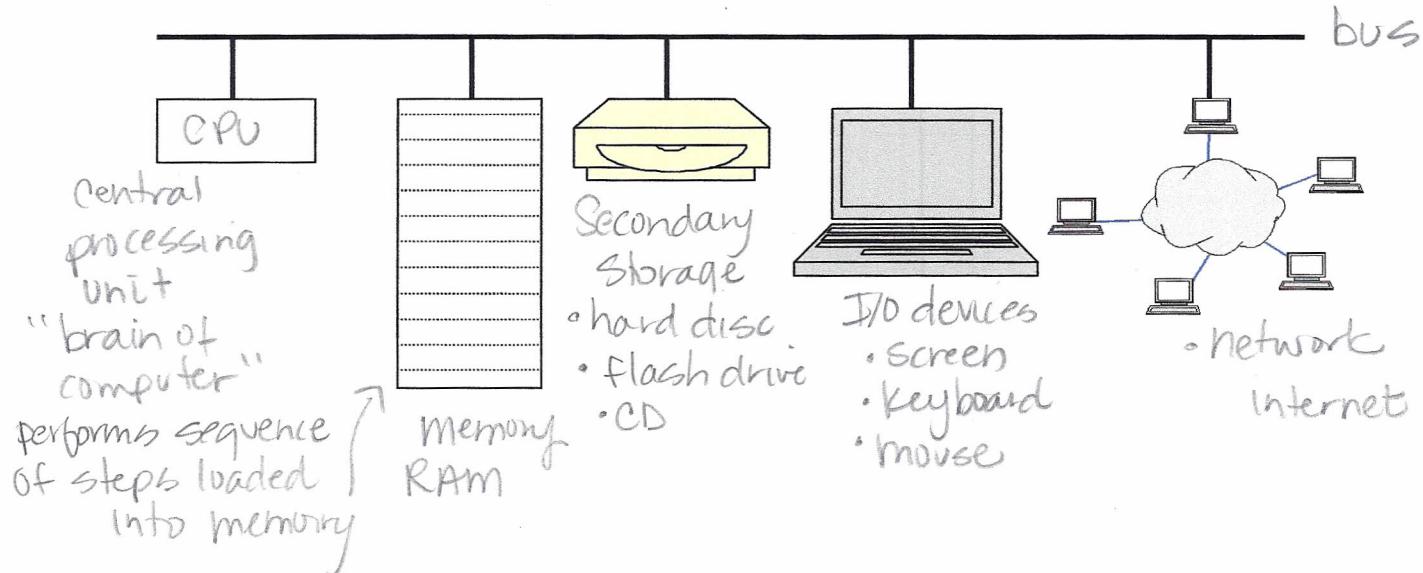


Introduction

Computer Science is the science of problem solving where the solutions happen to use a computer.

A computer program is a set of instructions to a computer to allow it to process some data. These programs are generically known as software.

Hardware Basics



Memory

Memory is where actively running programs reside. Access to this area is very fast and is referred to as RAM or random access memory. Any information stored in this area is lost when the computer is turned off.

Each memory location is 8 bits or 1 byte long. Each bit is either a 1 or 0. Data that cannot fit in one byte can use consecutive locations. Each memory location has a unique address represented by a hexadecimal number. The memory address is like the address of your home. With the address, the software can locate the information it needs.

<u>address</u>	
(hex)	
4001	00000011
4002	A
4003	true
4004	5851
4005	
primitive data types	:
Object addresses	

upper memory

int value 3
char
boolean
<Object reference>



Number Systems

Binary (base 2) 0 or 1							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
0	1	1	0	0	1	0	1
0	1	1	1	1	0	1	1

Octal (base 8) 0,1,2,3,4,5,6,7			
8^3	8^2	8^1	8^0
512	64	8	1
0	0	2	8
0	4	0	1

all info stored as 1's and 0's in computer

Decimal

(base 10)

$$= 1(1) + 1(4) + 1(32) + 1(64) = 101$$

$$= 123$$

$$123/2 \rightarrow 1 \text{ remainder}$$

$$61/2 \rightarrow 1$$

$$30/2 \rightarrow 0$$

$$15/2 \rightarrow 1$$

$$7/2 \rightarrow 1$$

$$3/2 \rightarrow 1$$

1

Check! $\begin{array}{r} 64 \\ + 32 \\ \hline 96 \end{array}$

$\begin{array}{r} 96 \\ + 16 \\ \hline 112 \end{array}$

$\begin{array}{r} 112 \\ + 8 \\ \hline 120 \end{array}$

$\begin{array}{r} 120 \\ + 2 \\ \hline 122 \end{array}$

$\begin{array}{r} 122 \\ + 1 \\ \hline 123 \end{array} \checkmark$

$$257/8 \rightarrow 1$$

$$32/8 \rightarrow 0$$

14

$$= 8(1) + 2(8) = 8 + 16 = 24$$

$$= 257$$

$$4(64) + 6(8) + 1(1) = 256 + 1$$

$$= 257 \checkmark$$

Hexadecimal (base 16) 0,1,2,3,4,5,6,7, 8,9,A,B,C,D,E,F			
16^3	16^2	16^1	16^0
4096	256	16	1
4	0	F	B
0	1	4	1

Decimal

(base 10)

$$= 11(1) + 15(16) + 0 + 4(4096) =$$

$$= 321 \text{ Check } 1(256) + 4(16) + 1(1) = 321 \checkmark$$

$$321/16 \rightarrow 1$$

$$209/16 \rightarrow 4$$

1

Solve and express answer as a decimal:

$$1. 12(\text{octal}) + 42(\text{decimal}) =$$

$$1(8) + 2(1) + 42 = \boxed{52}$$

$$2. 1E(\text{hex}) + 17(\text{octal}) + 2(\text{decimal}) =$$

$$1(16) + 14(1) + 1(8) + 7(1) + 2 = 16 + 14 + 8 + 7 + 2 =$$

$$30 + 17 = \boxed{47}$$

Algorithms

Algorithms are a strategy for solving a problem. These abstract solutions will be the core of the programs you write. At first the algorithms will be simple and come right to mind. As you move through the course, they will become more complex and will require more thought and planning. Write an algorithm to print the minimum of two numbers, x & y:

IF $x > y$
 print y
otherwise
 print x

Programming Process

There are two steps to the programming process:

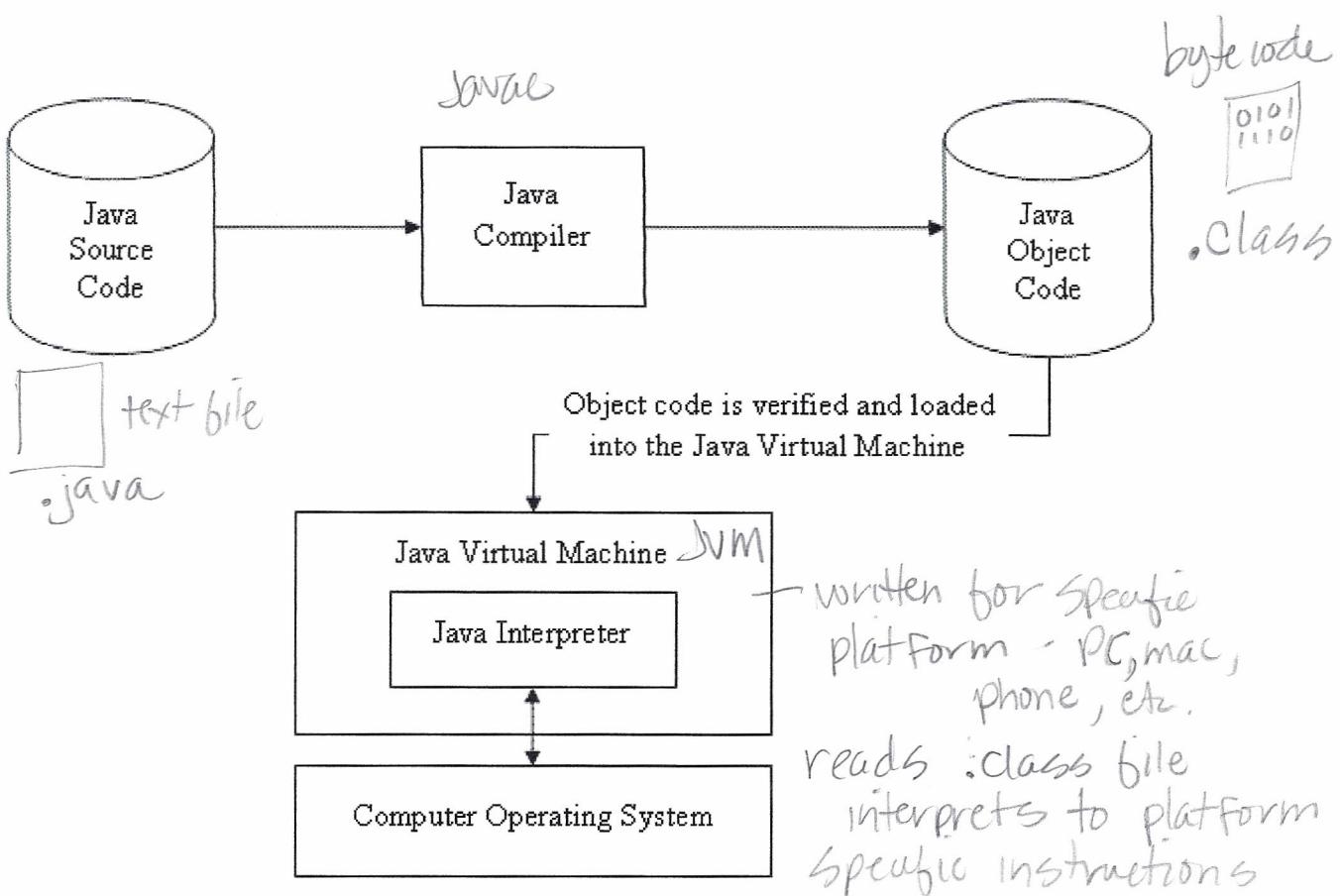
1. Develop or choose an existing algorithm. This is called algorithmic design.
2. Express the algorithm as a computer program in a programming language. This is called coding.

At first, coding will seem to be the most difficult part of programming. You are learning a new language whose Syntax and rules are unfamiliar. DO NOT GIVE UP!! Please be reassured that coding will become easier as you learn more about the programming process.

Computer programs can be written in several different programming languages. In this class you will learn Java to illustrate the more general concepts of programming and computer science. Java is called an Object oriented programming language. Other languages include: Objective-C (iPhone), C++, Pascal, Fortran, C#. Java was written by James Gosling at Sun Microsystems (now Oracle) in 1975.

"Write once, run anywhere" is a slogan created by Sun Microsystems to illustrate the cross-platform benefits of the Java language. Ideally, this means Java can be developed on any device, compiled into a standard bytecode machine and be expected to run on any device equipped with a Java virtual machine (JVM). The installation of a JVM or Java interpreter on chips, devices or software packages has become an industry standard practice.

This is intended to save software developers the effort of writing a different version of their software for each platform or operating system they intend to deploy on.



Programming languages are made up of English words which have meaning to the computer. There are rules when writing the words into a program. The rules (program syntax) let the compiler change the words into instructions (machine language or byte code) the computer can understand.

When you compile a program, the compiler first checks to see whether your program is syntactically correct. If you have violated the syntactic rules, the compiler displays an error message. These errors are called syntax errors. These errors can be frustrating, but your biggest source of frustration will occur when your program compiles but fails to operate correctly. This type of mistake is called a bug and the process of finding and fixing such mistakes is called debugging. All programmers make logic errors. You will make logic errors and it is your job as a programmer to find and fix the bugs!

Some of the English words (called Identifiers) have special meaning in Java, these are called reserved words. Some examples are `public`, `static`, and `class`. Below are the Java reserved words:

abstract	else	interface	super
assert	enum	long	switch
boolean	extends	native	synchronized
break	false	new	this
byte	final	null	throw
case	finally	package	throws
catch	float	private	transient
char	for	protected	true
class	goto	public	try
const	if	return	void
continue	implements	short	volatile
default	import	static	while
do	instanceof	strictfp	
double	int		

Other identifiers are made up by other programmers or yourself and can represent class names, variables or methods. Identifiers can be made up of letter, digits, underscore and dollar sign characters, are case sensitive and cannot begin with a digit.

Which of the following identifiers are legal in Java?

ILoveAPCS yes

2_names no

\$myNum no

iloveapcs! no

my\$num yes

MY\$NUM yes

I_love_apcs yes

Account_2 yes

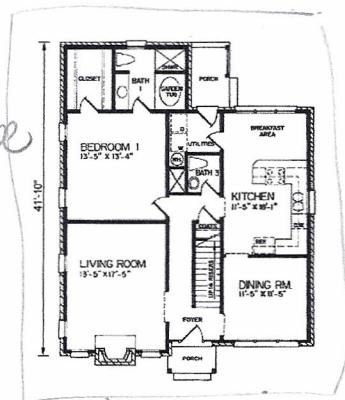
account^num no

→ show Call of Duty clip

Every Java program is made up of classes. Classes are usually small and are created for a specific function. Classes are like blueprints, they define what something is going to look like (called Variables) and how it will behave (called methods). lego pieces lego instructions

Imagine a blueprint for a house. You know what the house will look like and how it will function by looking at the blueprint, but the house doesn't exist yet. You cannot live in a blueprint. You have to create the house (Object) from the blueprint (class). This creation is called Instantiation. The house object is instantiated from the house blueprint (class). In other words, the house is an instance of the house blueprint. Many houses can be built (instantiated) from one blueprint.

new - created object from class



... color = "green"
rooftype = "tile"



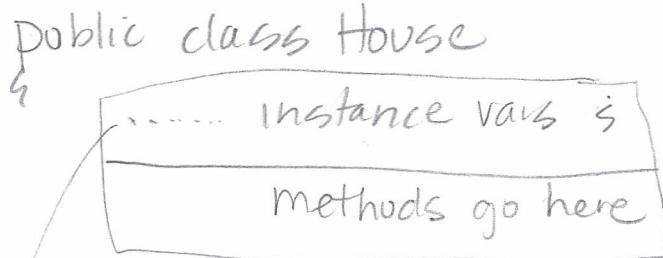
color = "red"
rooftype = "shingle"



color = "red"
rooftype = "shingle"

Class Creation

Each class is written in a separate class file with a file extension .java. The convention is to start class names with a capital letter. Class names and the .java file name are identical. Each class is defined with a class heading using this syntax (rules for declaring a class):



Classes contain information specific to the class, called instance variables. Instance variables must be defined with a type and variable name. Types can be one of Java's primitive types (int -100, 3.75, boolean true or false or double 3.725) or an object type (class name). The convention is to start variable names with a lower case letter.

```
public int x = 7;
private boolean fun = true;
private double cost = 2.75;
String address;
```

Classes also contain behavior called methods. Each method has a set of instructions that perform a function. The set of instructions is the algorithm for the method. Each statement in the method is a specific instruction. Statements end with a Semi-colon. Method declarations follow this syntax:

```
public int min (int x, int y)
{
    if (x < y)
        return x;
    else
        return y;
}
```

Each program you write will contain a class with a method named main. The JVM will look for this method when you run your program. This method will also create objects from the other classes in your program, but will not become an object itself.

```
public static void main (String [] args)
```

Write a class named MyMessage that will print the message I love computer science!:

```
public class MyMessage
{
    public static void main (String [] args)
    {
        System.out.println("I love computer science!");
    }
}
```

Java API

The Java API contains all of the class information for the many classes that come with Java. You have already used two of these ready-made classes, System and String. Become familiar with the Java API.

Open the API up (Java API icon on your desktop) and look at the instance variables and methods for the following classes that you will be required to know for the AP Exam and answer the questions in the space provided.

Integer What are the MAX_VALUE and MIN_VALUE for this object?

Double Look under the “Method Summary” section. What does the method longBitsToDouble() do?

String Look under the “Method Summary” section. What does the method length() do?

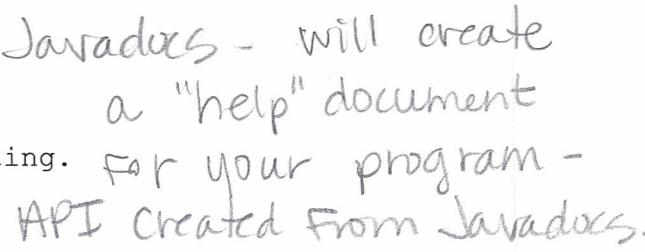
Math Look under the “Method Summary” section. What does the method max(double a, double b) do?

ArrayList Look under the “Method Detail” section. What does the method isEmpty() return?

System Look under the “Field Detail” section. What does it say under the field in?

Comments

Comments are a very important part of your program. They are written for human beings not the computer. The compiler ignores comments. The three types of comments in Java are:

1. 
/*
 * File: MyMessage.java
 * This program prints out a simple message to the console
 / / & */ mark start and end of comment - computer
 won't read anything in between.
2. // Use this type for short, one-line comments.
3. 
/** Start comment
 * File: CallOfDuty.java
 * @author Mrs. Allen
 * This program simulates war time fighting.
 */ End comment
@ have special meaning.

A few words on JCcreator

JCcreator is an IDE (Integrated Development Environment) which allows programmers to write, compile and execute Java programs.

- **write** – Using the Java commands and syntax, create class files and save with the .java file extension.
- **compile** – Invoke the Java compiler to read the class files and if there are no syntax errors, convert the Java language into bytecode or machine code (.class file) which can be read only by computers.
- **execute** – Invoke the JVM (Java Virtual Machine) to read the .class bytecode file and carry out the instructions of the program.

You can set up your file structure in many different ways. To help organize your programs, you will need to create an \APCS folder in your student director. Each chapter will have a .zip file to download from Moodle, containing the chapter lab manual and any lab files. Keep each chapter separate by using the folders created during the unzip. They will be named \Chapter 1, \Chapter 2, etc.

The first time you start JCcreator, you will need to do the following:

1. Start *JCcreator*.
2. Make sure there is no workspace open (choose *File/Close Workspace* if one is open). From the *Configure* menu, choose *Options...* In the dialog box that pops up expand the *Editor* node and click on *Java*.
3. Set the tab size to 2 and select the *Insert spaces* option.
4. Create an \APCS folder in your student directory – each chapter will have a separate folder containing the chapter lab manual and lab files.
5. Click on the *Directories* node and enter the path of the \APCS in your student directory in the *Default Project Directory* line.
6. Click on the *Code Insertion* node and check the *Start open brace on new line* box.
7. Then click *OK*.

- text editor like Notepad
- javac to compile
- java to run

- can use BlueJ, eclipse
or command prompt