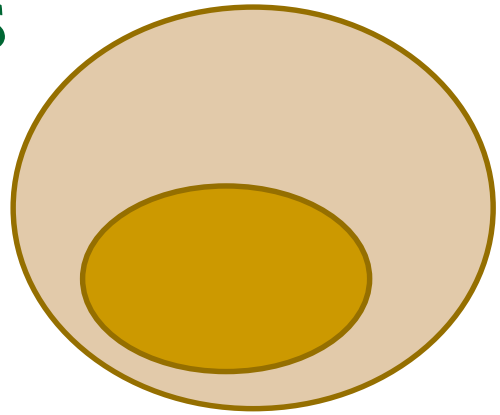
Java Primitives and Operations
(Base Set To Do ALL Computing)

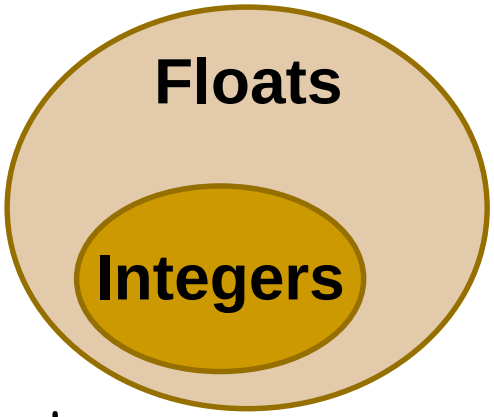
- Primitive data types
- Variable names
- while loop
- Operators (binary rep of info)

Base Types = Primitive Types

- byte
- short
- int
- long
- float
- double



Base Types = Primitive Types
continued




```
int bigInt = 2147483646; //2nd biggest int
int anInt;
float aFloat;

//Convert bigInt to a float (32 bits to 32 bits)
aFloat = bigInt;
System.out.println(bigInt + " " + aFloat);

//Convert it back to an integer
anInt = (int) aFloat;
System.out.println(bigInt + " " + anInt);
```

Base Types = Primitive Types continued

- boolean
- char
- Notice that there is no String as a base type

Naming Things

- Same as Python

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

Naming Things

- Conventions:
  - Data
  - Methods
  - Projects
  - Classes
  - Use CamelStyle
  - Be descriptive!

Classes

- All executable code must be in a class
  - Exception(?): import statements, project declarations

- Two Ways to Use a Class

1.

2.

ENCAPSULATION

Methods - Signature

- Two parts: _____ and _____
- Must be unique
- Return type being different does not affect uniqueness, as return type is not part of the signature
- E.g.

```
public static long power(int x, int y){
      if (y > 0)
            return (x * power(x, y - 1));
      else
            return 1;
}//power
```

Loop

- **`while`** is the basic loop in Java

- What should you do if need more than one statement to repeat?

Loop Example

```
1.   int kiwi = 20;
2.   int pineapple = 4;
3.   while (pineapple < kiwi){
4.      pineapple = pineapple + pineapple / 2;
5.      System.out.println(pineapple);
6.   }
```

Loop Example continued

```
1.   int kiwi = 20;
2.   int pineapple = 4;
3.   while (pineapple < kiwi){
4.      pineapple = pineapple + pineapple / 2;
5.      System.out.println(pineapple);
6.   }
```

- Control Variable(s):
- Initialization of Control Variable(s):
- Stopping Condition:
- Control Variable(s) Change Statement:

Loop Example continued

```
int kiwi = 20;
int pineapple = 4;
while (pineapple < kiwi)
   pineapple = pineapple + pineapple / 2;
   System.out.println(pineapple);
```

Write Java Code – Blast Off

- Write a Java method, blastOff, that takes a +ve integer and prints a triangular count down to 1, followed by "BLAST OFF"
- Write a main method to test this
- E.g. blastOff(4)

```
4 4 4 4
3 3 3
2 2
1
BLAST OFF
```

Blast Off Method

- Put your method here:

Blast Off Testing

- Put your main here:

Operators

- Arithmetic
  - + - * / %
- String concatenation
  - +
  - Implicit type conversion: `"HI" + 5`

Operators continued

- Increment and decrement
  - `++  --`
  - Side effects! Be careful!

```
int x = 10;
x++;
int y = x++ + 3;
int z = --y + x--;
```

Operators continued

- Comparison
  - < <= == != >= >

```
public static void main(String[] args) {
    Scanner input = new Scanner(System.in);
    String t;
    t = input.nextLine(); //As user, enter the String "Hi"
    String u = "H" + "i"; //Compiler premakes to "Hi"
    checkEquHi(t);
    checkEquHi(u);
}//main
public static void checkEquHi(String h){
    if (h == "Hi")
        System.out.println(h + " is equal to Hi");
    else
        System.out.println(h + " is NOT equal to Hi");
}//checkEquHi
```

Operators continued

- Creating complex Boolean expressions
  - ! && ||
  - Short circuit evaluation

Short Circuit Evaluation

1

```
int x = 2;
int y = 5;

if ((x <= 2) || (y == 6))
    System.out.println("Wow");
```

2

```
int x = 2;
int y = 5;

if ((x <= 2) && (y == 6))
    System.out.println("Wow");
```

3

```
int x = 2;
int y = 5;

if ((x > 2) && (y == 6))
    System.out.println("Wow");
```

4

```
int x = 2;
int y = 5;

if ((++x <= 2) && (y++ == 5))
    System.out.println("Wow");
```

Operators continued

- Assignment
  - =

- Compound assignment
  - <binary operator>=
  - E.g. += /=

```
x = 5;
x += 5;
x /= 2;
```

Representation of Information

```
int x = 9;
byte y = 12;
```

- Everything is finite binary

Representation of Information continued

3846

- Let's go back to Base-10
  - __ X 1000 + __ X 100 + __ X 10 + __ X 1
  - __ X $10^{\square}$ + __ X $10^{\square}$ + __ X $10^{\square}$ + __ X 10^0

___	___	___	___	___	___	___	___
10^7	10^6	10^5	10^4	10^3	10^2	10^1	10^0

Representation of Information continued

- Binary
  - Base-2
  - Two symbols: 0 and 1

- E.g. int x = 9;

… 2^4 2^3 2^2 2^1 2^0

- E.g. byte y = 42;

Representation of Information continued

- E.g. byte zebra = 67;

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

- E.g. byte giraffe = 260;

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

Representation of Information continued

- E.g. what is this?

```
0 1 1 0  1 0 0 0
```

Representation of Information continued

- Everything is binary
  - Integers
  - Real numbers
  - Characters
  - Strings
  - Window display

Bitwise Operators

```
byte x = 5;
byte y = 6;
```

- And: &
  - E.g. x & y
- Or: |
  - E.g. x | y
- Exclusive or: ^
  - E.g. x ^ y
- Complement (not): ~
  - E.g. ~x

Bitwise Operators

```
byte x = 5;
byte z = -8;
```

```
x: 0000 0101
z: 1111 1000
```

- Left shift fill with 0: <<
  - E.g. x << 3
- Right shift fill with sign: >>
  - E.g. z >> 2
- Right shift fill with 0: >>>
  - E.g. z >>> 3

Code in Java

- Write a Java method, isEven, to determine whether an integer is even, using a bitwise operator.

isEven Method

- Put your method here:

Code in Java

- Write a Java method to divide by 2, using a bitwise operator.

divBy2 Method

- Put your method here:

The End

Review

- Give two examples of types in Java.
- What is an implicit type conversion?
- What is a parameter? What is a local variable?
- If a method does not return anything, what is its return type?

Review

- Find all the syntax errors in the following Java code:

```
public int main(String args)
         Int z = 15
         system.print('Hi There')
         z = '   ' + z
         system.println(z)
} main
```

- What is the difference between an **int** and a **long**?

Review

- Write an infinite while loop that prints odd integers starting at 1
- Why are ints not a subset of floats?

Review

- Suppose we have

```
byte banana = 83;
```

- Where will banana be when your program is running?
- What is actually stored for banana?

Review

- What is the output of the following code? And what is the value of each variable when the code is done?

```
int a = 5;
int b = 6;
int c = 10;
System.out.println( ++b - c-- / a++ );
```

Review

- What is output of following Java code?

```
int a = 5;
int b = --a + 3;
int c = 4 * b++;
int d = 4 * b++;
System.out.println(a + " " + b + " " + c + " " + d);
```

- What is output of the following Java code?

```
int a = 28;
a = a << 1;
System.out.println(a);
```

```
int a = 28;
a = a << 2;
System.out.println(a);
```