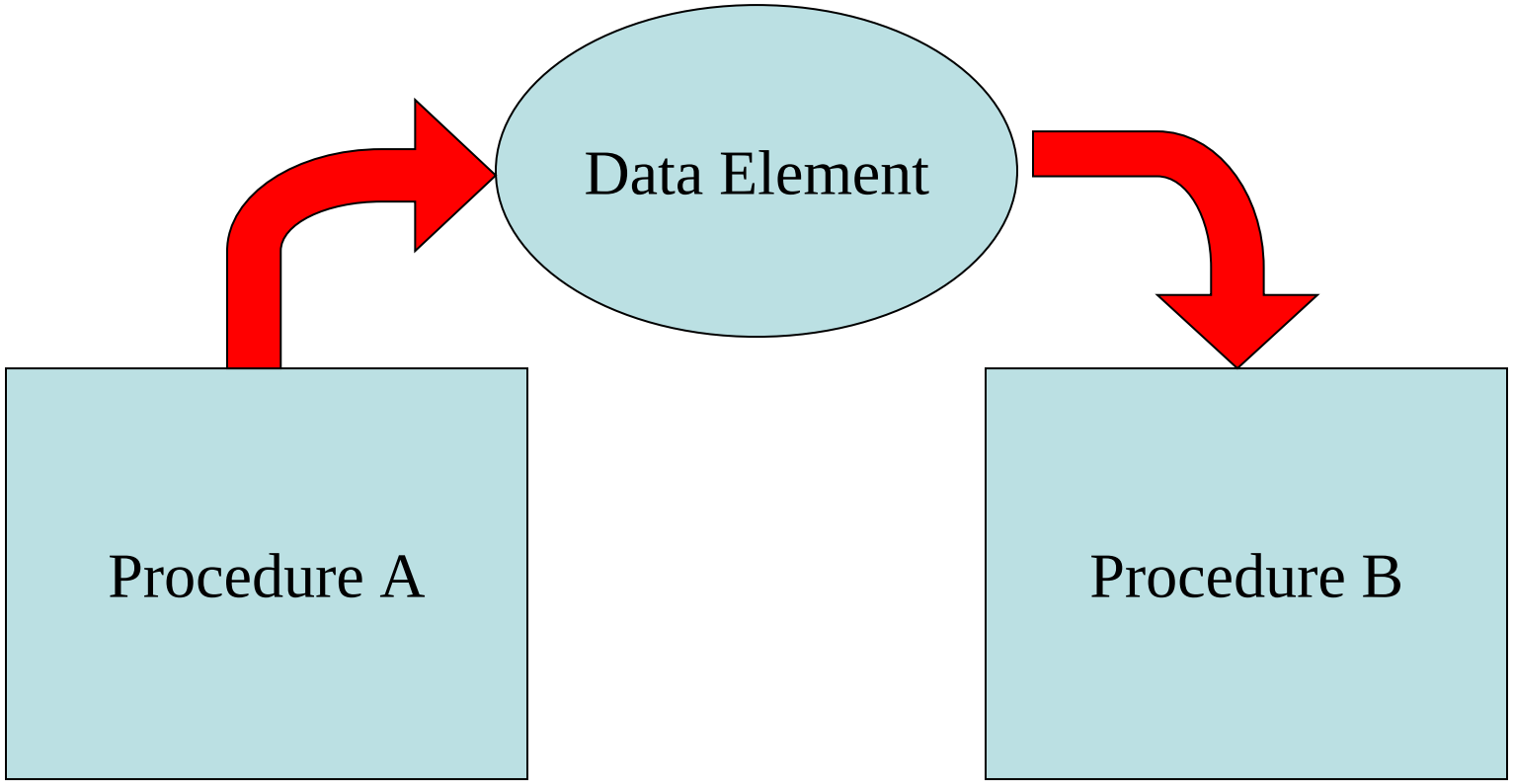
```
  # {method} 'additions' '()V' in 'Adder'
  #           [sp+0x20]  (sp of caller)
  0x00000000026e2480: sub     rsp,18h
  0x00000000026e2487: mov     qword ptr [rsp+10h],rbp  ;*synchronizatio
                                                       ; - Adder::additions@-1
  0x00000000026e248c: add     rsp,10h
  0x00000000026e2490: pop     rbp
  0x00000000026e2491: test    dword ptr [120000h],eax  ;   {poll_return
  0x00000000026e2497: ret
  0x00000000026e2498: hlt
  0x00000000026e2499: hlt
  0x00000000026e249a: hlt
  0x00000000026e249b: hlt
  0x00000000026e249c: hlt
  0x00000000026e249d: hlt
  0x00000000026e249e: hlt
  0x00000000026e249f: hlt
[Exception Handler]
[Stub Code]
  0x00000000026e24a0: jmp     26df220h          ;   {no_reloc}
[Deopt Handler Code]
  0x00000000026e24a5: call    26e24aah
  0x00000000026e24aa: sub     qword ptr [rsp],5h
  0x00000000026e24af: jmp     26b9000h          ;   {runtime_call}
  0x00000000026e24b4: hlt
  0x00000000026e24b5: hlt
  0x00000000026e24b6: hlt
  0x00000000026e24b7: hlt
```

Assembly

Procedural Programming (1 of 3)

- Older programming languages were procedural.
- A *procedure* is a set of programming language statements that, together, perform a specific task.
- Procedures typically operate on data items that are separate from the procedures.
- In a procedural program, the data items are commonly passed from one procedure to another.

Procedural Programming (2 of 3)



Procedural Programming (3 of 3)

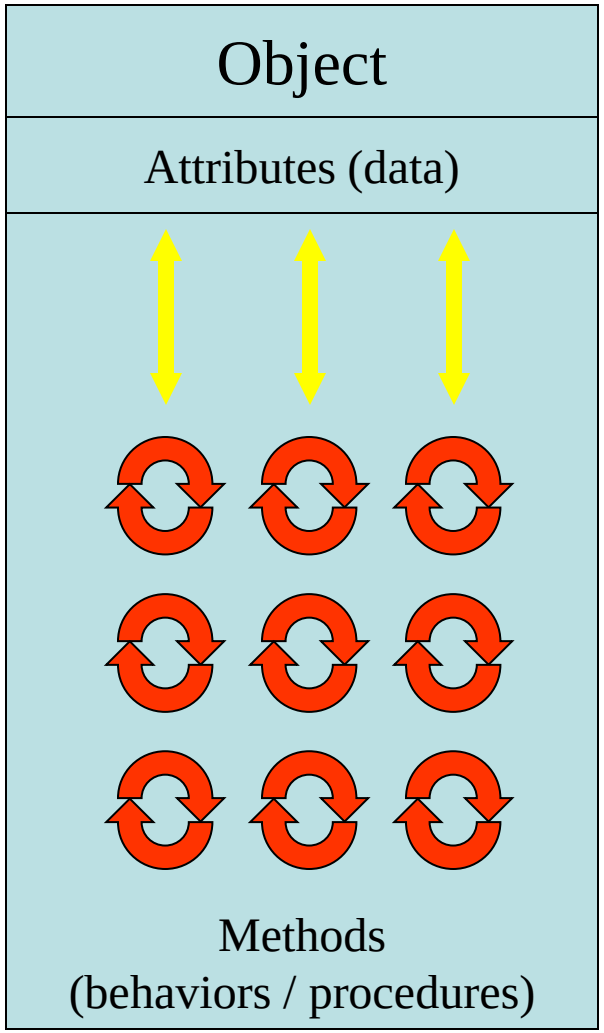
- In procedural programming, procedures are developed to operate on the program's data.
- Data in the program tends to be global to the entire program.
- Data formats might change and thus, the procedures that operate on that data must change.

Object-Oriented Programming (1 of 5)

- Object-oriented programming is centered on creating objects rather than procedures.
- Objects are a melding of data and procedures that manipulate that data.
- Data in an object are known as *attributes*.
- Procedures in an object are known as *methods*.

Object-Oriented Programming (2 of 5)

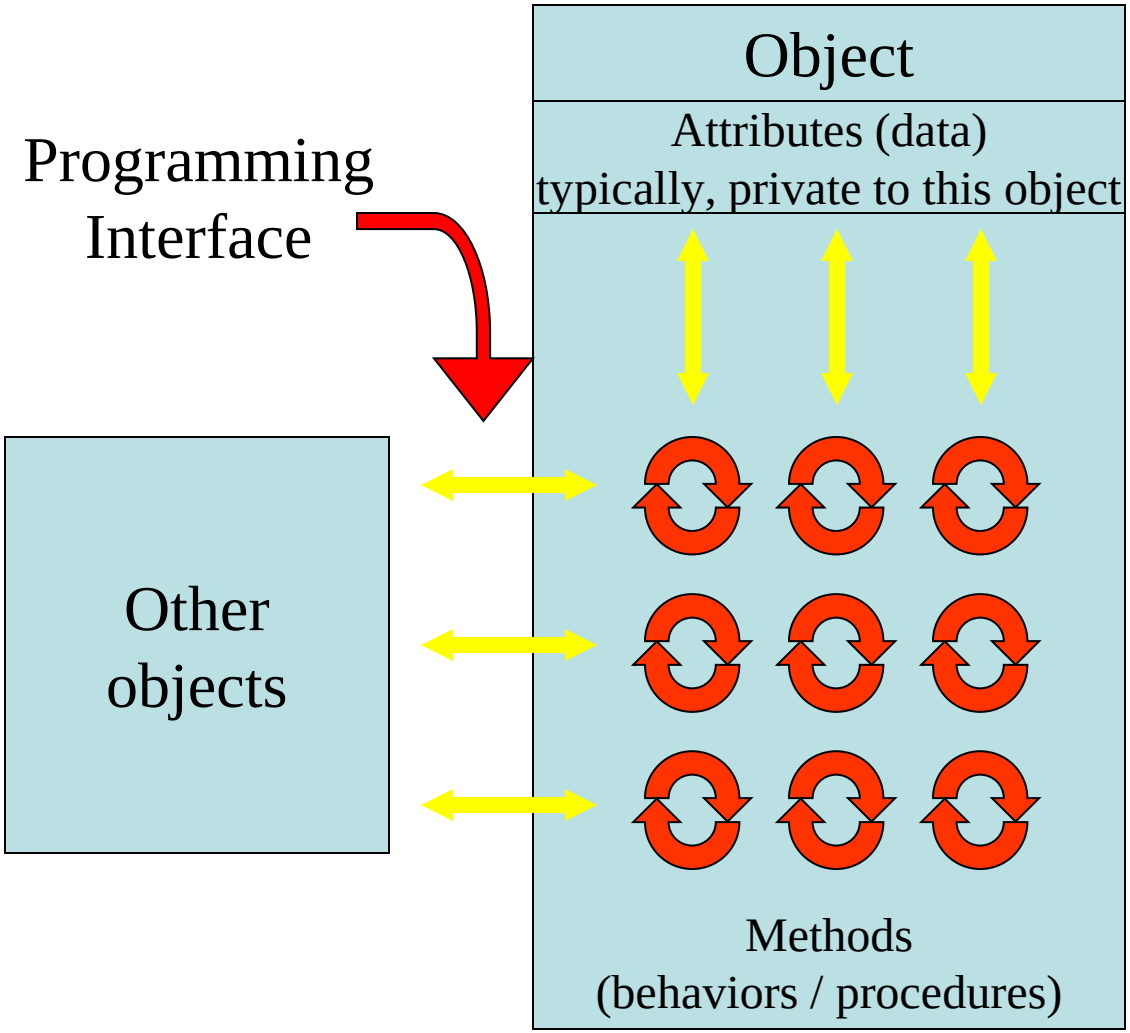
Self-contained unit consisting of attributes and methods



Object-Oriented Programming (3 of 5)

- Object-oriented programming combines data and behavior via *encapsulation*.
- *Data hiding* is the ability of an object to hide data from other objects in the program.
- Only an object's methods should be able to directly manipulate its attributes.
- Other objects are allowed manipulate an object's attributes via the object's methods.
- This indirect access is known as a *programming interface*.

Object-Oriented Programming (4 of 5)



Object-Oriented Programming (5 of 5)

- Java is an object-oriented programming (OOP) language!

Software Engineering (1 of 3)

- Encompasses the whole process of crafting computer software.
- Software engineers perform several tasks in the development of complex software projects.
  - designing,
  - writing,
  - testing,
  - debugging,
  - documenting,
  - modifying, and
  - maintaining.

Software Engineering (2 of 3)

- Software engineers develop:
  - program specifications,
  - diagrams of screen output,
  - diagrams representing the program components and the flow of data,
  - pseudocode,
  - examples of expected input and desired output.

Software Engineering (3 of 3)

- Software engineers also use special software designed for testing programs.
- Most commercial software applications are large and complex.
- Usually a team of programmers, not a single individual, develops them.
- Program requirements are thoroughly analyzed and divided into subtasks that are handled by
  - individual teams
  - individuals within a team.

CalPolyPomona

Introduction to Computers and Java

Lecture 1b

Topics

- Why Program?
- Computer Systems: Hardware and Software
- Programming Languages
- Object-Oriented Programming
- Software Engineering
- Java History
- What Is a Program Made Of?
- The Programming Process

Java History (1 of 8)

- 1991 - Green Team was formed at Sun Microsystems (now owned by Oracle) to speculate about important technological trends for the future.
- The team's conclusion: computers would merge with consumer appliances.

Java History (2 of 8)

- Point to ponder #1

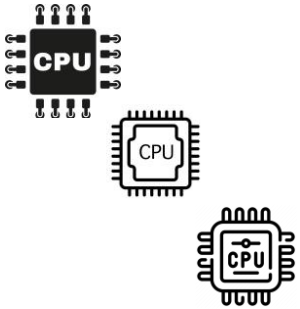
Was this team right?

Yes!

Java History (3 of 8)

- There was a need for a programming language that would run on various consumer devices, with different processors.
- Point to ponder #2

What is the problem of having different processors?



0010011110011101 0

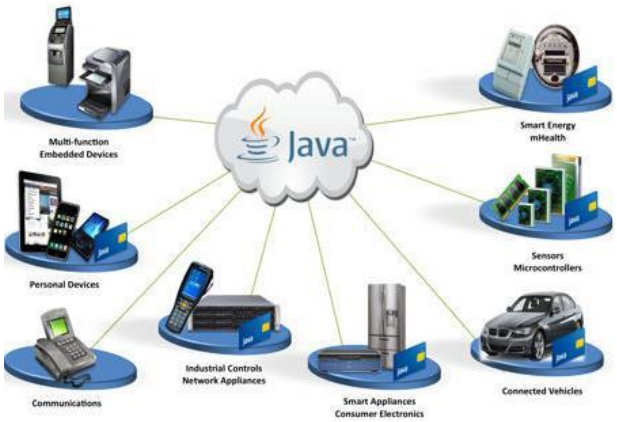
1100110101000101 1

1101101000000010 1

Each has its own machine language ...

Java History (4 of 8)

- Java (first named Oak) was developed for this purpose.



- Programs written in Java were not translated into the machine language of a processor, but into an intermediate language known as *byte code*.
- Then, another program would translate the byte code into machine language that could be executed by the processor of a consumer device.

Java History (5 of 8)

- Unfortunately, the technology developed by the Green Team was ahead of its time.

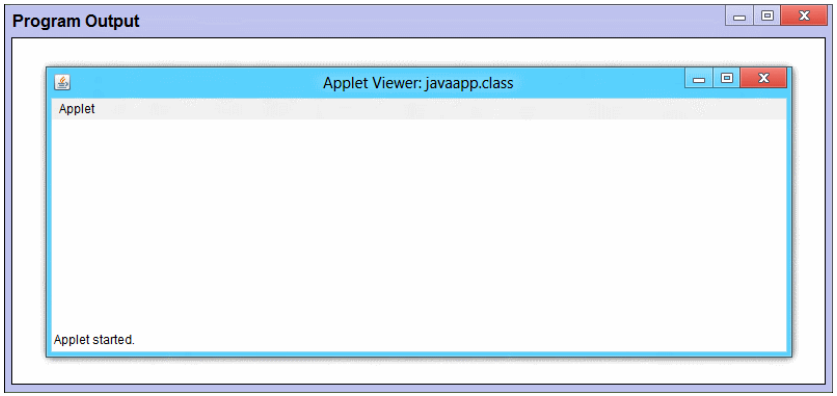
- Point to ponder #3

  Why?

The computer-controlled consumer appliance industry was just beginning at that time.

Java History (6 of 8)

- Let's wait then … Meanwhile, Java enabled the web browser (*HotJava*), demonstrating it at the 1995 Sun World conference. This browser was able to run small programs known as Java Applets. Animations! User interactions!



- Java was incorporated into Netscape shortly after.
- Java is "cross platform", meaning that it can run on various computer operating systems.

Java History (7 of 8)

- Point to ponder #4

What was the slogan created by Sun Microsystems to illustrate the cross-platform benefits of the Java language?

Java History (8 of 8)

- Java applications can be of multiple types:
  - Desktop applications.
  - Web applications
  - Mobile applications
- To create:
  - Stand-alone applications
  - Client-server applications in multiple layers

What Is a Program Made Of?
Common Language Elements

- There are some concepts that are common to virtually all programming languages.
- Common concepts:
  - Key words (reserved)
  - Operators
  - Punctuation
  - Programmer-defined names (identifiers)
  - Syntax (strict syntactic rules)

What Is a Program Made Of?

Sample Program

```
public class Payroll
{
  public static void main(String[] args)
  {
      int hours = 40;
      double grossPay, payRate = 25.0;

      grossPay = hours * payRate;
      System.out.println("Your gross pay is $" + grossPay);
  }
}
```

What Is a Program Made Of?

Key words in the sample program (1 of 2)

```
public class Payroll
{
  public static void main(String[] args)
  {
      int hours = 40;
      double grossPay, payRate = 25.0;

      grossPay = hours * payRate;
      System.out.println("Your gross pay is $" + grossPay);
  }
}
```

What Is a Program Made Of?

Key words in the sample program (2 of 2)

- Key words are lower case (Java is a case-sensitive language).
- Point to ponder #5

  What does it mean to be a case-sensitive language?

  `Public ≠ public`

- Key words cannot be used as a programmer-defined identifier.

What Is a Program Made Of?

Programmer-defined names in the sample program

```
public class Payroll
{
  public static void main(String[] args)
  {
      int hours = 40;
      double grossPay, payRate = 25.0;

      grossPay = hours * payRate;
      System.out.println("Your gross pay is $" + grossPay);
  }
}
```

What Is a Program Made Of?

Operators in the sample program

```
public class Payroll
{
  public static void main(String[] args)
  {
      int hours = 40;
      double grossPay, payRate = 25.0;

      grossPay = hours * payRate;
      System.out.println("Your gross pay is $" + grossPay);
  }
}
```

What Is a Program Made Of?

Punctuation in the sample program (1 of 2)

```
public class Payroll
{
  public static void main(String[] args)
  {
      int hours = 40;
      double grossPay, payRate = 25.0;

      grossPay = hours * payRate;
      System.out.println("Your gross pay is $" + grossPay);
  }
}
```

What Is a Program Made Of?

Punctuation in the sample program (2 of 2)

- Semi-colons are used to end Java statements, like a period in English; however, not all lines of a Java program end a statement.

```
public class Payroll {
}
```

- Part of learning Java is to learn where to properly use the punctuation.

What Is a Program Made Of?

Lines vs Statements (1 of 2)

- There are differences between lines and statements when discussing source code.
- A statement is a **complete Java instruction** that causes the computer to perform an action.
- A line is just that **a single line of code**.
- Blank lines are only used to make a program more readable.

What Is a Program Made Of?
Lines vs Statements (2 of 2)

- Point to ponder #6

  How many statements and lines do we have below?

```
System.out.println("Your gross   pay
is $" + grossPay);
```

  This is one Java statement written using two lines.

What Is a Program Made Of?

Variables (1 of 4)

- Data in a Java program is stored in memory (RAM).
- Variable names represent a location in memory.
- Variables are created by the programmer who assigns it a programmer-defined identifier.

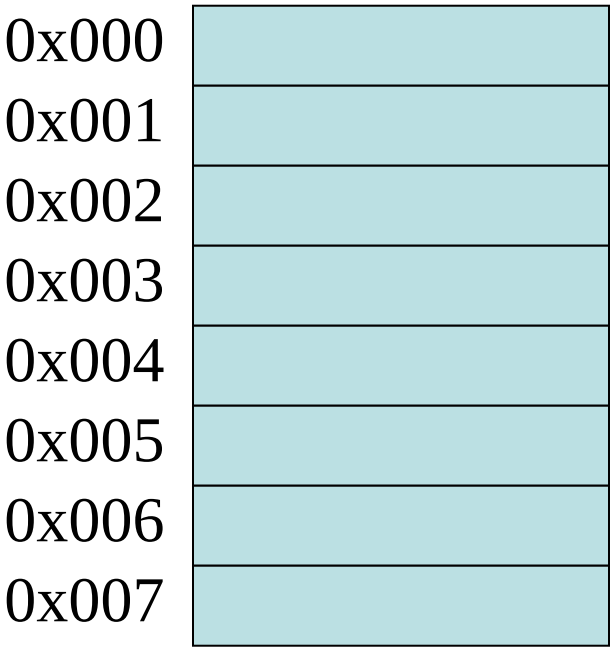
example: `int hours = 72;`

- In this example, the variable *hours* is created as an integer (more on this later) and assigned the value of 72.

What Is a Program Made Of?

Variables (2 of 4)

- Variables are simply a name given to represent a place in memory.



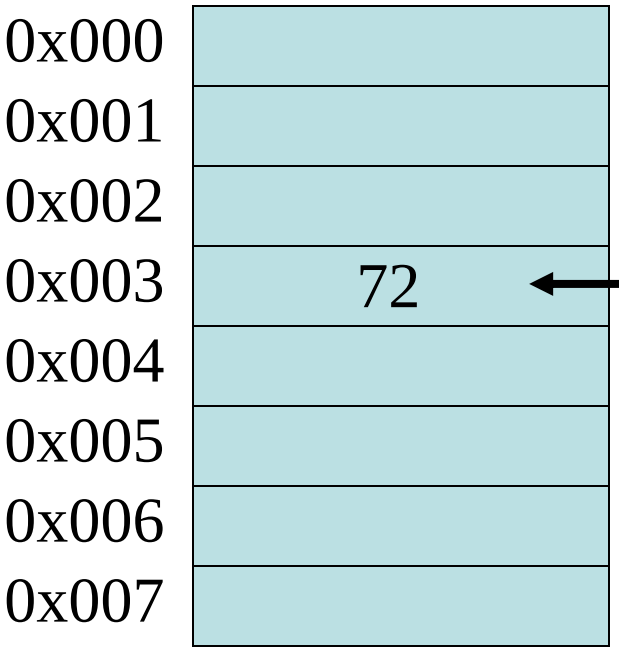
What Is a Program Made Of?

Variables (3 of 4)

Consider this variable declaration.

```
int length = 72;
```

The Java Virtual Machine (JVM) actually decides where the value will be placed in memory.



The variable *length* is a symbolic name for the memory location 0x003.

What Is a Program Made Of?
Variables (4 of 4)

- Point to ponder #7

  Why do we call those named storage locations as variables?

  Because data stored in a variable may change while the program is running.

What Is a Program Made Of?

The compiler and the Java Virtual Machine (JVM) (1 of 4)

- A programmer writes Java programming statements for a program.
- These statements are known as *source code*.
- A *text editor* is used to edit and save a Java *source code file.*
- Source code files have a *.java* file extension.
- A *compiler* is a program that translates source code into an executable form.

What Is a Program Made Of?

The compiler and the Java Virtual Machine (JVM) (2 of 4)

- A compiler is run using a source code file as input.
- Syntax errors that may be in the program will be discovered during compilation.
- *Syntax errors* are mistakes that the programmer has made that violate the rules of the programming language.
- The compiler creates another file that holds the translated instructions.

What Is a Program Made Of?

The compiler and the Java Virtual Machine (JVM) (3 of 4)

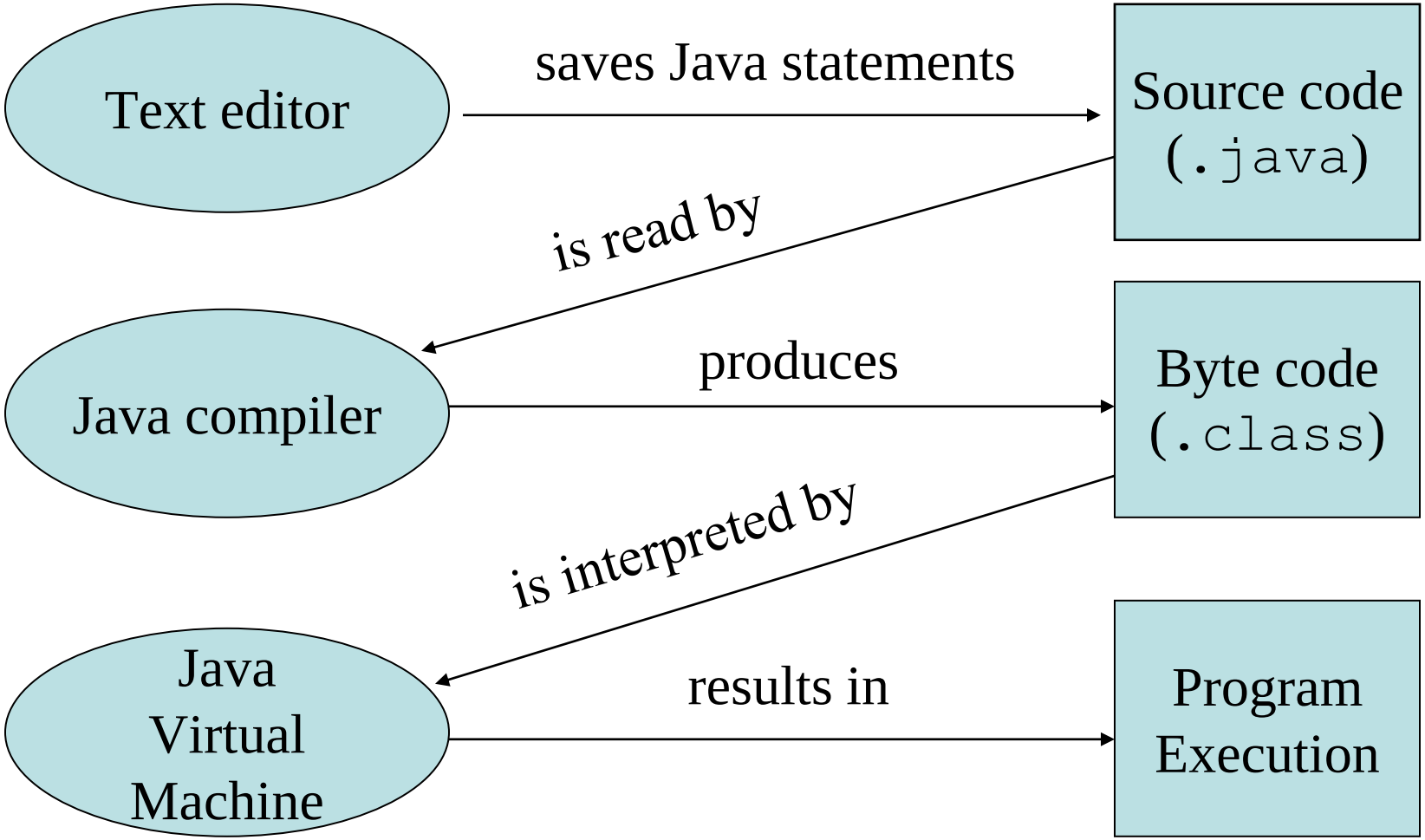
- Most compilers translate source code into *executable* files containing *machine code*.
- The Java compiler translates a Java source file into a file that contains *byte code* instructions.
- Byte code instructions are the machine language of the *Java Virtual Machine (JVM)* and cannot be directly executed directly by the CPU.

What Is a Program Made Of?

The compiler and the Java Virtual Machine (JVM) (4 of 4)

- Byte code files end with the *.class* file extension.
- The JVM is a program that *emulates* a micro-processor (simulating the CPU).
- The JVM executes instructions as they are read.
- JVM is often called an *interpreter.*
- Java is often referred to as an *interpreted language*.

Java Program Development Process (1 of 2)



Java Program Development Process (1 of 2)

- Point to ponder #8

  Is Java a compiled language?
  Is Java an interpreted language?
  Is Java a compiled and an interpreted language?

  Java is actually both, a compiled and an interpreted language!

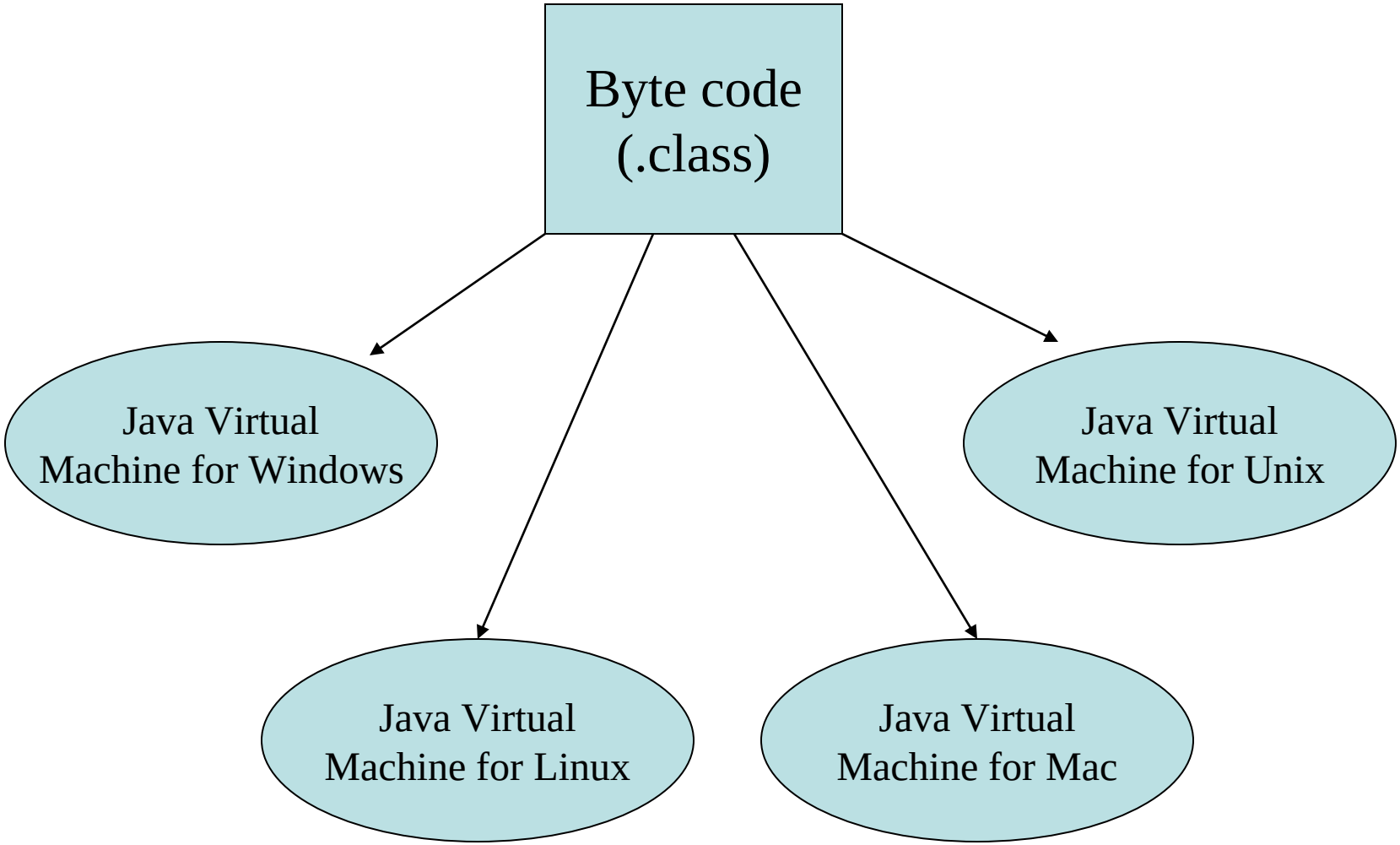
Portability (1 of 4)

- *Portable* means that a program may be written on one type of computer and then run on a wide variety of computers, with little or no modification.
- Java byte code runs on the JVM and not on any particular CPU; therefore, compiled Java programs are highly portable.
- JVMs exist on many platforms:
  - Windows
  - Mac
  - Linux
  - Unix
  - BSD
  - Etc.

Portability (2 of 4)

- With most programming languages, portability is achieved by compiling a program for each CPU it will run on.
- Java provides an JVM for each platform so that programmers do not have to recompile for different platforms.

Portability (3 of 4)



Portability (4 of 4)

- Point to ponder #9

  Why the name Java **Virtual Machine** (JVM) then?

- You don't program for the code to be executed on the physical machine; you write the code to be executed on the Java Virtual Machine.

Java Editions

- The software you use to write Java programs is called the Java Development Kit, or JDK.
- There are different editions of the JDK:
  - Java SE - Java *Standard Edition*.
  - Java EE - Java *Enterprise Edition*.
  - Java ME - Java *Micro Edition*.
- Available for download at http://java.oracle.com

Compiling a Java Program

- The Java compiler is a *command line* utility.
- The command to compile a program is:

  javac filename.java
- javac is the Java compiler.
- The `.java` file extension must be used.

  Example: To compile a java source code file named Payroll.java you would use the command:

  `javac Payroll.java`

Running a Java Program

- To run a Java program, you use the java command in the following.

  java *ClassFileName*

- ***ClassFileName*** is the name of the *.class* file that you wish to execute; however, you do not type the *.class* extension. For example, to run the program that is stored in the *Payroll.class* file, you would enter the following command:

  java Payroll

The Programming Process (1 of 3)

1. Clearly define what the program is to do.
2. Visualize the program running on the computer.
3. Use design tools to create a model of the program.
4. Check the model for logical errors.

The Programming Process (2 of 3)

5. Enter the code and compile it.
6. Correct any errors found during compilation.

   Repeat Steps 5 and 6 as many times as necessary.
7. Run the program with test data for input.
8. Correct any runtime errors found while running the program.

   Repeat Steps 5 through 8 as many times as necessary.
9. Validate the results of the program.

The Programming Process (3 of 3)

- Point to ponder #10

  What is the difference between a compile-time error, run-time error, and logical error?

  - compile-time error -> `x = 2:`
  - run-time error -> `x = 2/0;`
  - logical error -> `grossPay = hours + payrate;`

Activity …

Create, compile, and run the Java program below by using a basic text editor and the command prompt.

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