AP Computer Science A
(Java Programming 1)

Mr. Wie

WiFi:
`ArdentGuest`

Password:
`MathNScience`

Remind App
http://www.remind.com/join/**ardentapcs**

Text
(949) 542-6858

Email
gene.wie@gmail.com

--

What language do humans communicate in?

Natural Language
Speech, gestures, writing

English, Korean, Japanese, Spanish, French, Chinese, etc.

What language do machines communicate in?

Binary - 1's and 0's

Transistor - state-based device that can be set to on or off, to represent true/false or 1/0. A modern CPU (central processing unit) in a computer can contain millions of transistors.

Business office light switch (room)

s1	**s2**
0	0
0	1
1	0
1	1

This lighting is controlled by two switches, which account for four total possible states.

Each 1 or 0 is called a **bit**, and thus this light switch system stores 2-bit binary numbers.

1 byte = 8 bits
1 kilobyte = 1024 bytes
1 megabyte = 1024 kilobytes
1 gigabyte = 1024 megabytes
1 terabyte = 1024 gigabytes

How many 1's and 0's can an iPhone with 64GB of storage hold?

549,755,813,888

--

What do humans have to do in order to communicate with a machine?

We need some layers to translate machine language (1's and 0's) to natural language (English).

We need to explore different number-bases first in order to get from binary to decimal.

decimal: base-10

286

2 x 10^2 + 8 * 10 ^1 + 6 * 10^0
200 + 80 + 6 = 286

binary: base-2

1101

1 * 2^3 + 1 * 2^2 + 0 * 2^1 + 1 * 2^0
8 + 4 + 0 + 1 = 13

hexadecimal: base-16

DEC	HEX
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
10	A
11	B
12	C
13	D
14	E
15	F

HTML color codes are expressed as six digit hexadecimal numbers like this:

hexadecimal: 3366FF

3	3	6	6	F	F
0011	0011	0110	0110	1111	1111

binary: 001100110110011011111111

ASCII
American Standard Code for Information Interchange

ASCII TABLE

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(	72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29	)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[	123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	|
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D	]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Each character in the ASCII table can be represented using an 8-bit binary number. Remember that 8 bits equals 1 byte! The byte is the smallest unit of data that most computers will display to you (a single ASCII character).

Why do computers have a hard time understanding natural language?

What aspect of natural language do computers have a hard time with?

Context!

Let's eat, grandma.
Let's eat grandma.

I love dogs.
I love dog.

--

Machines want you to be Literal and Precise.

--

Programming Languages

In order for us to communicate with machines and give them instructions, we use a *programming language* has a way to translate our natural language instructions into machine instructions. Programming languages range from low-level versions that are closer to binary to high-level versions that are closer to English.

HIGH level languages

Scratch
Processing
Python

Kotlin
C#
Swift

Java
C++
Objective-C

LOW level language

Assembler, assembly language
Motorola 68k, Intel x86, IBM PowerPC, etc.

--

The Java Programming language has been in common use for applications online for over 20 years.

The feature that makes Java useful is its **portability**. Java programs can be run on any computer that has the JVM (Java Virtual Machine) installed on it. Nearly every single personal computer these days comes with the JVM already integrated into the operating system, including Windows, macOS, Linux, and most smartphones.

In traditional programming languages, the code that you write is **compiled** by the language compiler into a **machine binary** which is run directly by a specific processor. Java however, has a compiler that creates an

intermediary form of code called **bytecode**. This bytecode is what is distributed online and then is taken by the JVM and converted into machine code for that specific computer. This allows a programmer to write a single program in Java, compile it to bytecode, and allow any machine with the JVM to run it. Java was one of the first languages to have this kind of portability.

The tradeoff is that Java does not run as quickly as programs compiled to direct binaries from languages like C and C++. As a result, most gaming titles are not written in a language like Java, but in C++ and its derivatives. Publishers of AAA game series still rely on other languages which compile directly to binaries, instead of Java.

--

IDE - Integrated Development Environment

An IDE is a collection of tools that is used to write computer programs. It includes three major elements:

1. Editor - this is where you write your code. The editor provides support in the form of automatic indentation, highlighting, line counts, etc. Think of it as "Microsoft Word" for your code.

2. Compiler - this takes your written code and converts it into machine code. For Java, it generates bytecode that can be compiled into machine-specific binaries by the JVM.

3. Debugger - this tool allows us to pause a program while it is running and examine the current state of all things inside the program.

The IDE you'll be using for Java program is an educational one called **BlueJ**. While we would not ever develop full software using it, it has a visual interface that is extremely helpful for us to understand how the Java language works.

You can download the latest version at **www.bluej.org.**

Eclipse, Netbeans, IntelliJ IDEA are also possible options for those with previous programming experience.

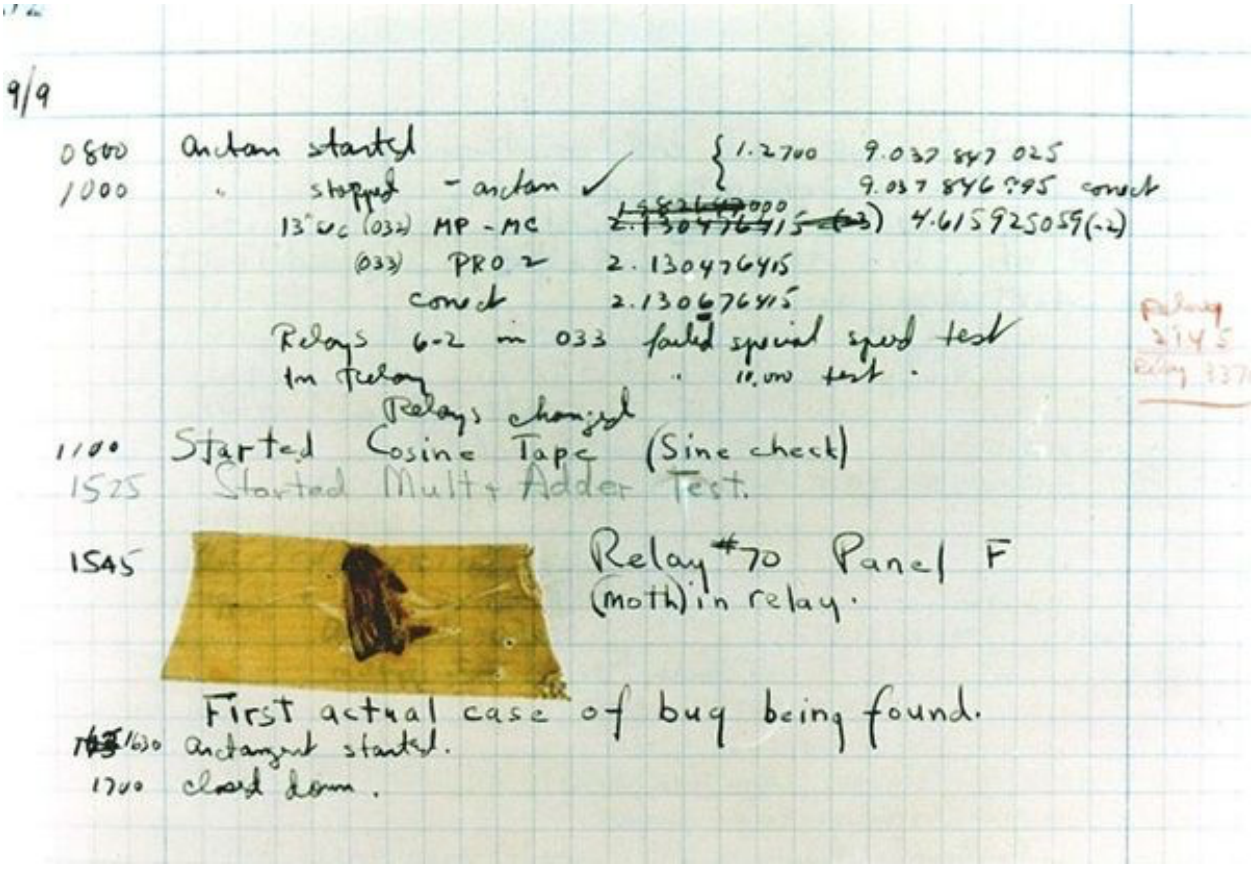
For educational and scientific use, you can download versions of all these IDE's at no cost.

--

For the moment, a **class** in Java is like a container that we use to hold our code. We will define it in more detail later.

An error in a computer program, like a **syntax error** (where we have typed something the compiler does not understand), is often called a **bug.**

In 1947, computers used moving parts called relays to connect different parts of their circuits. Grace Murray Hopper recorded the first "computer bug" when a moth got stuck in one of the relay contacts, which was causing a math error in the computer's output.



Today, we refer to all computer problems as *bugs* and we write them down for future corrections when we create big software systems in *bug trackers*.

What makes up a computer program?

While "Hello, World!" is am example of a simple program, it doesn't much except output. For me, a complete program includes the following elements:

1. Input
2. Computation
3. Output

This is where a computer really shines...it should be able to complete tasks that would take us excessive amounts of time, complete them very quickly, and not have the kinds of errors that we would make calculating them. In fact, this is one of the reasons why people want electronic voting machines, because even when ballots are hand-counted there can be errors. Using both systems together ensures more accuracy with election systems.

Input

Getting input in Java, at least from a text-based perspective, requires us to *import* a specific tool meant to handle it. It is called **Scanner.**

```
import java.util.Scanner;
```

To make use of the Scanner after we import it, we have to **instantiate** it, or create an **instance** of the Scanner:

```
Scanner input = new Scanner( System.in );
```

To save the name that they enter, I will create a **variable**. A variable is like a label that identifies a unique piece of information in the computer's memory. To declare one, I have to state it's **type** followed by its **name**.

```
String name = "Bob";
```

Data Types

Computers store different kinds of information, from numbers, to letters, to more complex things like games, simulations, images, audio, video, etc. Java has eight **primitive data types** to store numbers and letters and other simple concepts:

integer types:

byte	8 bits
short	16 bits
int	**32 bits**
long	64 bits

floating point types (decimals):

float	32 bits
double	**64 bits**

other:

boolean	true or false
char	single ASCII character (8 bits)

However, saving someone's name is a bit more complex. In that case, we have **Abstract Data Types (ADTs)** which hold more information.

String is an ADT which literally holds "a string of characters." It is the primary ADT we use in all computers today to represent text. Text can be as short as a name, or as long as all of the text in a book, or multiple books.

Primitive data types are built-in to Java, and don't require any supporting libraries to use (unlike Scanner). String is a special kind of ADT that is so commonly used, Java includes it already whether you want it or not. In most cases though, to use ADTs you must import them separately.

Math

Now that we can do input and output, let's do some computation so that we actually have a complete program by my definition.

Let's write a program that prompts the user for a distance in miles. Let's convert that from miles to kilometers, then display the result for the user.

When I write out the comments line-by-line with the steps that program hopes to accomplish, I am putting together something called an **algorithm.** An algorithm is like a

recipe that you follow to accomplish a task. For any program that you write, it is critical that you establish what the algorithm is first. Lots of critical failures in the software world can be attributed to people who didn't believe that they needed to write an algorithm first before writing code.

To manipulate numbers, we use operators:

Operators

+	addition
-	subtraction
*	multiplication
/	division
%	modulus/mod (remainder)

While there are other kinds of operators that we can use, they are not built-in and must be imported from Java.lang.Math.

Programming Exercise No. 1

Write a program that prompts the user for a temperature in Fahrenheit. Convert that temperature to Celsius and display the result.

Programming Exercise No. 2

Write a program that prompts the user for a distance in parsecs. Convert that distance to light-years, and display the result.

Programming Exercise No. 3

Write a program that prompts the user for a two-digit number. Tell us what digit in the tens place and what digit in the ones place. For example:

```
Enter a two-digit number: 84
Tens place: 8
Ones place: 4
```

What kind of primitive data type should we use?

int

What mathematical operators can we use to solve this problem?

/ division
% modulus/mod (remainder)

Did you run into any sort of error related to the mathematical operators? What was it?

When we do math with integers, if we ever get a result that has decimals/fractions, the integer type will simply **truncate** anything after the decimal. *It does not round the number, it floors it to the whole number without the decimal*.

Generate Random Numbers

We often need to generate random numbers (RNG - random number generation) to play a variety of games, like the Lottery, poker, blackjack, Monopoly, etc. For humans playing the physical versions of these games, that is generated using devices like dice.

Computers can't generate truly random numbers. They rely on using some sort of algorithm that is based around a number called a *seed value*. This value needs to be some number that changes constantly so that the results of the algorithm are *pseudorandom*. It is common to use the computer's clock, or system time, as this seed value.

How do computers store time?

They store it as a very big integer, that counts the number of seconds that has elapsed since January 1, 1980. We call this reference in time an *epoch*.

Social and Ethical Impacts of Computing

Alan Turing

Father of theoretical computer science and artificial intelligence.

"Turing Machine" are the principles on which all modern computer processors are built.

During WWII, he was part of the team that intercepted and decrypted the coded messages from a machine called ENIGMA used by the AXIS powers (especially the Nazi high command) to communicate with their commanders in the field. In breaking the ENIGMA code, Turing and his team are estimated to have shortened the length of the war in Europe by at least two years, and saved over 14 million lives.

--

ACM - Association of Computing Machinery
Turing Prize