girldevelopit

Beginning Java for Android
Session 1: Java—not just coffee

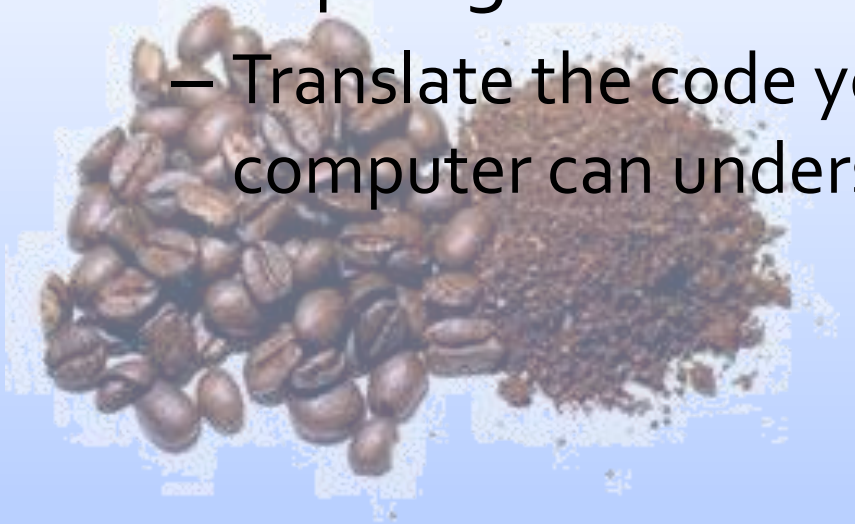
Izzy Johnston

izzycjohnston@gmail.com

@izzy_johnston

What is Java?

- Programming Language developed in 1995
- Platform independent
  - Can run on any environment once compiled
- Compiling? What's that?
  - Translate the code you wrote into something your computer can understand



How does it work?

- Object Oriented Programming
  - (or OOP to the initiated)
- Objects put together to form a program
- Objects contain
  - Data
    - Known as state
  - Behavior
    - Known as method

Think: Legos

Lego States

- Color
- Width
- Length
- Height
- Stackable from the Top
- Stackable from the Bottom

Lego Methods

- Stack on Top
- Stack Under
- Stack Perpendicular
- Stack Parallel
- Stack Staggered

What happens to objects?

- Small Classes
  - Code that create objects
- Big Classes
  - Code that combine objects made in small classes
- Library
  - 10,000s of methods made by people
  - 1000s of classes made by people
  - API (Application Programming Interface)
    - Rules of the library

Legos: Part 2

Lego Design

Designing own lego creation like creating own class

NOTE:

BOTH use the same Legos

Like using methods from from Java Library

They did not mold and cast new lego shapes.

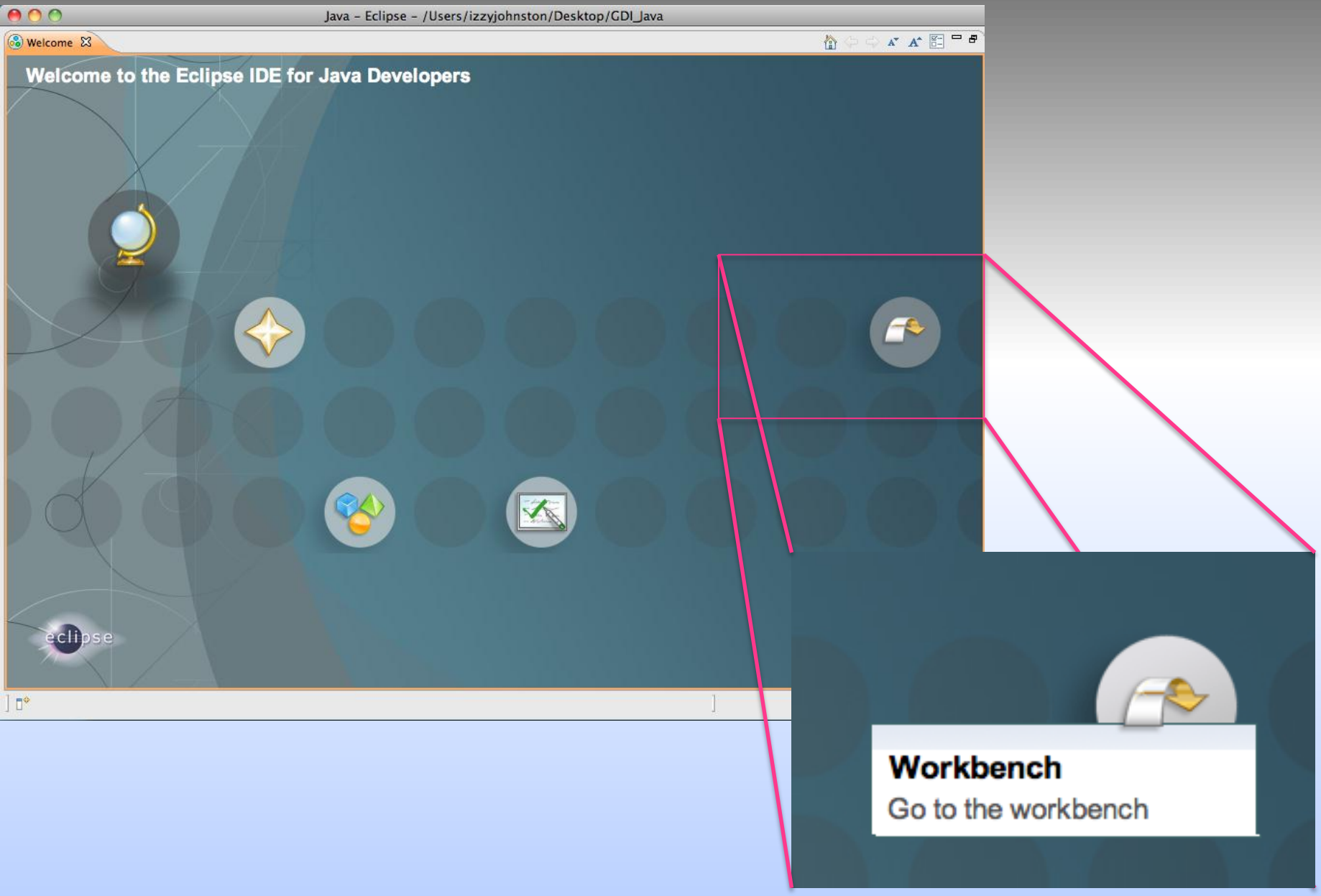
That would be like writing own method

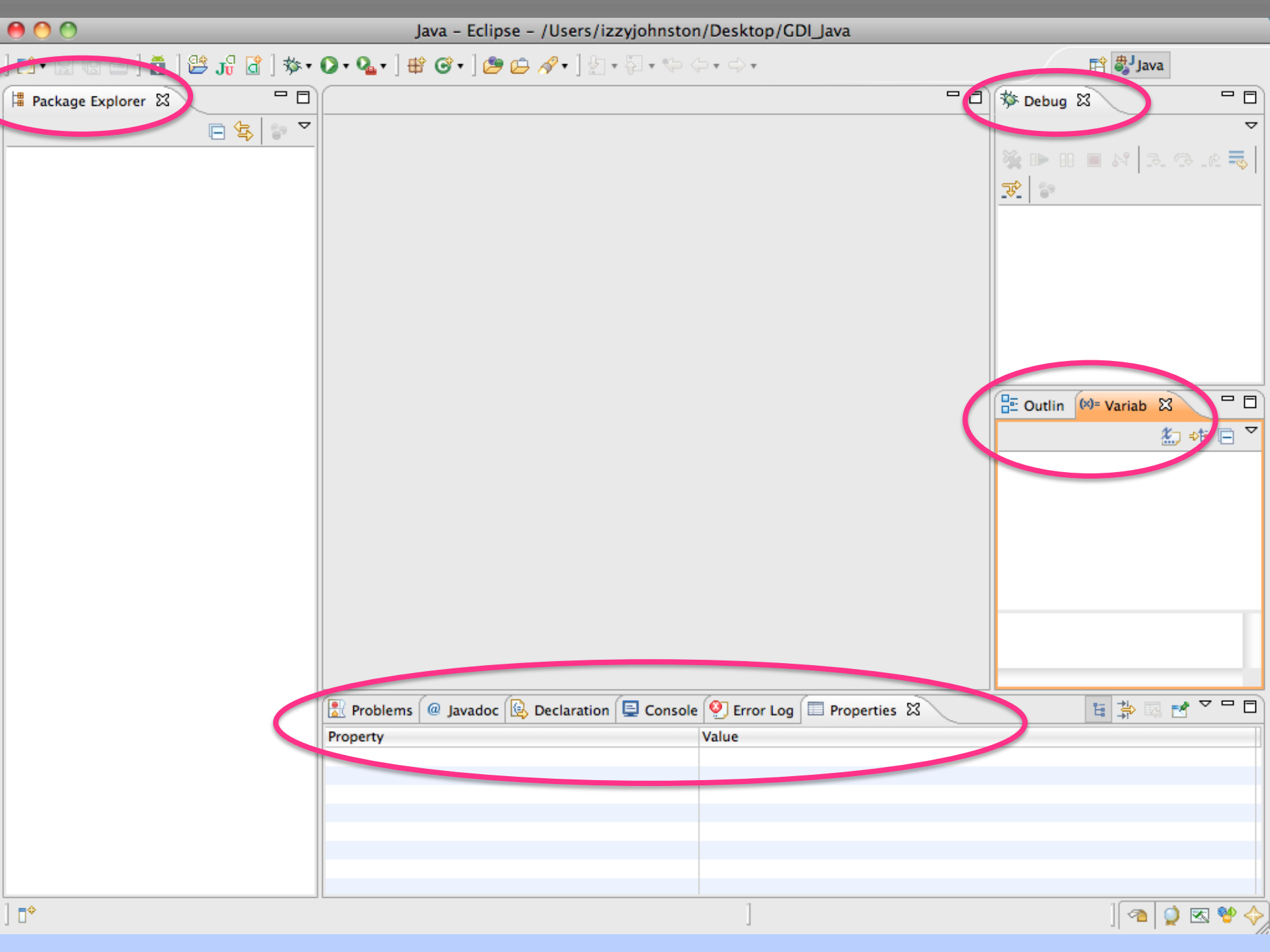
Lego Library

Store bought set is like using a class from the Java Library

How to get started

- Eclipse
  - Software Development Environment
  - Open Source
  - Mostly for Java, but you can use it for almost any language
- Android SDK
  - Software Development Kit
  - Tools to develop software on a specific platform
  - Contains APIs and special libraries just for Android


Java – Eclipse – /Users/izzyjohnston/Desktop/GDI_Java
Welcome
Welcome to the Eclipse IDE for Java Developers
eclipse
Workbench
Go to the workbench


Java – Eclipse – /Users/izzyjohnston/Desktop/GDI_Java
Java
Package Explorer
Debug
Outlin
Variab
Problems
Javadoc
Declaration
Console
Error Log
Properties
Property
Value

Variables

- Chunk of content
  - Data (integers, strings) or Objects (Legos)
- Can be local
  - Only used in one method
- Can be global
  - Used anywhere in the whole program

Types of Variables

- Integers
  - int, short, long
- Decimals
  - float, double
- Booleans
  - can be either true or false
- Chars
  - One character
- Strings
  - Any number of characters

Using Variables

- "Declaring"
  - Saying that something is a variable
  - int x;
  - String y;
  - float z;
- "Assigning"
  - Giving variables a value
  - x=26; OR int x=26;
  - y="Hi, there"; OR String y = "Hi, there";
  - z= 6.01; OR float z= 6.01;

Mathematical Operators

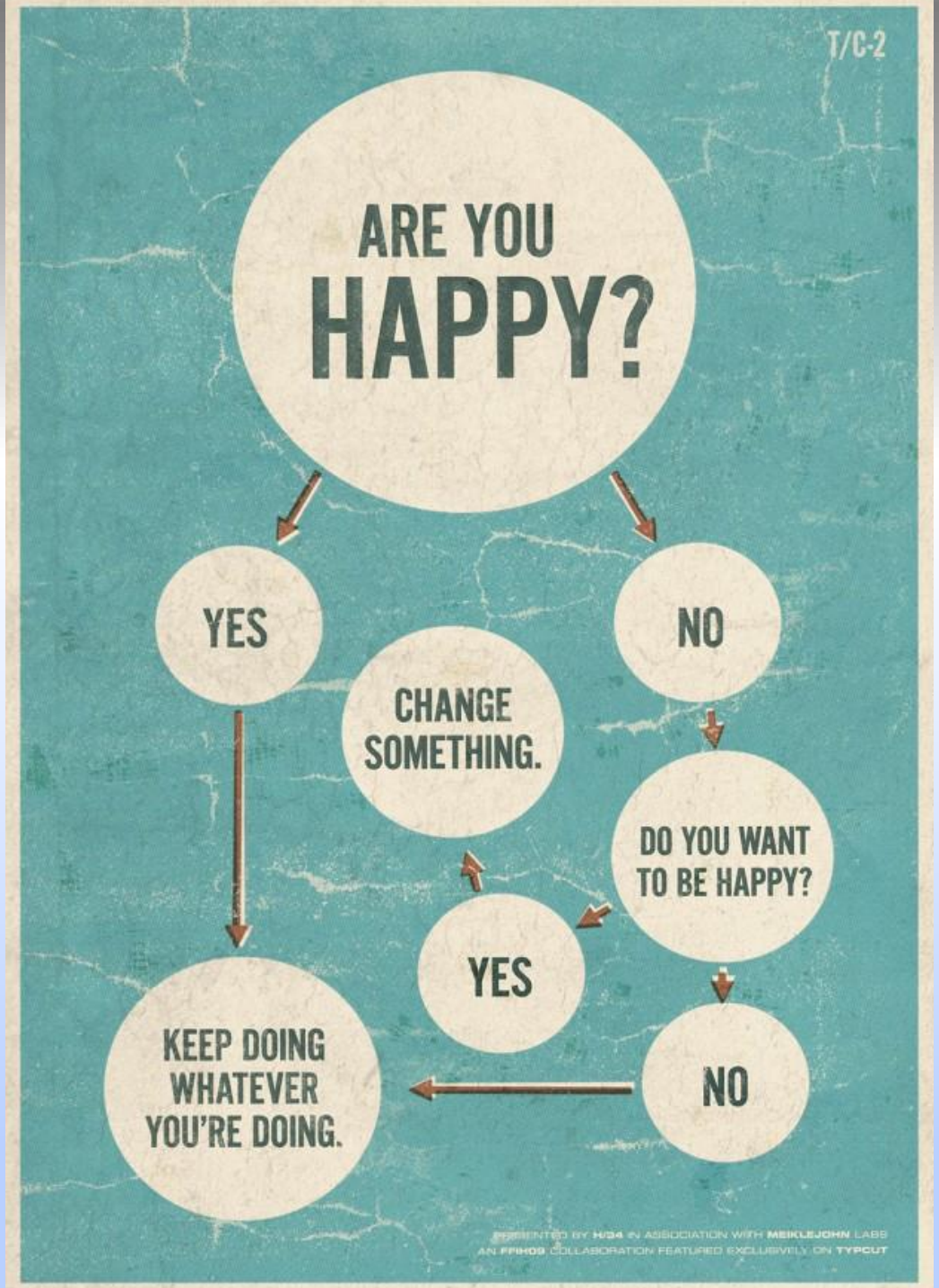
Symbol	Meaning	Example
+	Addition	x + 2 = 4
-	Subtraction	x - 2= 0
*	Multiplication	x * 2 = 4
/	Division	x / 2 = 1
%	Modulus (Remainder)	5 % x = 1
++	Increment	x ++ = 3
--	Decrement	x -- = 1

*For all examples, int x=2;

Comparison Operators

Symbol	Meaning	Example
==	Equals?	x == y
!=	Does not equal?	x != y
>	Is greater than?	x > y
<	Is less than?	x < y
&&	AND	x<5 && y >5
||	OR	x<5 || $y >5)
!	NOT	!(x==y)

*For all examples, int x=2 and int y=3


T/C-2
ARE YOU
HAPPY?
YES
NO
CHANGE
SOMETHING.
DO YOU WANT
TO BE HAPPY?
YES
KEEP DOING
WHATEVER
YOU'RE DOING.
NO
PRESENTED BY H/34 IN ASSOCIATION WITH MEIKLEJOHN LABS
AN FFIH09 COLLABORATION FEATURED EXCLUSIVELY ON TYPCUT

If/Else Statements

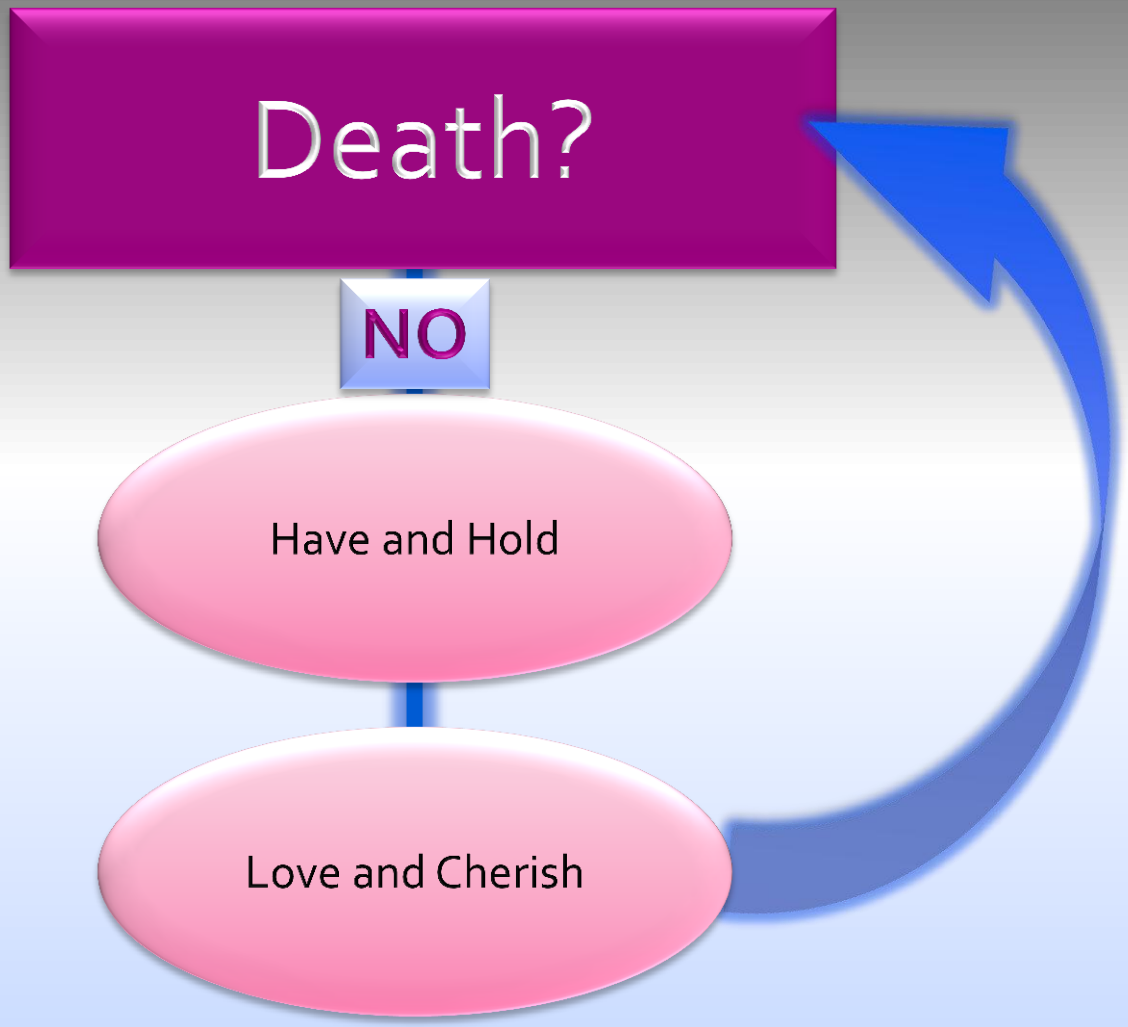
```
if (some parameter){
    do this;
}
else {
  do that;
}
```

If/Else Statements, cont.

```
int a=2;
int b=3;
if (a>b){
    System.out.println (a);
}
else {
  System.out.println (b);
}
```

If/Else Statements, cont.

```
int a=5;
int b=3;
if (a>b){
    System.out.println (a);
}
else {
  System.out.println (b);
}
```


Death?
NO
Have and Hold
Love and Cherish

While Loops

```
int a=0;
while (a<10){
    System.out.println (a);
    a++;
    /* OR a=a+1; */
}
```

Arrays

- Holds multiple variables with index numbers instead of unique name

```
String [] daysOfWeek;
daysOfWeek[0] = "Sunday";
daysOfWeek[1] = "Monday";
daysOfWeek[2] = "Tuesday";
daysOfWeek[3] = "Wednesday";
daysOfWeek[4] = "Thursday";
daysOfWeek[5] = "Friday";
daysOfWeek[6] = "Saturday";
```

For Loop

- Like the while loop but with counting

```
for (int i=0; i<7; i++){
   System.out.println (daysOfWeek[i]+ ", ");
}
```

For Loop…a more practical example

```
int [] randomNumbers;


for (int i=0; i<100; i++){
    randomNumbers[i] =
    (int)(Math.random()*10)+1;
}
```

Making a Class

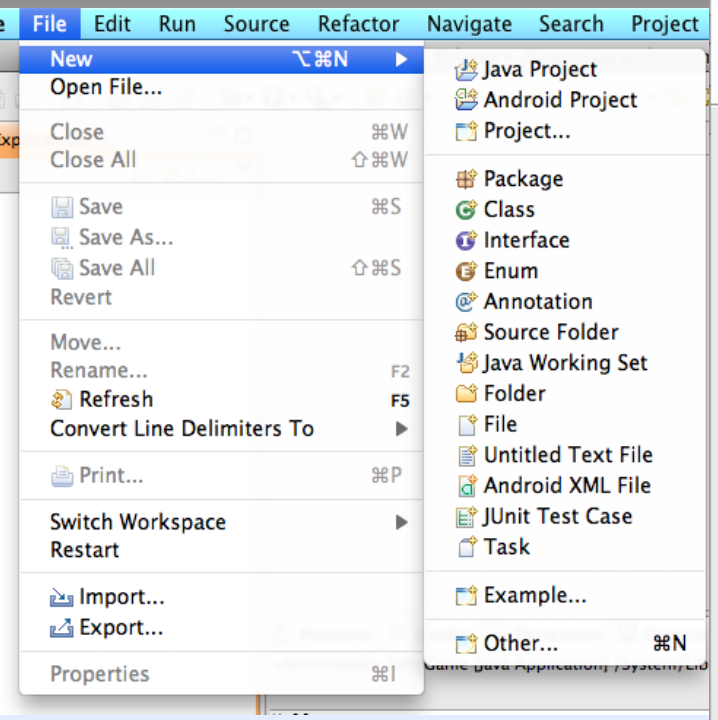
Every application has at least one class

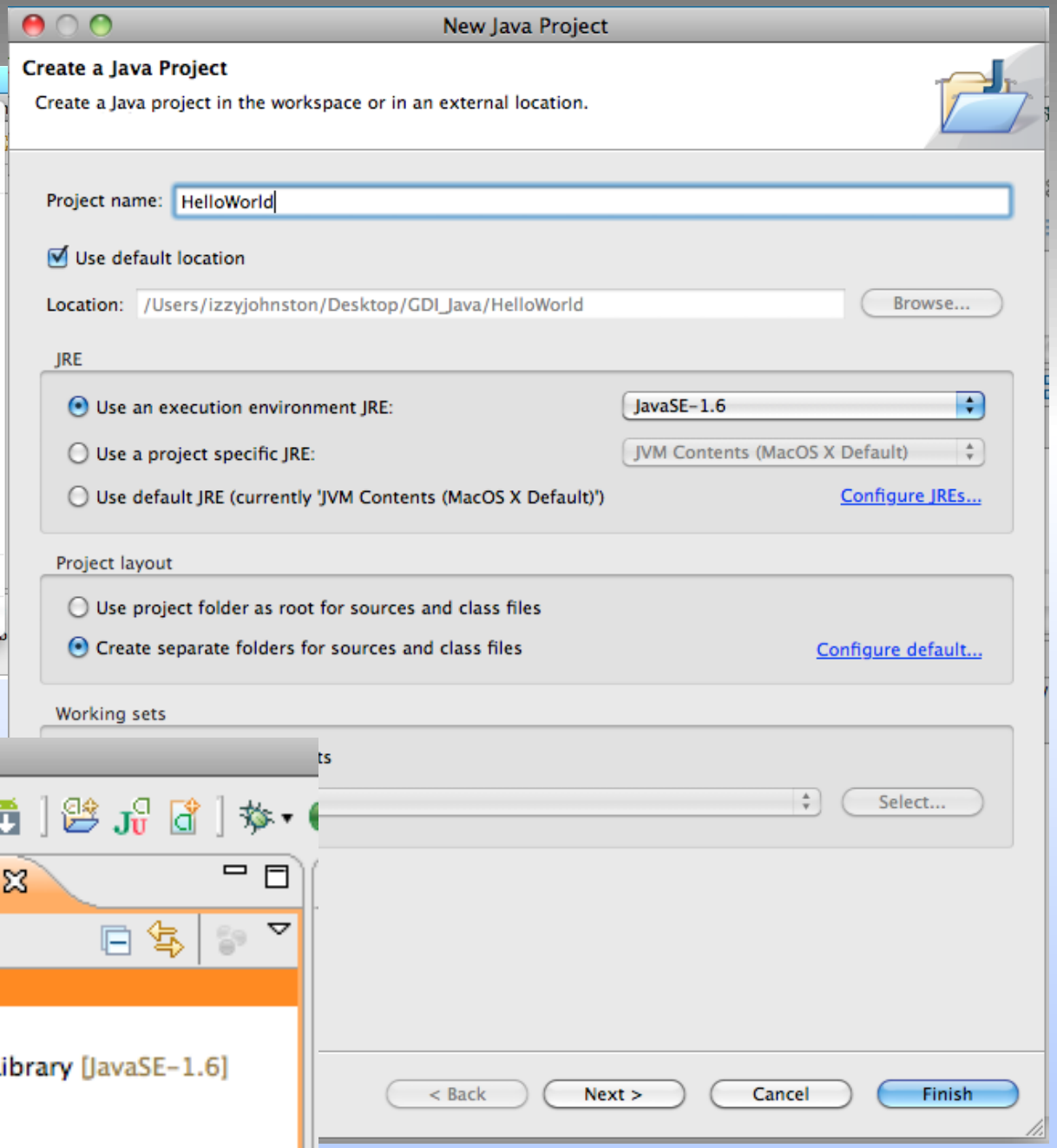
```
public class ClassName{

    class body;

}
```

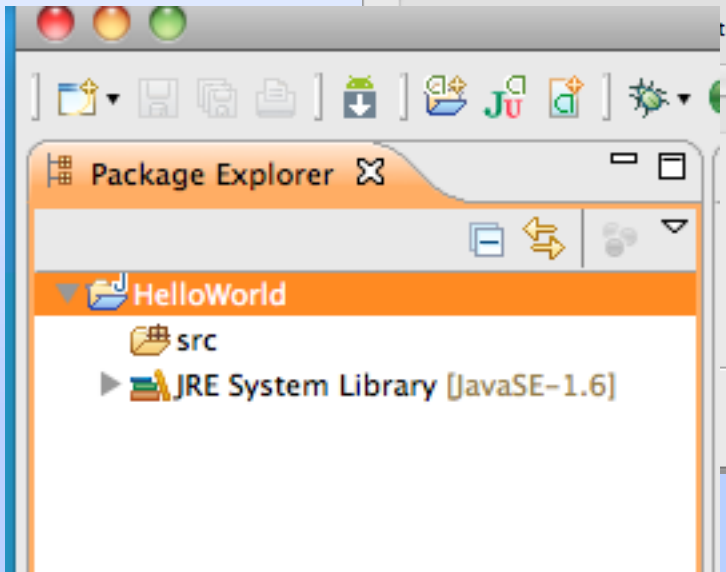
Making a Method

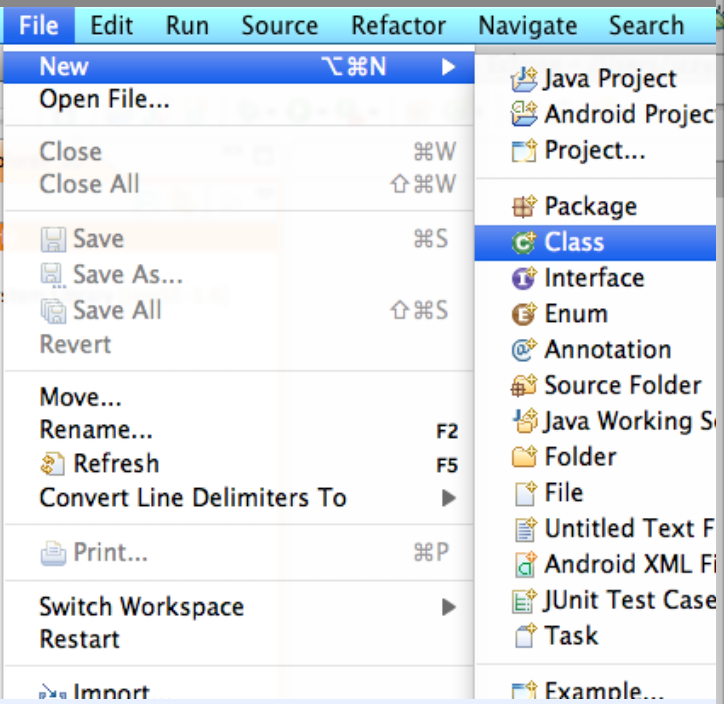
Every application has at least one method

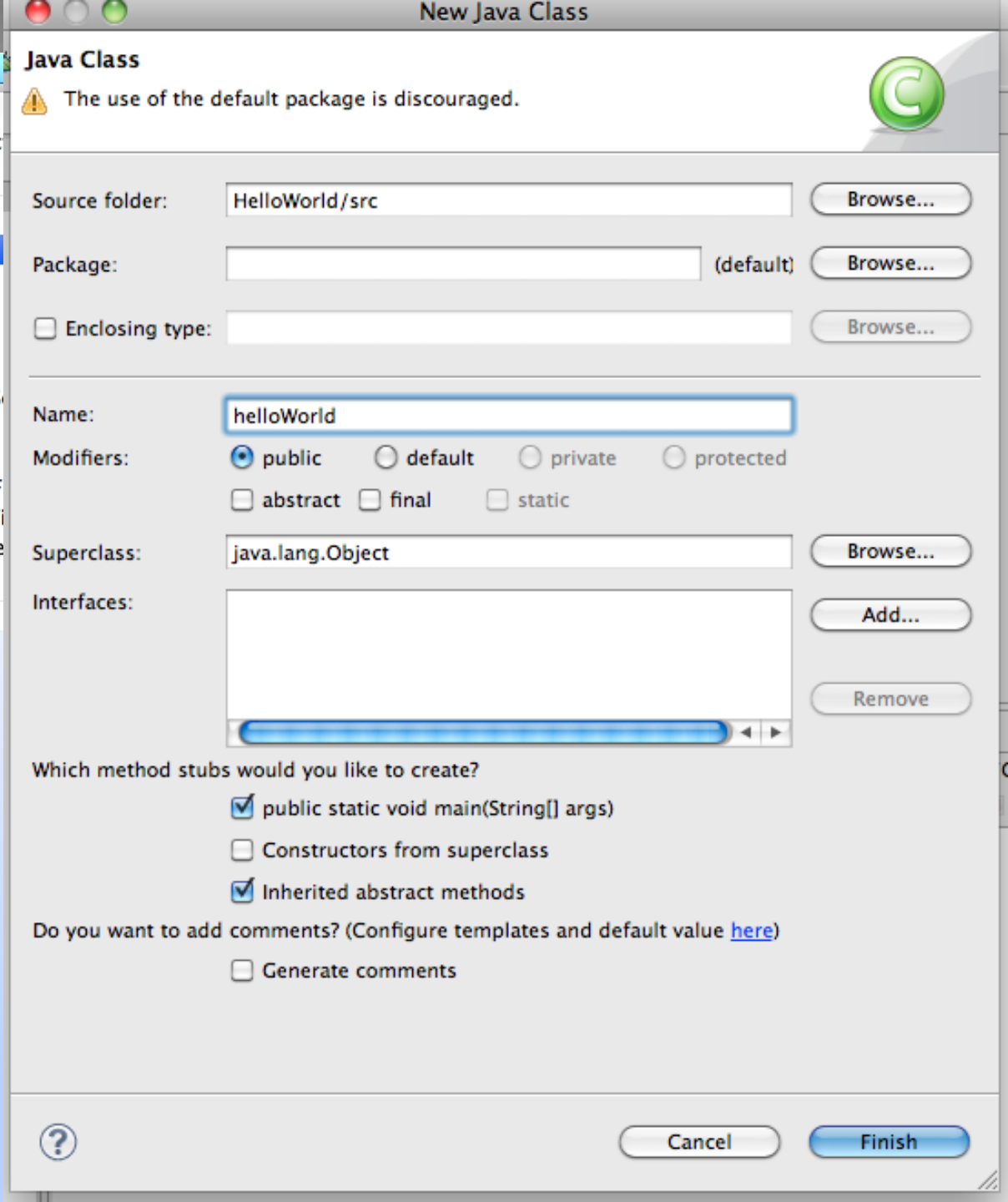
```
public static void methodName (arguments) {

    method body;

}
```

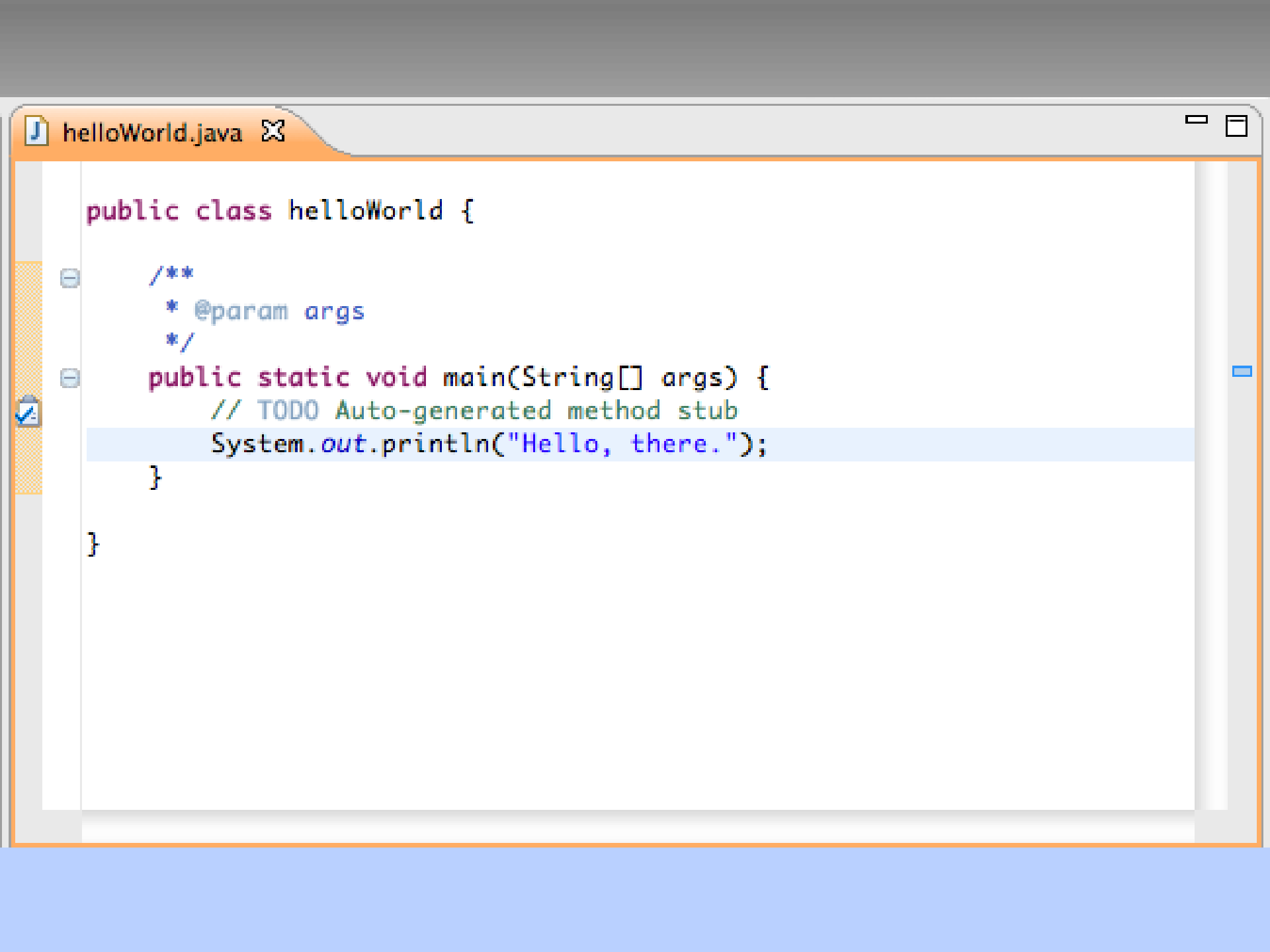

File
Edit
Run
Source
Refactor
Navigate
Search
Project
New ⌥⌘N
Open File...
Close ⌘W
Close All ⇧⌘W
Save ⌘S
Save As...
Save All ⇧⌘S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... ⌘P
Switch Workspace
Restart
Import...
Export...
Properties ⌘I
Java Project
Android Project
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working Set
Folder
File
Untitled Text File
Android XML File
JUnit Test Case
Task
Example...
Other... ⌘N


New Java Project
Create a Java Project
Create a Java project in the workspace or in an external location.
Project name: HelloWorld
Use default location
Location: /Users/izzyjohnston/Desktop/GDI_Java/HelloWorld
Browse...
JRE
Use an execution environment JRE: JavaSE-1.6
Use a project specific JRE: JVM Contents (MacOS X Default)
Use default JRE (currently 'JVM Contents (MacOS X Default)')
Configure JREs...
Project layout
Use project folder as root for sources and class files
Create separate folders for sources and class files
Configure default...
Working sets
Select...
< Back
Next >
Cancel
Finish


Package Explorer
HelloWorld
src
JRE System Library [JavaSE-1.6]


File
Edit
Run
Source
Refactor
Navigate
Search
New ⌥⌘N
Open File...
Close ⌘W
Close All ⇧⌘W
Save ⌘S
Save As...
Save All ⇧⌘S
Revert
Move...
Rename... F2
Refresh F5
Convert Line Delimiters To
Print... ⌘P
Switch Workspace
Restart
Import
Java Project
Android Projec
Project...
Package
Class
Interface
Enum
Annotation
Source Folder
Java Working S
Folder
File
Untitled Text F
Android XML Fi
JUnit Test Case
Task
Example...


New Java Class
Java Class
The use of the default package is discouraged.
Source folder:
HelloWorld/src
Browse...
Package:
(default)
Browse...
Enclosing type:
Browse...
Name:
helloWorld
Modifiers:
public
default
private
protected
abstract
final
static
Superclass:
java.lang.Object
Browse...
Interfaces:
Add...
Remove
Which method stubs would you like to create?
public static void main(String[] args)
Constructors from superclass
Inherited abstract methods
Do you want to add comments? (Configure templates and default value here)
Generate comments
Cancel
Finish


helloWorld.java
public class helloWorld {
/**
* @param args
*/
public static void main(String[] args) {
// TODO Auto-generated method stub
System.out.println("Hello, there.");
}
}

Math Class

Name	What it does	Parameters	Example	Result
absolute value	Absolute Value of your number	One Number	Math.abs(-35);	35
random	Generates number between 0.0 and 1.0	None	Math.random()* 6;	Number between 0 and 5
Maximum	Returns bigger of two number	Two numbers	Math.max(35, 72);	72
Minimum	Returns smaller of two numbers	Two numbers	Math.min(35, 72);	35
Power	Returns first number to power of second	Two numbers	Math.pow(10, 3);	1000
Square Root	Returns Square root of number	One number	Math.sqrt(16);	4

Lotto Game

Goal: Guess the number randomly generated by the computer

Step 1. Create new Java Project named LottoGame

Step 2: Create new Java Class name lottoGame

Lotto Game, cont.

- Importing—letting the program we want to use a class from the Java API
- Written above the class declaration

```
import java.lang.Math;
import java.util.Scanner;
```

Lotto Game, cont.

Declare three variables.

Inside method.

```
int guess;
int answer;
Scanner sc = new Scanner(System.in);
```

Lotto Game, cont.

```
System.out.println("Guess a number between 1 and 6");

guess =  sc.nextInt();

answer = (int)(Math.random()*6+1);

if (guess==answer) {
    System.out.println("Congratulations! You guessed right!");
}
else {
    System.out.println("Sorry, that wasn't right. The correct answer
    was " + answer+".");
}
```

String Operators

Name	What it does	Parameters	Example	Result
Compare To	Compares two strings	Compares two strings	String.compareTo("cat", "cats");	0
Length	Tells you length of a string	One string	String.length("pickles");	7
toString	Turns a number into a string	Any number (int, float, double)	Integer.toString(72);	"72"
Parse	Turns string into number	A string known to only contain numbers	Integer.parseInt("72");	72
Replace	Replaces a character with another	Two characters	"Happy".replace('a', 'o');	Hoppy
toCharArray	Splits string into array of characters	None	char myArray[]="Hey".toCharArray();	myArray[0]='H'; myArray[1]='e'; myArray[2]='y';

Homework: String Scrambler

- Using the String operators and a for loop
  - Build an application that prints the letters in a word backward
  - Hints:
    - Get a String from the scanner using .next()
    - Split the string into an array.
    - Print out the array in backward order
- Bonus:
  - Try to use Math.random() to print out the array in random order
- There are completed versions of HelloWorld, LottoGame, and StringScrambler in the practice files

Questions?