

Panel 1

Last Time Quiz on Wed

Framework of every Java program

```

public class _____
{
    public static void main (String args [])
    {
        ---- BLAH
    }
}

```

3 Steps to create a program

Edit Source → Compile → Execute

Compiling + executing with BlueJ ✓

HW: Create a snowman or Olympic Rings ✓

Panel 2

Minor HW:

- Create a folder CSAS1111 in "My Documents"
- Every BlueJ project from now on should live inside that CSAS1111 folder
- Turn file extension viewing on!

→ Tools | Folder Options | View

Open any Folder Hide File Ext.
 uncheck

Panel 3

Why Java?

- Modern yet tried language (by SUN)
- Open + free
- Object - Oriented
- Great documentation
- Machine - independent (create once, run everywhere)
e.g. Mac or Linux or Sun

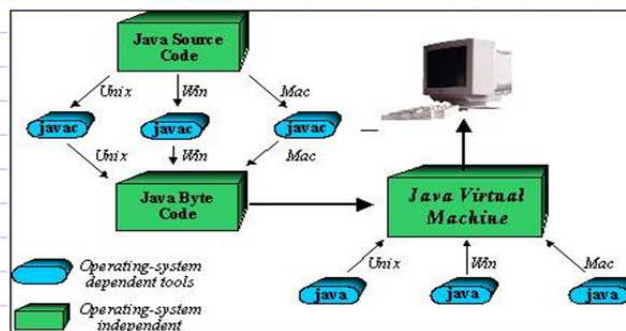
3

Panel 4

Definition 1.06: Java Virtual Machine (JVM)

The Java Virtual Machine (JVM) is a platform-independent engine used to run Java applets and applications. The JVM knows nothing of the Java programming language, but it does understand the particular file format of the platform and implementation independent class file produced by a Java compiler. Therefore, class files produced by a Java compiler on one system can execute without change on any system that can invoke a Java Virtual Machine.

When invoked with a particular class file, the JVM loads the file, goes through a verification process to ensure system security, and executes the instructions in that class file.



4

Panel 5

Data

Java understands 4 "atomic" or basic data types:

int - integer, or whole #, pos. or neg

char - character, single character

double - real or decimal #

boolean - logical value true or false

void - no type

(String) - list of characters (or text)

5

Panel 6

A little more details:

Definition 1.08: Primitive Java Data Types

Java supports the following primitive, or basic, data types:

- int, long, or short to represent integer numbers
- double or float to represent decimal numbers
- char to represent character values
- boolean to represent logical values
- void to represent "no type"

Each numeric type has a largest and smallest possible value, as indicated in table 1.10.

Definition 1.09: Literals

Literals are constant values for the basic data types. Java supports the following literals:

- int, short: digits only, with possible leading plus (+) or minus (-) sign
- long: like int literals, but must end with an "L"
- double: digits including possible periodic point or leading plus (+) or minus (-) sign, or numbers in scientific notation: #####E#####, where each # represents a digit
- float: like double literals, but must end with an "F"
- char: Single Unicode characters enclosed in single quotes, including the special control sequences described in table 1.11
- boolean: true or false

In addition, Java has an object literal called `null` for object references.

Examples: int: 1;42 , long: 123L
float: 5.2f , boolean: true

6

Panel 7

Binary Representation

Ultimately, every computer must store all data as a sequence of 0's and 1's inside switches!

A 0 or 1 is called 1 bit (not switches)

100101 is a number - which one?

First binary number:

0	=	0
1	=	1
10	=	2
11	=	3
100	=	4
101	=	5
110	=	6
111	=	7
→ 1000	=	8

First decimal number: 0

1
2
⋮
9
10
11
⋮
19
20

7

Panel 8

Convert 11001 from binary to decimal

① Make a table of powers of 2, right to left

2^5	2^4	2^3	2^2	2^1	2^0
32	16	8	4	2	1
		1	1	0	0
		1	0	0	0

② Fill in binary number, right to left

③ Add the "powers with a 1": $16 + 8 + 1 = 25$

Try 1000

8

Panel 9

Convert 13 to binary

- ① Make table of powers of 2
- ② Find largest power of 2 that fits your number, enter a 1 in that column
- ③ Find remainder and repeat

32	16	8	4	2	1
		1	1	0	1

$$\Rightarrow 13 = (1101)_2$$

9

Panel 10

10

Panel 11

Convert ~~(10010001)~~₍₂₎ to decimal

256	128	64	32	8	4	2	1	
1	0	0	1	0	0	0	1	

256								
	32							
		1						
			289					

Convert (11011011)₍₂₎ to decimal

256	128	64	32	8	4	2	1	256
1	1	0	1	0	0	1	1	128

256 + 128 + 32 + 8 + 2 + 1

Fix
as HW

32								
	8							
		2						
			1					31
			7					

11

Panel 12

Convert 31 to binary:

32	16	8	4	2	1	
		1	1	1	1	1

~~AO~~

Convert 321 to binary

HW

12

Panel 13

Counting in binary:

on one hand - 5 slots \rightarrow $\underbrace{\quad}_{\text{largest}}$ $\underbrace{\quad}_{\quad}$ $\underbrace{\quad}_{\quad}$ $\underbrace{\quad}_{\quad}$ $\underbrace{\quad}_{\quad}$

$2^5 = 32$

2^0 count to 1023 on both hands

Is $(1101001101)_2$ even or odd?

13

Panel 14

Definition 1.11: Bits and Bytes

Computer memory is measured in bits and bytes. One bit is a unit that can store either a 0 or a 1. A group of 8 bits is called one byte.

How many bits or bytes are required to store one boolean?

If I used 4 bytes to store integers, what is the biggest integer possible? How about plus/minus?

$\Rightarrow 2^{32} - 1 = 4,294,967,296 - 1$

Plus + Minus : $2^{31} - 1$ or $\approx 2^{32} / 2$

14

Panel 15

Suppose my PC has 512 MB of memory.

a) What is the largest integer it could store?

$$512 \text{ MB} = 512 \cdot 1000000 \text{ bytes} =$$

$$512000000 \times 8 \text{ bits}$$

$$2$$

b) If one integer takes 4 bytes, how many integers could be stored?

$$\frac{512000000}{4} \text{ integers}$$

15

Panel 16

All data types stored on a computer have a smallest and largest value. In Java:

Type	Range	
double	largest positive/negative value:	$\pm 1.7976931348623157 \text{E}308$
	smallest non-zero value:	$\pm 4.9 \text{E}-324$
	significant digits:	16 digits after decimal point
float	largest positive/negative value:	$\pm 3.4028235 \text{E}38$
	smallest non-zero value:	$\pm 1.4 \text{E}-45$
	significant digits:	8 digits after decimal point
int	largest value	2147483647
	smallest value:	-2147483648
short	largest value	32767
	smallest value:	-32768
long	largest value	9223372036854775807
	smallest value:	-9223372036854775808

16

Panel 17

Declare Variables

on Wed.

Quiz Wed.

17