An Introduction to Software Engineering

David Greenstein
Monta Vista High School

Software Today

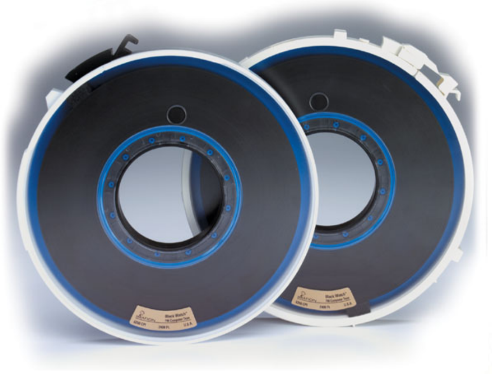
"This Internet parental controls software is kind of difficult to install. You might want your kids to do it for you."

Software Development

- **Pre-1970’s - Emphasis on efficiency**
    - Compact, fast algorithms on machines with limited memory
    - Required long learning curve, cryptic code
- **Today**
    - Emphasis on programmer productivity, team development, reusable code, maintainable code, portable code
    - Relatively user-friendly code

Evolution of Computer Architecture

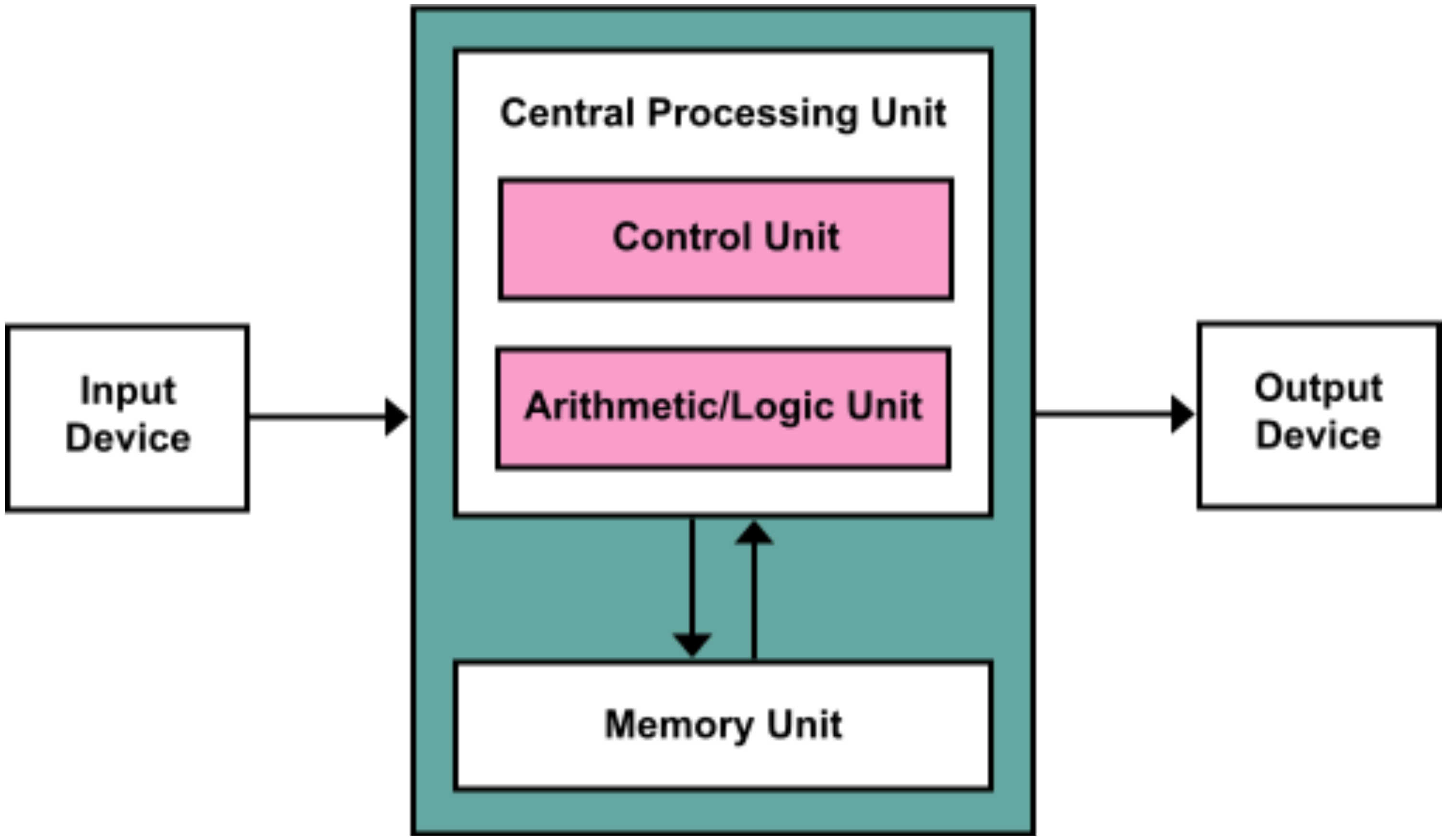
- **Pre-Programmable machines**
  - **Fixed machines** - the instructions were in the design
  - **Wired machines** - the instructions were in the wiring



- **Early stored program machines**
  - Program by loading cards, tape, etc.
- **Von Neumann architecture**
  - Modern computer architecture
  - Stores both the instructions and the data

- **Modified Harvard machine**
  - Special-purpose processors, like a GPU
  - Two sets of registers: one dedicated for instructions and the other for data

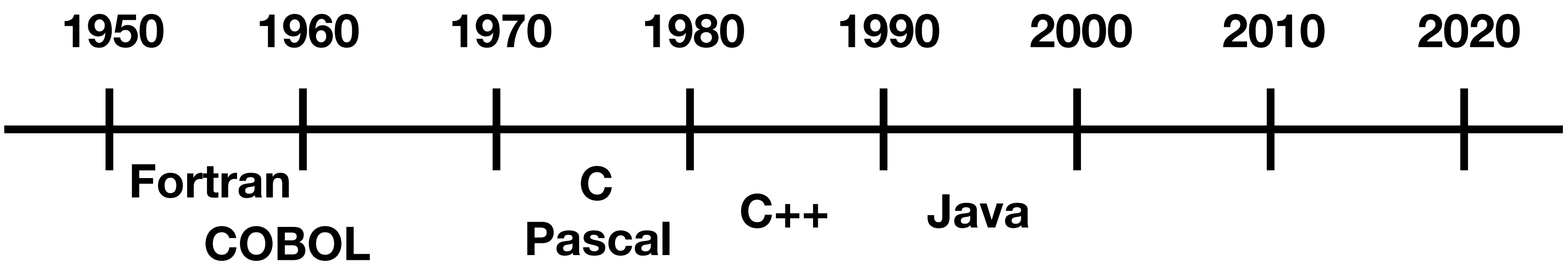
Modern Von Neumann Architecture



Needs a high-level language

High-level Language Timeline (Abridged)

- FORTRAN (1956) - FORmula TRANslator, for scientific applications
- COBOL (1960) - for business applications
- Pascal and C (1970's) - block structured
- C++ (1980's) - the OOP version of C
- Java (1990's) - a platform-independent language for the Internet, also OOP

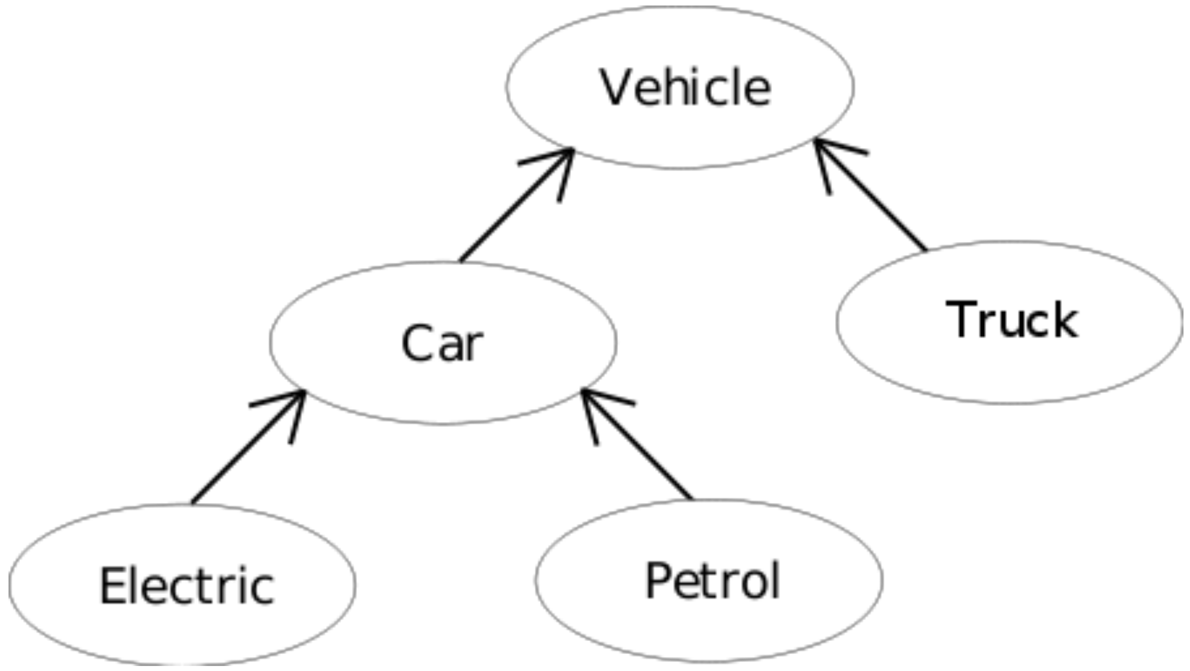


OOP - Object-Oriented Programming

- OOP models a world of **active objects**.
- An object has “**memory**” or “**state**”, and can contain other objects.
- An object has “**behaviors**” or “**methods**” that process messages from other objects.
- An object’s method can change it’s state, send messages to other objects, and create new objects.
- An object belongs to a particular **class.** A class determines the functionality of all objects that belong to that class.
- Programmers define classes to create an OOP application.

Main OOP Concepts

- **Inheritance**: a <u>subclass</u> can take on all of the attributes (states) and behaviors (methods) of another class, can redefine those behaviors, add new behaviors, and add new attributes.

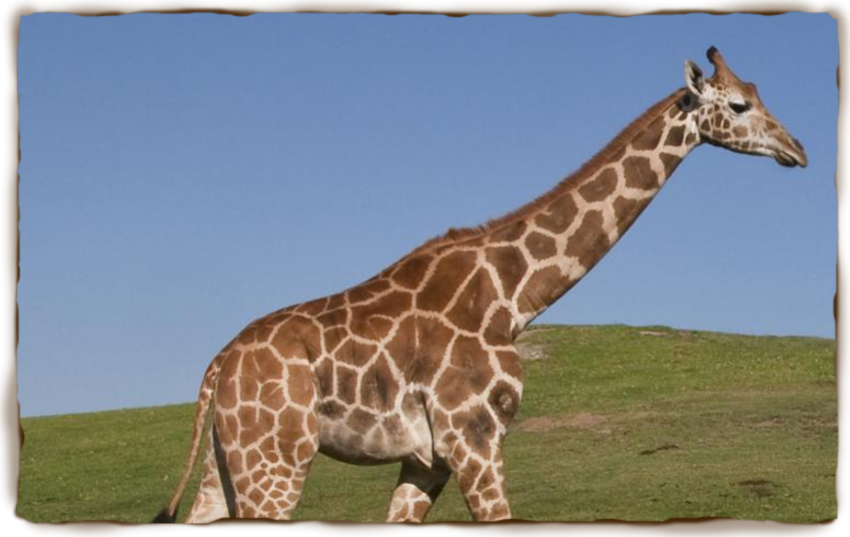


Main OOP Concepts

- **Polymorphism:** to process objects differently based upon their data types

Mammals

Function: eats()

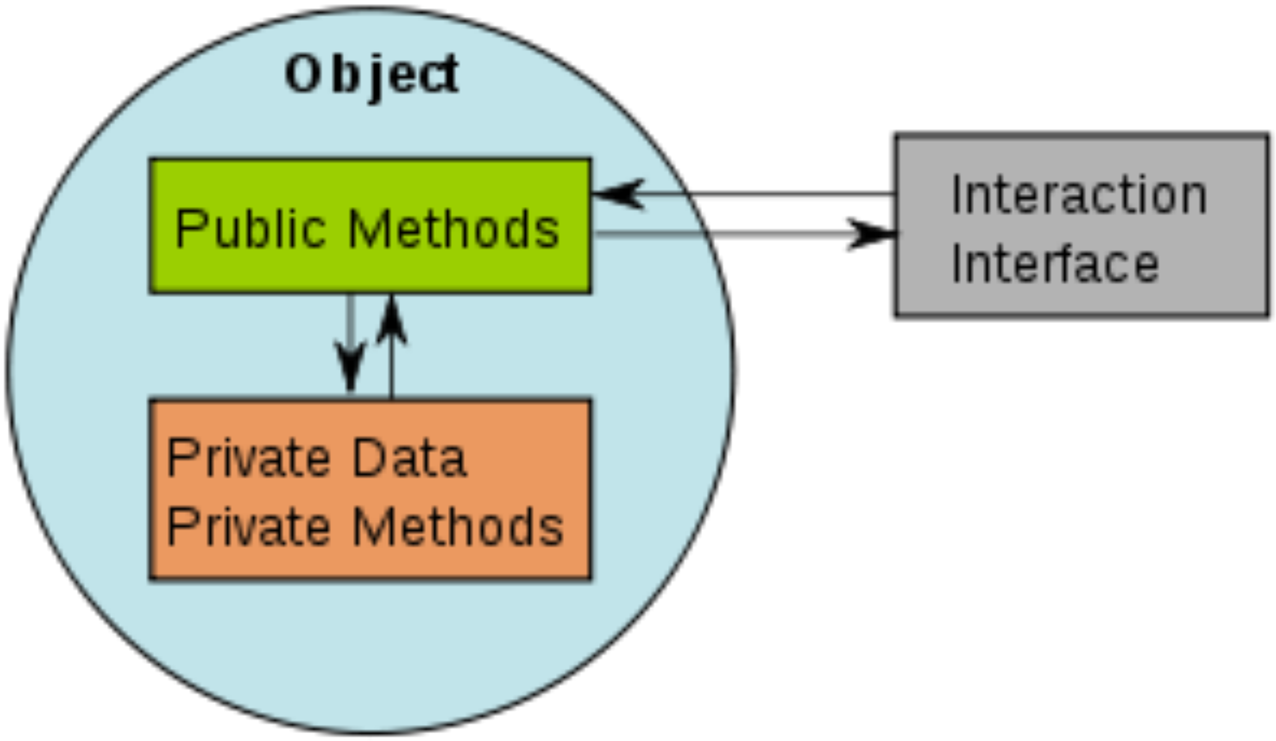


Function: eats()

eats() is defined differently for each.

Main OOP Concepts

- **Encapsulation:** keeps the data and code safe from outside influence



OOP Benefits

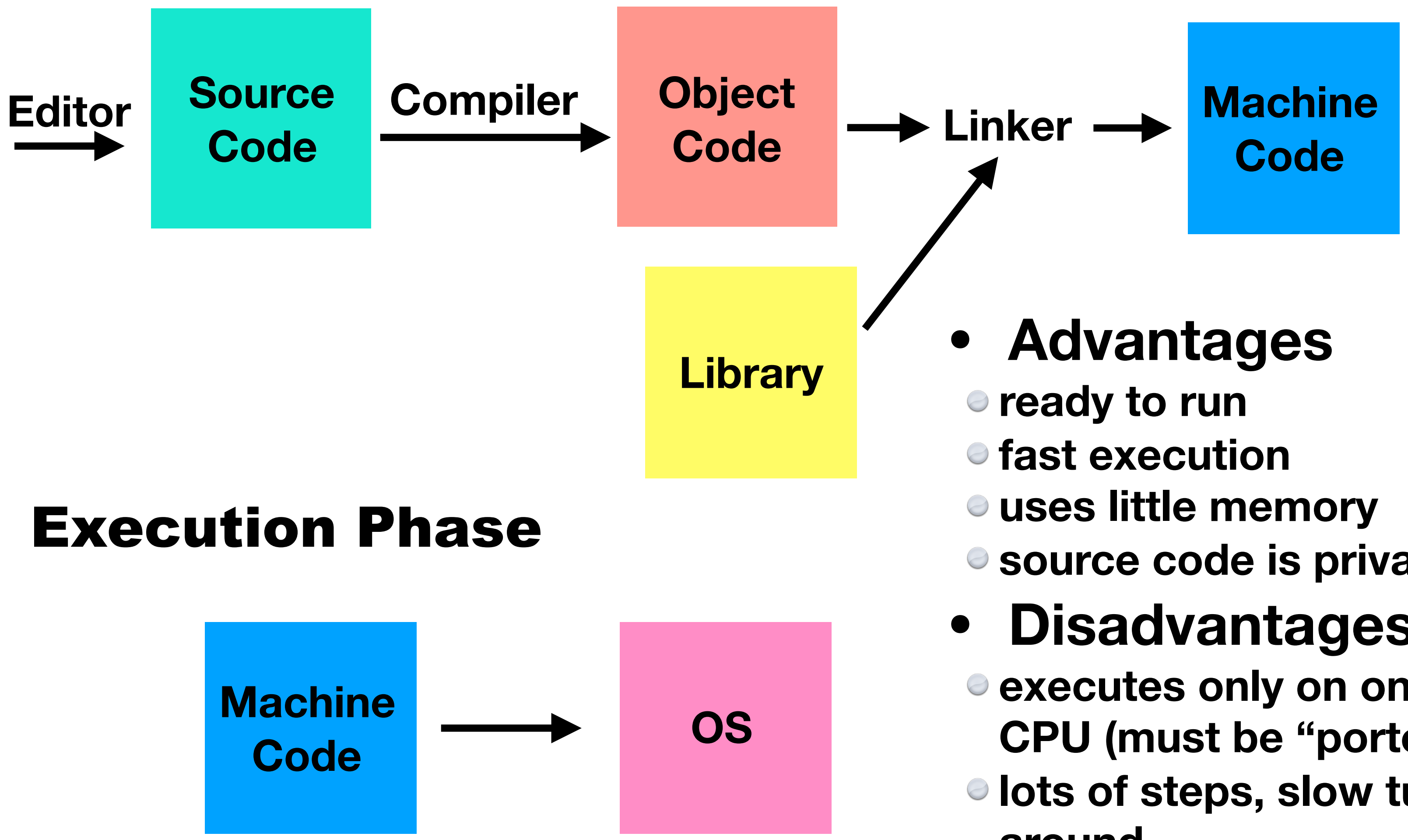
- Easier to reuse components
- Easier to maintain
- Allows for team development

High-level Language Development Environments

File Types

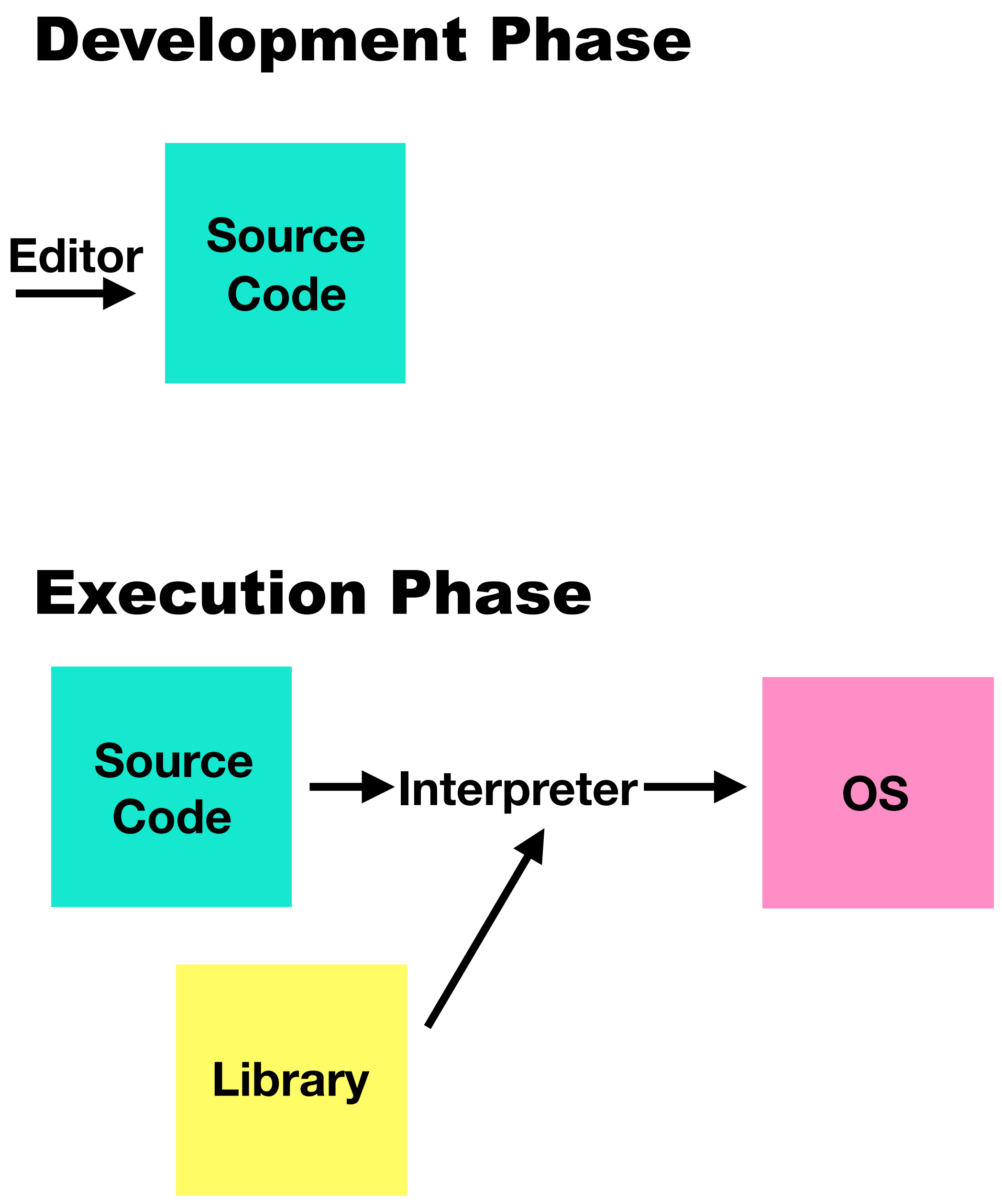
- **source code** - a file that contains the program in a programming language
- **object code** - a file that is generated by the compiler which contains the program in a form specific to the CPU
- **machine code** - an executable file that runs on a specific CPU
- **bytecode** - a file generated by the Java compiler that can be run by the Java interpreter. It is neither object nor machine code!

C Program (Compiler Paradigm)



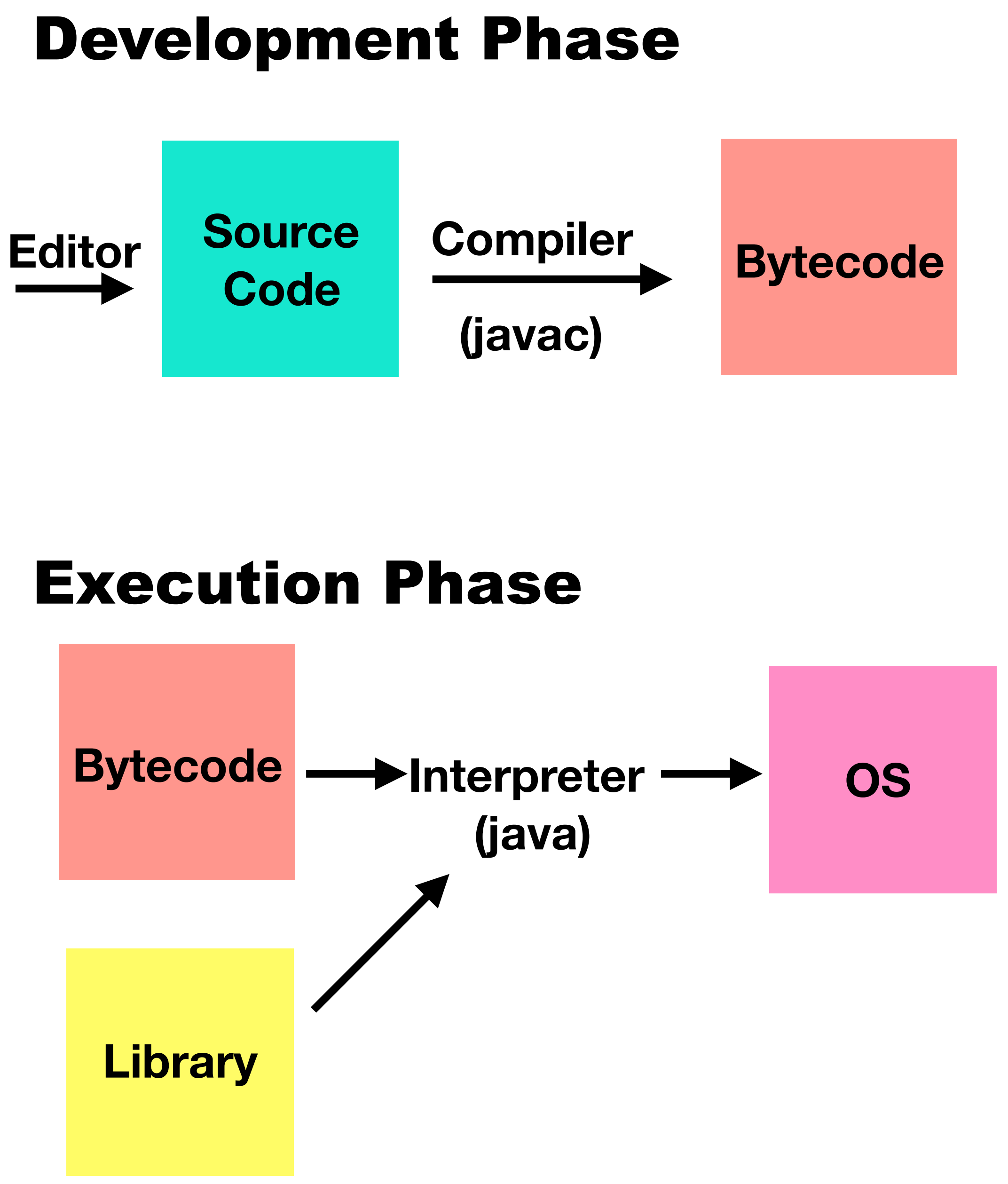
- **Advantages**
  - ready to run
  - fast execution
  - uses little memory
  - source code is private
- **Disadvantages**
  - executes only on one CPU (must be "ported")
  - lots of steps, slow turn-around
  - inflexible

BASIC Program (Interpreter Paradigm)



- **Advantages**
  - portable between CPUs
  - dynamic typing and scoping
  - simple to test
  - easy to debug
- **Disadvantages**
  - slower than other paradigms
  - uses more memory
  - requires interpreter specific to each CPU and OS
  - source code not private

Java Program (Compiler + Interpreter Paradigm)



- **Advantages**
  - Bytecode portable between CPUs and OSs
  - runs faster than Interpreter Paradigm
  - source code is private
  - interpreter can perform extra checking (like virus checking)
- **Disadvantages**
  - every code change requires recompiling
  - requires compiler and interpreter ported to each CPU and OS
  - generally slower than Compiler Paradigm

Programs for Development

- **editor** - a program that provides the user a GUI to input text and output a source file
- **compiler** - a program that converts high-level code into object code (for CPU) or bytecode (for Virtual Machine)
- **linker** - a program that “links” the object code with prebuilt library functions
- **interpreter** - a program that reads and executes source code or bytecode, and contains an embedded linker

Java Compiler

- **Checks program syntax** - reports syntax errors based on strict syntactic rules

- **Creates bytecode file** - only when there are no syntax errors

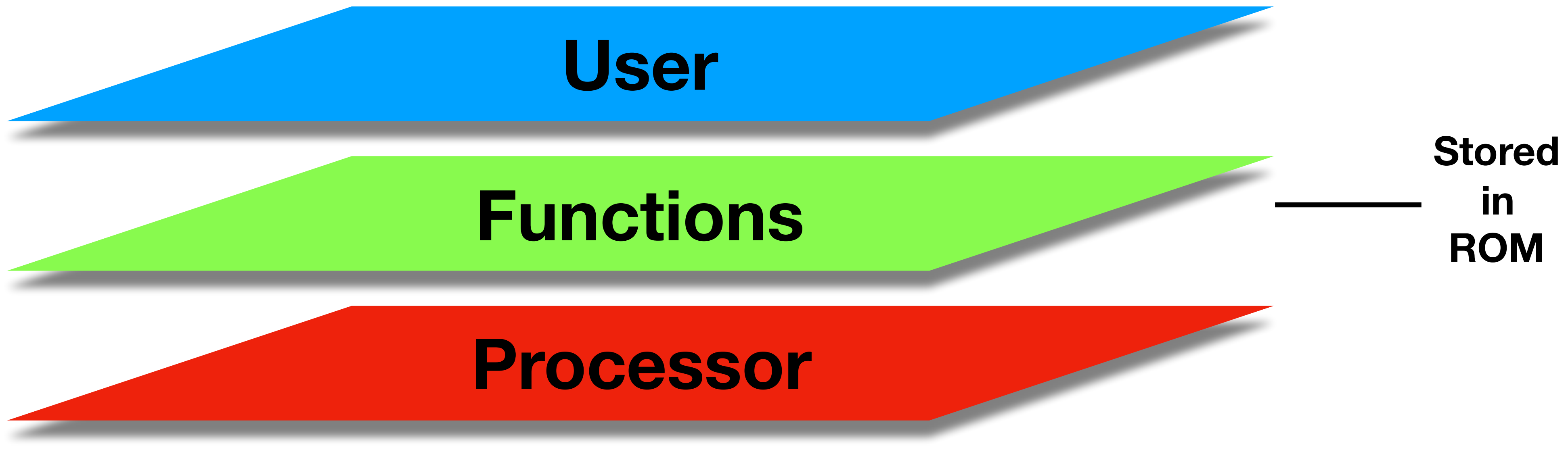
```
% javac MyPerfectProgram.java
MyPerfectProgram.java:7: cannot find symbol
symbol  : variable i
location: class MyPerfectProgram
       i = 1;
       ^
MyPerfectProgram.java:8: incompatible types
found   : int
required: boolean
       for (int a; 5; a++)
                   ^
2 errors
%
```

Java Interpreter

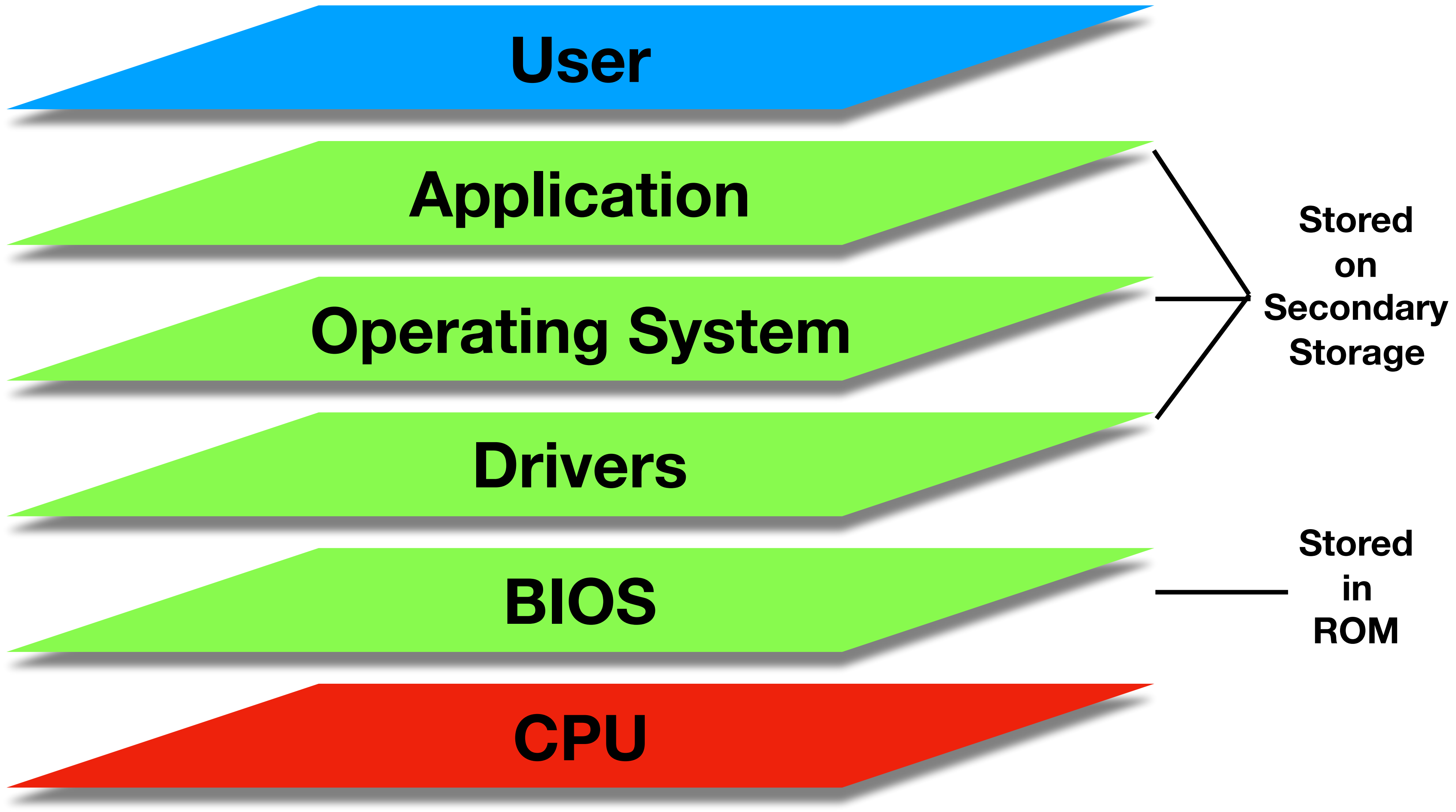
- **Links and executes the bytecode program**
- **Allocates memory during runtime**
- **Catches and reports runtime errors** - allows for a graceful exit to the OS if there is a problem

```
% java HereWeGo
Exception in thread "main"
java.lang.ArrayIndexOutOfBoundsException: 5
   at HereWeGo.main(HereWeGo.java:10)
%
```

Layers of a Simple Calculator



Layers in a Computer



The Java Development Environment

JDK - Java Development Kit

- **javac**
  - Java compiler
- **java**
  - Java interpreter
- **javadoc**
  - Generates HTML documentation from the source code
- **jar**
  - Creates a Java package file (JAR file)

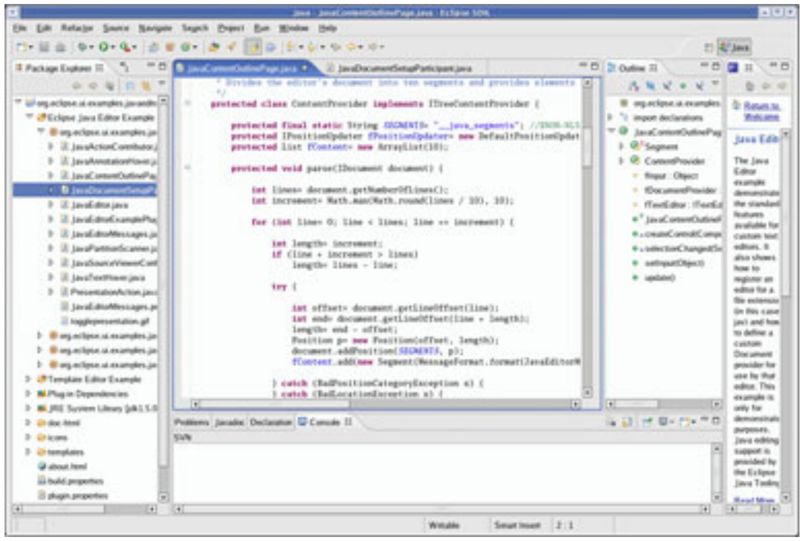
Command line tools, no GUI

JDK - <u>J</u>ava <u>D</u>evelopment <u>K</u>it (cont.)

- **Originally developed by Sun Computer (now Oracle)**
  - Jim Gosling - “Father of Java”
- **Download is available for all OS platforms (we use runtime version 7 in the lab)**
  - For PC or Linux: Google “jdk 7”
  - For Mac: Google “legacy jdk 6 mac” (unfortunately, must have maintenance agreement to get version 7)

IDE
(Integrated Development Environment)

- **GUI frontend for programming languages**
- **Integrates editor, compiler, interpreter, and debugger into one tool**
- **Popular IDEs for Java**
  - Eclipse
  - BlueJ
  - NetBeans



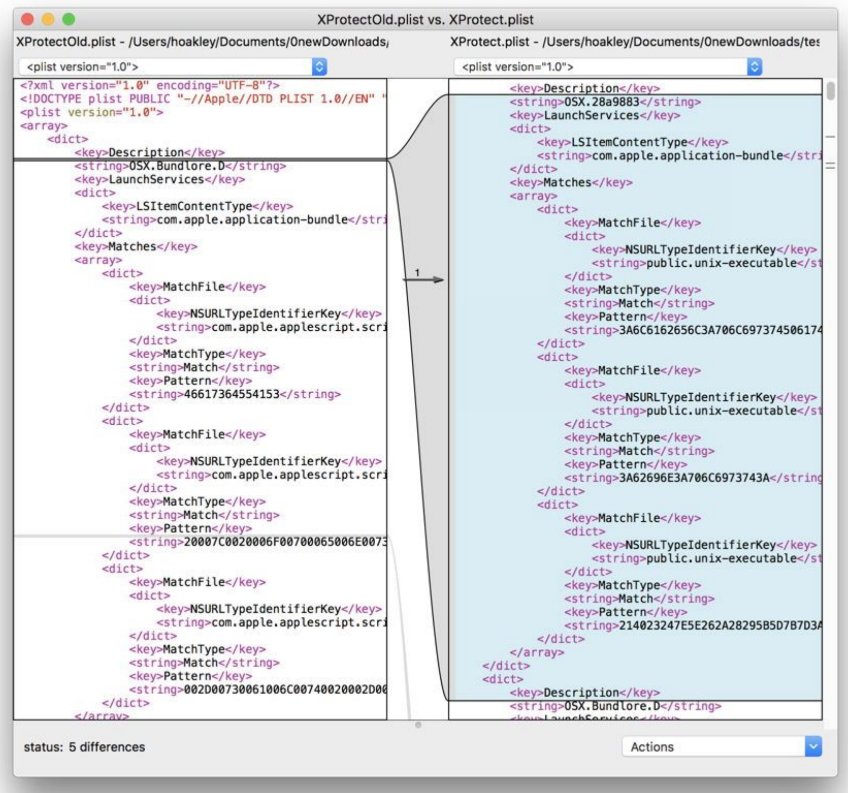
```
 * Divides the editor's document into ten segments and provides elements
 */
protected class ContentProvider implements ITreeContentProvider {

    protected final static String SEGMENTS= "__java_segments"; //$NON-NLS
    protected IPositionUpdater fPositionUpdater= new DefaultPositionUpdat
    protected List fContent= new ArrayList(10);

    protected void parse(IDocument document) {

        int lines= document.getNumberOfLines();
        int increment= Math.max(Math.round(lines / 10), 10);

        for (int line= 0; line < lines; line += increment) {

            int length= increment;
            if (line + increment > lines)
                length= lines - line;

            try {

                int offset= document.getLineOffset(line);
                int end= document.getLineOffset(line + length);
                length= end - offset;
                Position p= new Position(offset, length);
                document.addPosition(SEGMENTS, p);
                fContent.add(new Segment(MessageFormat.format(JavaEditorM

            } catch (BadPositionCategoryException x) {
            } catch (BadLocationException x) {
```

- **The IDE good, bad, and ugly**
  - Good: Speeds the development process.
  - Bad: Hinders the process of learning languages. Mr Greenstein does not provide help if you have an IDE problem.
  - Ugly: <u>Constant use could negatively affect your grade!</u>

My advice: DO NOT USE IDE!

Text Editor

- **Creates the Java source code file**
- **Recommend using bare-bones editors**
  - For Linux we recommend *Geany*
  - For PC we recommend *NotePad* or *Sublime*
  - For Mac we recommend *BBEdit (free version)*


XProtectOld.plist vs. XProtect.plist
XProtectOld.plist - /Users/hoakley/Documents/OnewDownloads/
XProtect.plist - /Users/hoakley/Documents/OnewDownloads/tes
<plist version="1.0">
<plist version="1.0">
1
status: 5 differences
Actions

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "
<plist version="1.0">
<array>
    <dict>
        <key>Description</key>
        <string>OSX.Bundlore.D</string>
        <key>LaunchServices</key>
        <dict>
            <key>LSItemContentType</key>
            <string>com.apple.application-bundle</stri
        </dict>
        <key>Matches</key>
        <array>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>com.apple.applescript.scri
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>46617364554153</string>
            </dict>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>com.apple.applescript.scri
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>20007C0020006F00700065006E0073
            </dict>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>com.apple.applescript.scri
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>002D00730061006C00740020002D00
            </dict>
        </array>
```

```
        <key>Description</key>
        <string>OSX.28a9883</string>
        <key>LaunchServices</key>
        <dict>
            <key>LSItemContentType</key>
            <string>com.apple.application-bundle</stri
        </dict>
        <key>Matches</key>
        <array>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>public.unix-executable</st
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>3A6C6162656C3A706C697374506174
            </dict>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>public.unix-executable</st
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>3A62696E3A706C6973743A</string
            </dict>
            <dict>
                <key>MatchFile</key>
                <dict>
                    <key>NSURLTypeIdentifierKey</key>
                    <string>public.unix-executable</st
                </dict>
                <key>MatchType</key>
                <string>Match</string>
                <key>Pattern</key>
                <string>214023247E5E262A28295B5D7B7D3A
            </dict>
        </array>
    </dict>
    <dict>
        <key>Description</key>
        <string>OSX.Bundlore.D</string>
```

Questions???