

Java level-I

👍 Useful Icons

😊 Guaranteed course, if you think you have not learned Python after this class finished, you can always join my next class section for free.

Table of Contents

- [Table of Contents](#)
- [Familiar with your keyboard](#)
- [Getting Started](#)
- [Using Markdown](#)
- [Basic Git Command](#)
- [Learn from mistake](#)
- [How to turn in the homework](#)
- [Java Document](#)
- [open Eclipse](#)
- [Syntax basic](#)
- [Print String](#)
- [Using Variables](#)
- [Comments](#)
- [Data type](#)
- [Operator](#)
- [if-else](#)
- [Loop](#)
- [Simple Math](#)
- [Array](#)
- [ArrayList](#)
- [Switch](#)
- [Data Structure](#)
- [Exception](#)
- [OOP](#)
- [File Access](#)
- [All topics](#)
- [Applications](#)

- [String.format\(\)](#)

- [References](#)

Familiar with your keyboard

1. [Share Keyboard document](#)

[Table of Contents](#)

Getting Started

- Install JDK

Google search: jdk 8 download

Search result: Java SE Development Kit 8 - Downloads - Oracle

[Java Compiler Download](#)

Java SE Downloads

Java Platform, Standard Edition

Oracle JDK



[JDK Download](#)



[Documentation Download](#)

File: jdk-14.0.1_osx-x64_bin.dmg

- Install Eclipse

Google search: eclipse for java

Search result: Eclipse Downloads | The Eclipse Foundation

Eclipse Download



Get Eclipse IDE 2020-03

Install your favorite desktop IDE packages.

[Download 64 bit](#)

File: eclipse-inst-mac64.dmg

- Install Git

Google search: git downloads

Search result: Git - Downloads

[Git Download](#)

Downloading Git



You are downloading the latest **(2.29.2) 64-bit** version of **Git for Windows**. This is the most recent [maintained build](#). It was released **9 days ago**, on 2020-12-08.

[Click here to download manually](#)

Other Git for Windows downloads

[Git for Windows Setup](#)

[32-bit Git for Windows Setup.](#)

[64-bit Git for Windows Setup.](#)

[Git for Windows Portable \("thumbdrive edition"\)](#)

[32-bit Git for Windows Portable.](#)

[64-bit Git for Windows Portable.](#)

The current source code release is version **2.29.2**. If you want the newer version, you can build it from [the source code](#).

File: Git-2.29.2.3-64-bit.exe

For MacOS:

```
brew install git  
ls -la /usr/local  
sudo chown -R wangqianjiang:wheel /usr/local  
brew link git
```

- installation check

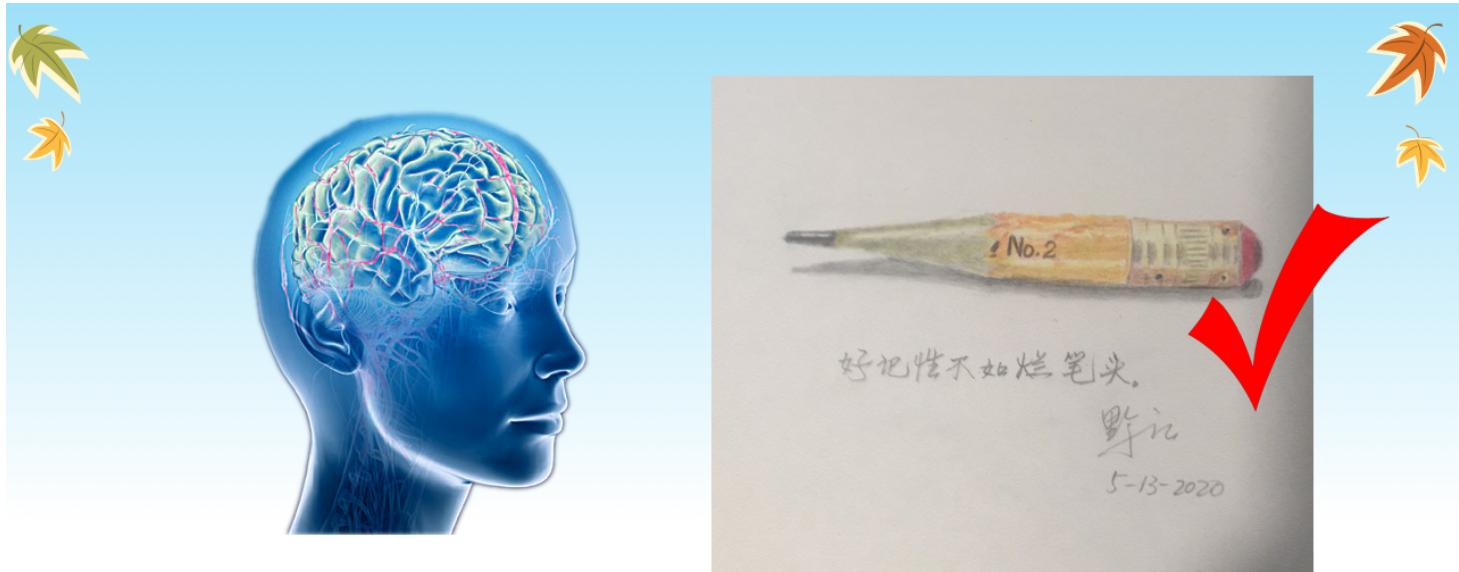
Open Dos Command Window:

```
java -version  
javac -version  
git --version
```

Table of Contents

Using Markdown

Markdown md文件的制作，制作课堂笔记



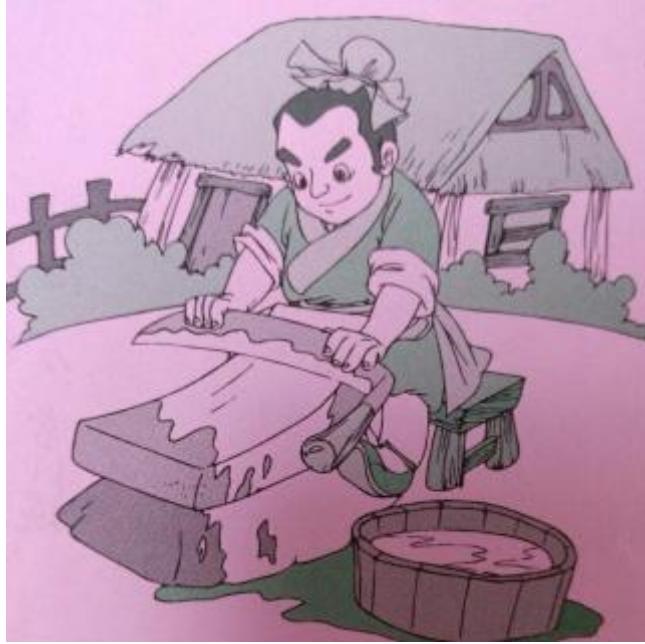
Good memory cannot be as good as Markdown



- add Markdown Extension
- 显示标题，子标题 #, ##

- 显示 tablet *, 1
- 显示命令行
- 显示图形
- 显示链接

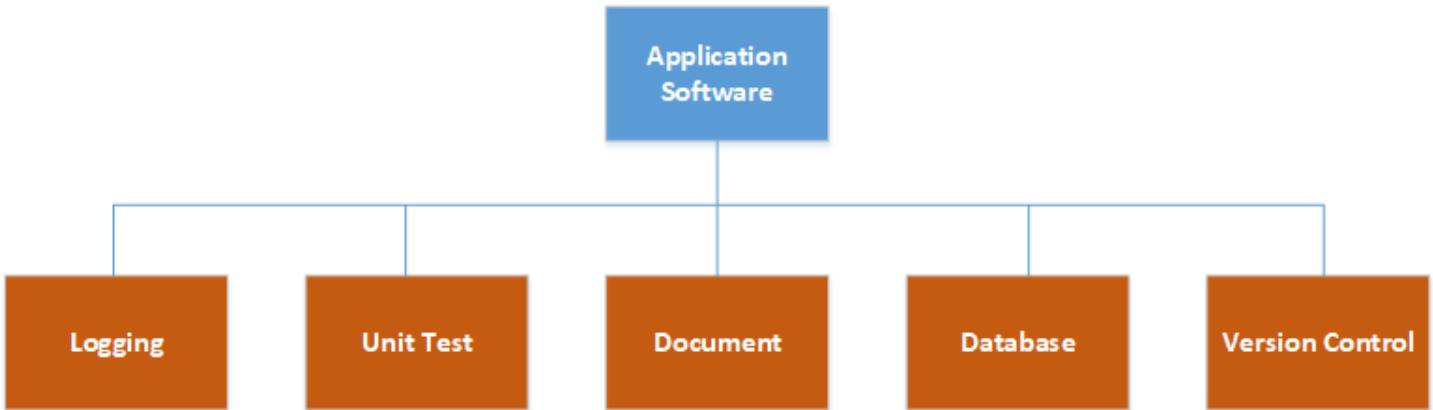
Sharpen your knife will not slowdown your work!



[Markdown Cheat Sheet](#)

[Table of Contents](#)

Basic Git Command

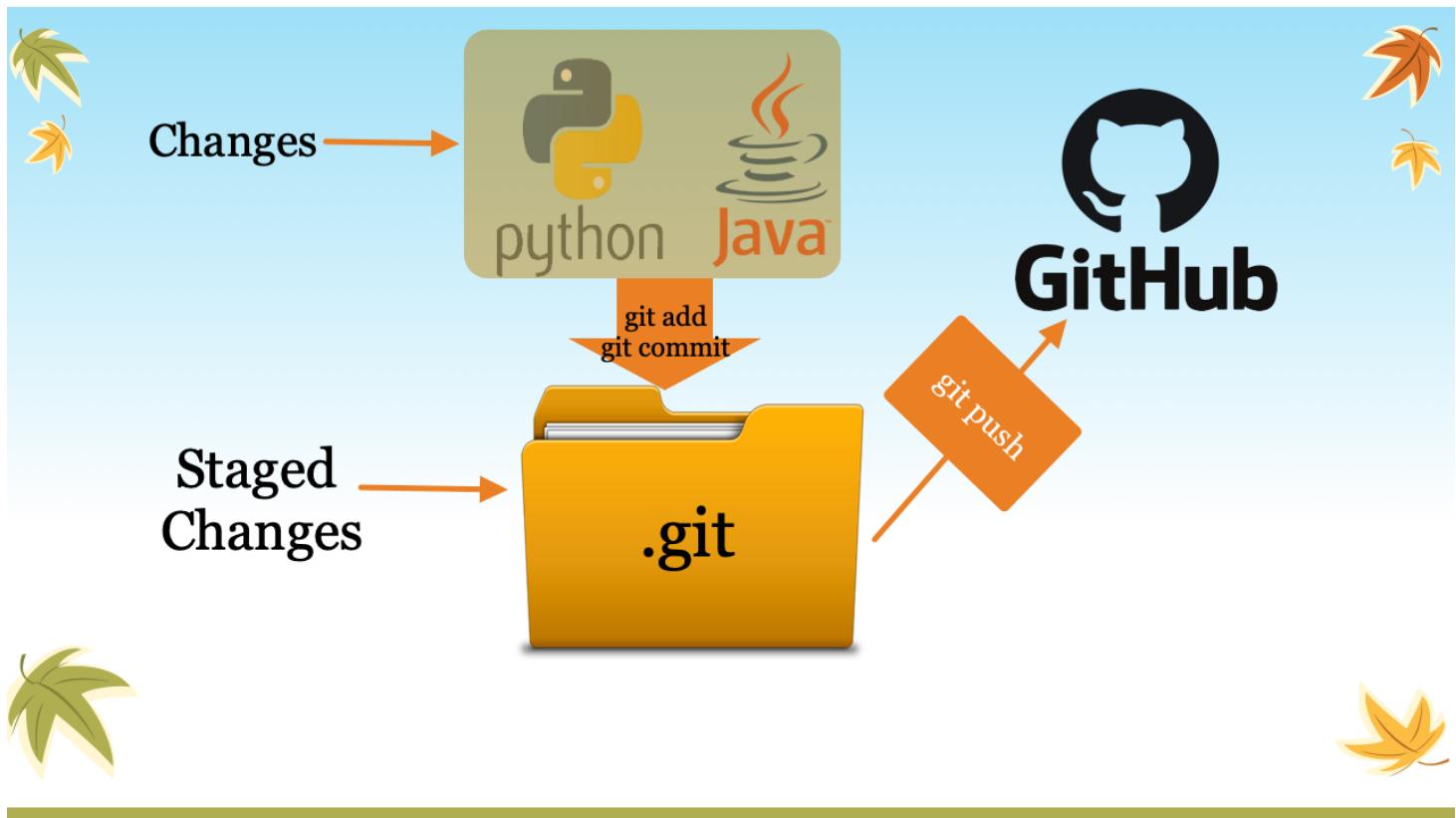


- Check Git Installation

```
git --version
```

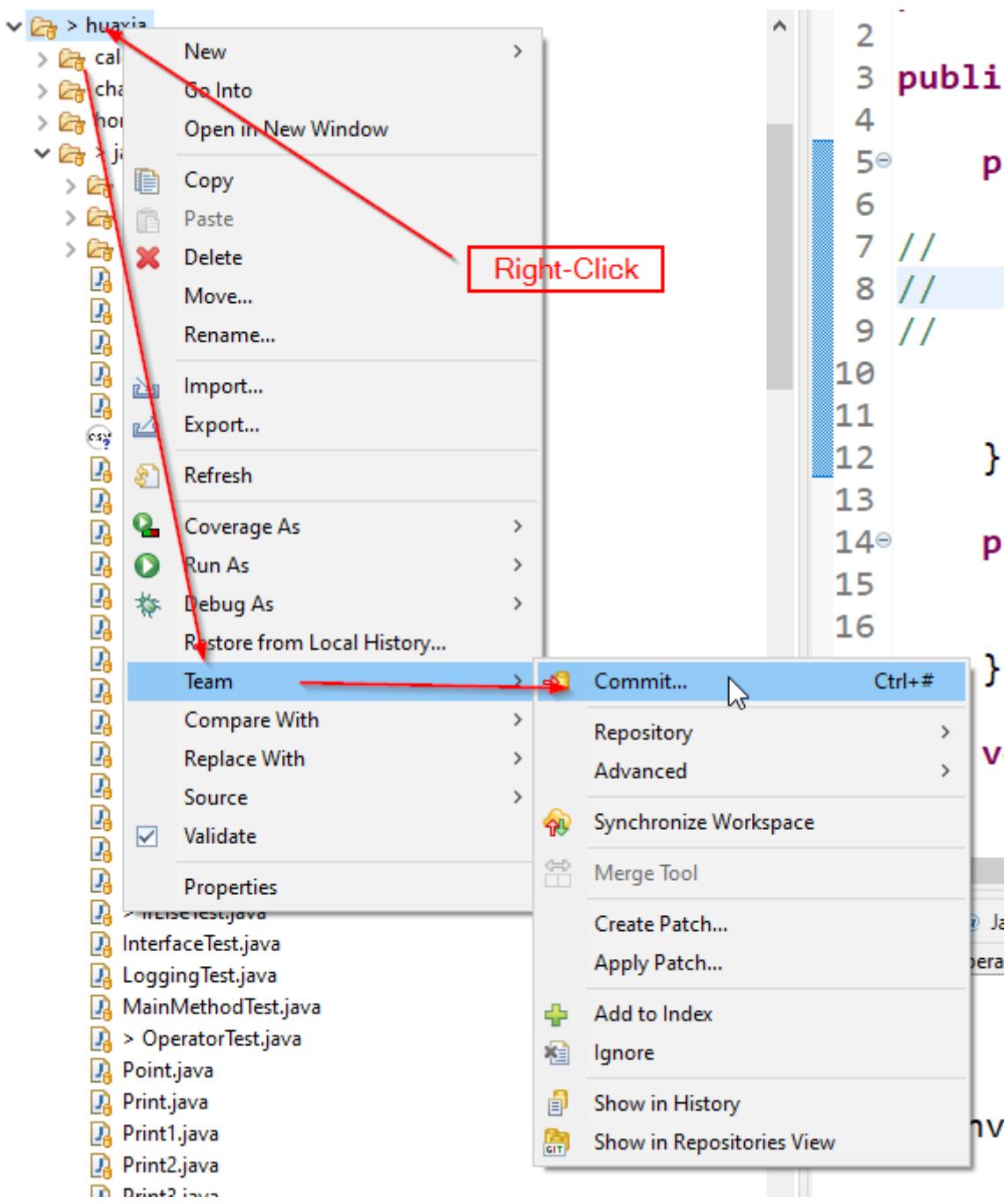
- introduce git repository basic

```
git init  
git --version  
git config user.name "<username>"  
git config user.email "<useremail>"  
git status  
git add .  
git log --oneline  
git branch  
git commit -m "initial commit"  
git push
```



test some code change

- install EGit (Git for Eclipse)



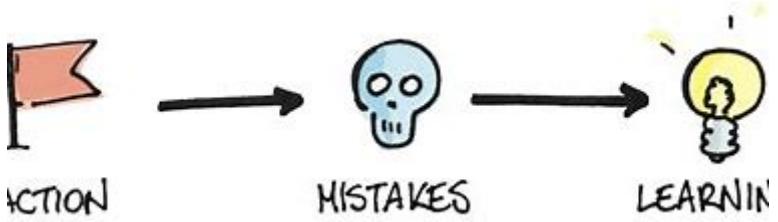
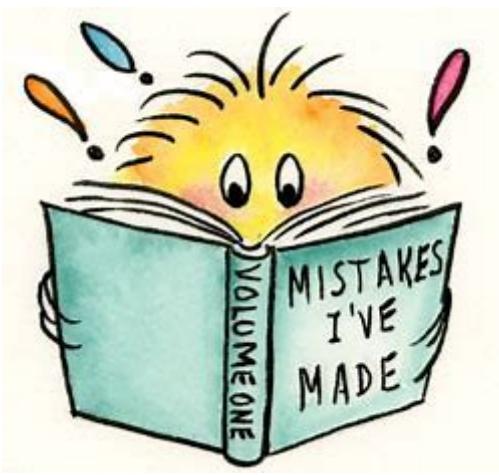
[EGit website](#)

help > install new software >

- Create a GitHub account, try upload some file

[GitHub Website](#)

Learn from mistake



Learn from other's mistake is better than from my own.

Why? you pay price on your own mistake, but if it is other's mistake, they pay the price for you.

Issue 1: he stucked the installation on wrong folder.

Solution: delete the folder in wrong place, recreate on the right place which is

Macintosh HD ▶ Users ▶ wangqianjiang ▶ workspace ▶ python

where wangqianjiang is the login user's name.

- show the homework /grace (Very good example except missing image files that she used in the notes)
- show the homework /angela (We should use what we have learned)
- talking about GitHub which is not show stopper, does not stop you learning Java, it is better to have

问学生要他们在自己周围看到的第一个物体，抽象出来制作一个class

学生: cup, computer

制作一个Cup实体类

refer: com.huaxia.blackjack.Cup

把Computer作为作业让学生模仿制作。

Table of Contents

How to turn in the homework

[Java-I Homework](<https://github.com/jwang1122/java1homework>)

- homework demo
 - HomeworkDemo.java
- make git pull for java repository
 - rename /workspace/java > /workspace/myjava
 - git clone <https://github.com/jwang1122/java.git>
 - help > Eclipse Marketplace > Find:git > Git integration

Table of Contents

Java Document

It is very important, you can learn Java all by yourself from Java document.

Google search: java document download

Search result: Java SE - Downloads | Oracle Technology Network | Oracle

[Java Document Download](#)

 Document download page

File: jdk-14.0.1_doc-all.zip

[JSE 8 Online API Document](#)

All classes > System > Field Detail > out

let student find scan

It is not easy to use for big project

- Generate Javadoc

Google Search: java 8 api document

[Java api document](#)

Packages

java.applet
java.awt
java.awt.color
java.awt.datatransfer

StreamSupport
StreamTokenizer
StrictMath
String
StringBuffer
StringBufferInputStream
StringBuilder
StringCharacterIterator
StringContent
StringHolder
StringIndexOutOfBoundsException
StringJoiner
StringMonitor
StringMonitorMBean
StringNameHelper
StringReader
StringRefAddr
StringSelection
StringSeqHelper
StringSeqHolder
StringTokenizer
StringValueExp
StringValueHelper
StringWriter
Stroke
StrokeBorder
Struct

compact1, compact2, compact3

java.lang

Class String

java.lang.Object
java.lang.String

All Implemented Interfaces:

Serializable, CharSequence, Comparable<String>

```
public final class String
extends Object
implements Serializable, Comparable<String>, CharSequence
```

The `String` class represents character strings. All string literals in Java programs, such as "abc", are implemented as instances of this class.

Strings are constant; their values cannot be changed after they are created. String buffers support mutable strings. Because `String` objects are immutable they can be shared. For example:

```
String str = "abc";
```

is equivalent to:

```
char data[] = {'a', 'b', 'c'};
String str = new String(data);
```

Project > Generate Javadoc... >

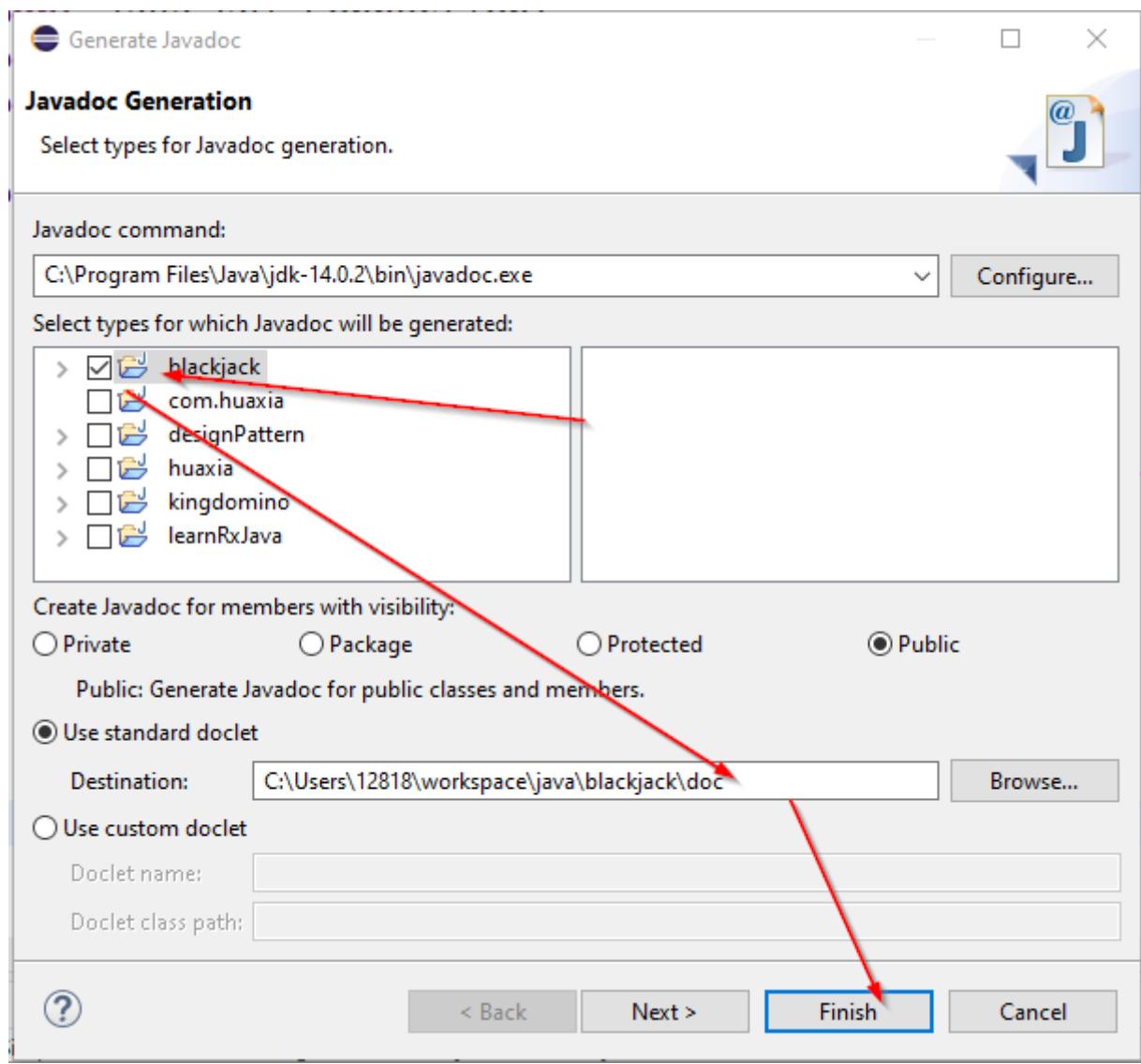
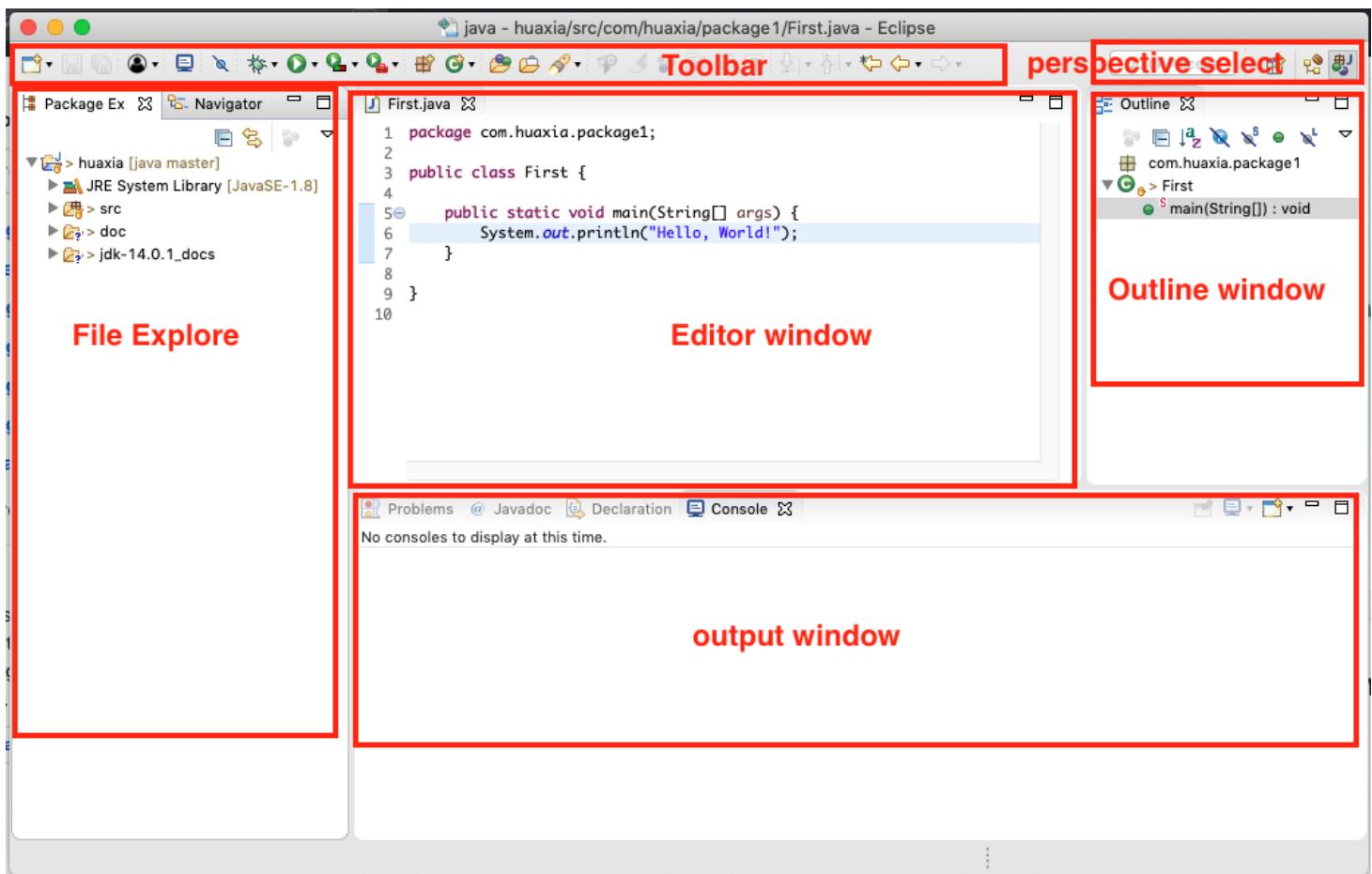


Table of Contents

open Eclipse

getting formillar with Eclipse IDE



Open File explore>create a workspace\java

git init

Run Eclipse icon > close Welcome window > File > new project > java project > huaxia > next

Project name: huaxia > [Finish] > Open Perspective

explain little bit about perspective > close Task List window

(Window > show view > navigator)

click project name > click package icon > com.huaxia.package1

click class icon > Name: First > check main() > [finish]

```
package com.huaxia.package1;
```

```
public class First {
```

```
    public static void main(String[] args) {
        System.out.println("Hello, World!");
```

```
}
```

```
}
```

```
Scanner scanner = new Scanner(System.in);
String name = scanner.nextLine();
System.out.println("Hello," + name);
scanner.close();
```

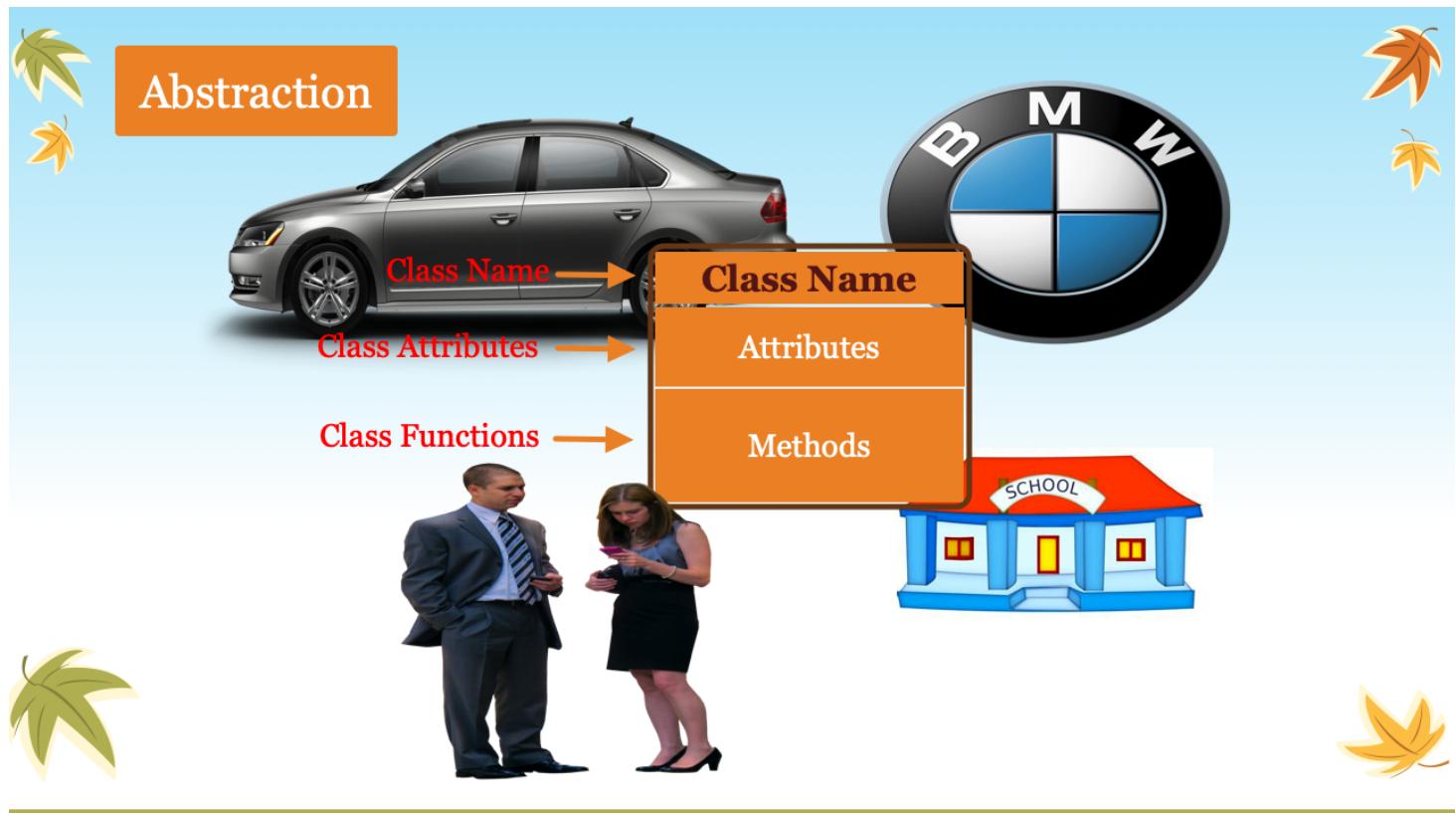
try to change the format

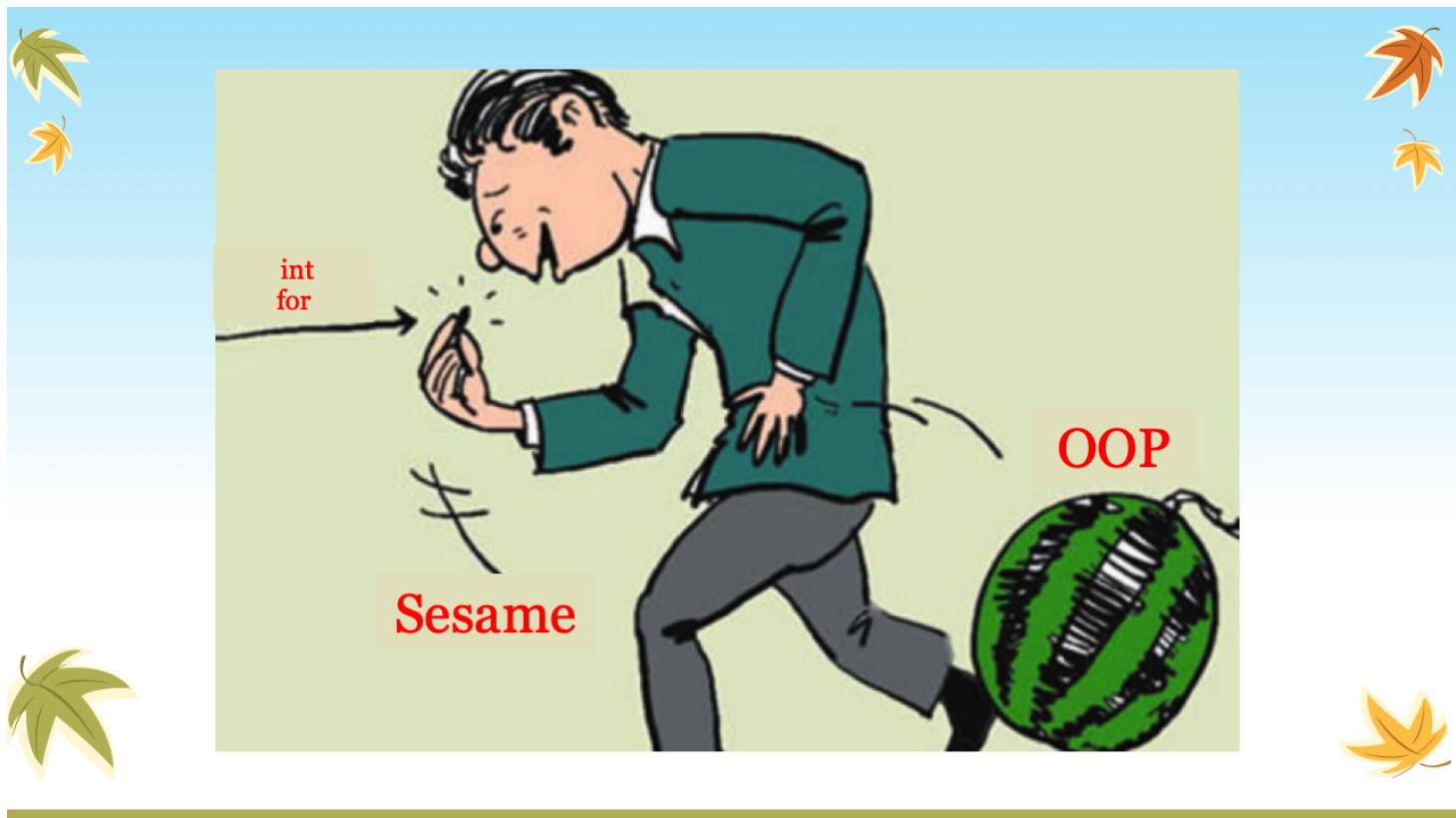
try to remove ;

try to create variable without data type

Table of Contents

Syntax basic





- Hello.java; create Java code from new class (key words, modifier, (), {})

Java Language Keywords

Here is a list of keywords in the Java programming language. You cannot use any of the following as identifiers in your programs. The keywords `const` and `goto` are reserved, even though they are not currently used. `true`, `false`, and `null` might seem like keywords, but they are actually literals; you cannot use them as identifiers in your programs.

<code>abstract</code>	<code>continue</code>	<code>for</code>	<code>new</code>	<code>switch</code>
<code>assert^{***}</code>	<code>default</code>	<code>goto[*]</code>	<code>package</code>	<code>synchronized</code>
<code>boolean</code>	<code>do</code>	<code>if</code>	<code>private</code>	<code>this</code>
<code>break</code>	<code>double</code>	<code>implements</code>	<code>protected</code>	<code>throw</code>
<code>byte</code>	<code>else</code>	<code>import</code>	<code>public</code>	<code>throws</code>
<code>case</code>	<code>enum^{****}</code>	<code>instanceof</code>	<code>return</code>	<code>transient</code>
<code>catch</code>	<code>extends</code>	<code>int</code>	<code>short</code>	<code>try</code>
<code>char</code>	<code>final</code>	<code>interface</code>	<code>static</code>	<code>void</code>
<code>class</code>	<code>finally</code>	<code>long</code>	<code>strictfp^{**}</code>	<code>volatile</code>
<code>const[*]</code>	<code>float</code>	<code>native</code>	<code>super</code>	<code>while</code>

* not used

** added in 1.2

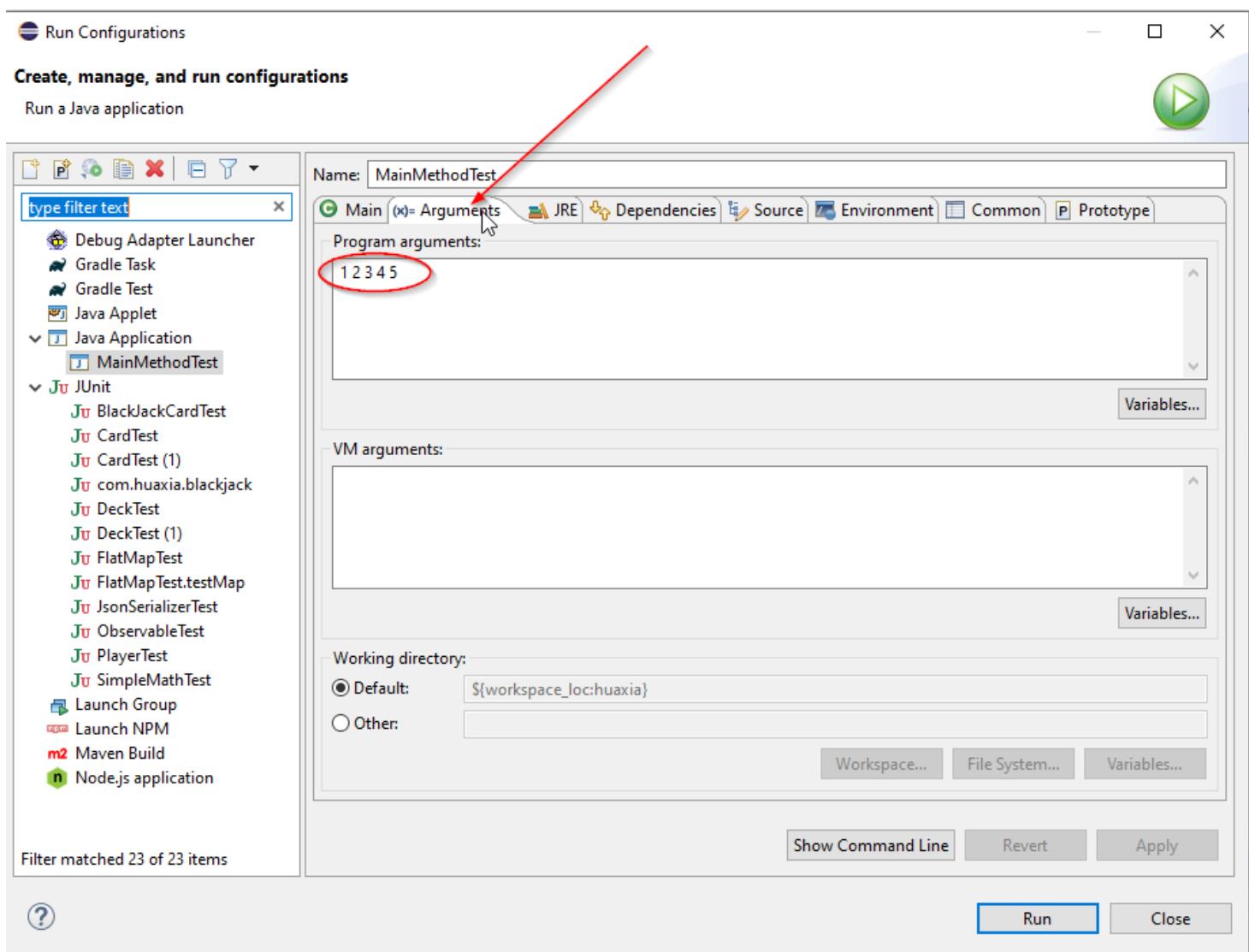
*** added in 1.4

**** added in 5.0

Modifier	Description
----------	-------------

Modifier	Description
public	access modifier, accessible for all classes
protected	access modifier, accessible in same package and subclasses
private	access modifier, accessible only in the declared class
[default]	no modifier, accessible in the same package
final	on class to make the class cannot be inherited by other class
final	on attribute and method so then cannot be override or make change
static	make attribute and method belong to the class, can be accessed directly by using class name.
abstract	abstract class or abstract method
transient	attributes and methods are skipped when serializing the object
synchronized	Method can only be accessed by one thread at a time
volatile	value of an attribute is not cached thrad-locally, and is always read from the "main memory"

- OOP concept > Abstraction (Cup.java; Computer.java)
- System print() (LearnForLoop.java right-angle triangle)
- main method (MainMethodTest.java); String[] args;
Run > Run Configurations... > Arguments



command line

```
$ java com.huaxia.test.MainMethodTest arg1 arg2 arg3 arg4
```

[Table of Contents](#)

Print String

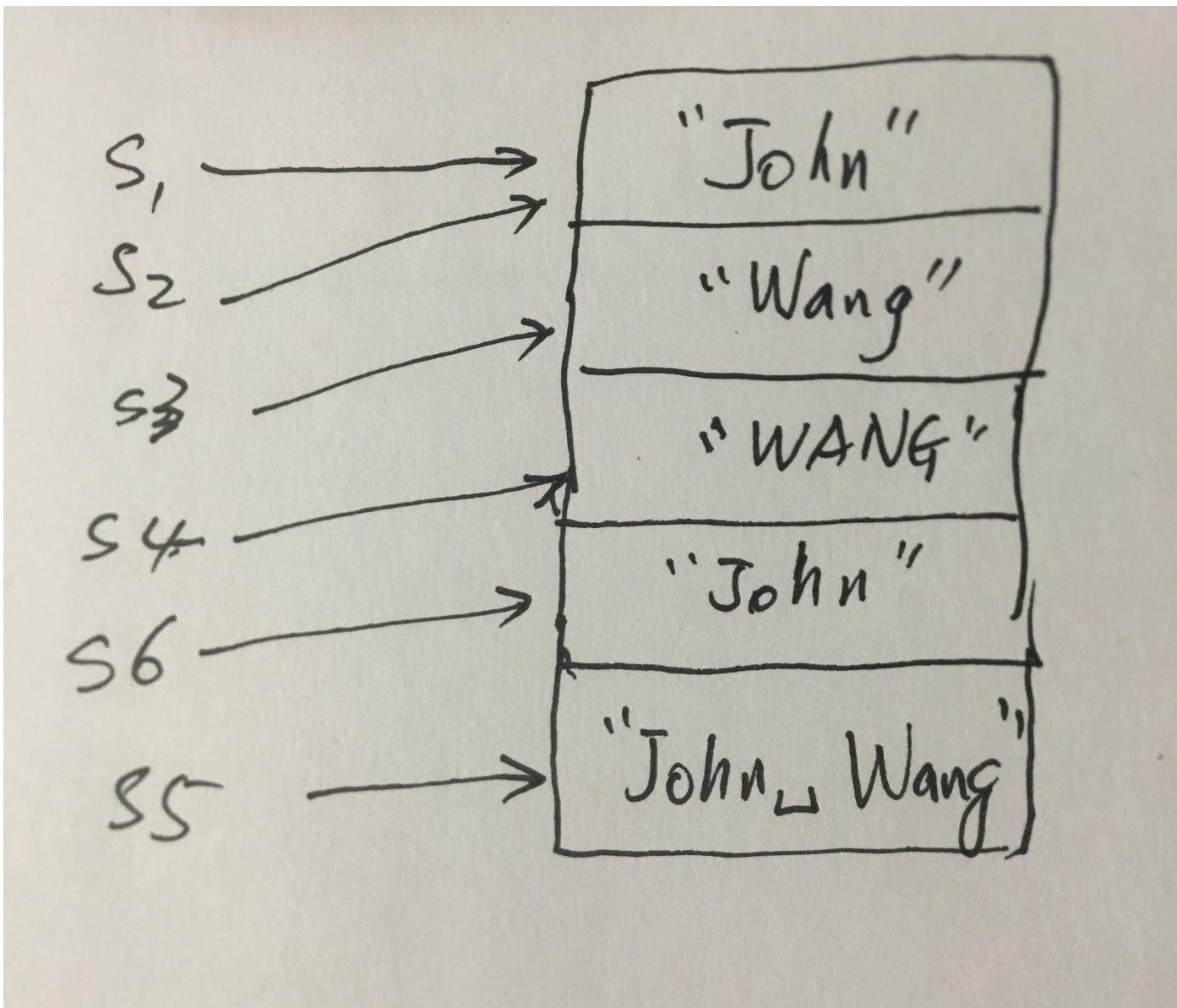
- Hello.java;
- Print.java; Simple math (Homework:)
- Print1.java; print String
- Print2.java; String.format() > java.util.Formatter
- Print3.java; StringBuilder vs. StringBuffer

Index	String	String Buffer	String Builder
Storage Area	Constant String Pool	Heap	Heap
Modifiable	No (immutable)	Yes(mutable)	Yes(mutable)
Thread Safe	Yes	Yes	No
Thread Safe	Fast	Very slow	Fast

- PrintTriangle.java; After learn Loop, print Triangle, Diamonds

[String format document](#) > String > format(String format, Object... args) > Parameters: format > java.util.Formatter

- StringTest.java; Understand String operations
 - stringLength
 - letterAtIndex
 - upperLowerCase
 - stringEquals



- escapeSequence
- stringBuilder
- stringBuilder2stringBuffer

[Table of Contents](#)

Using Variables

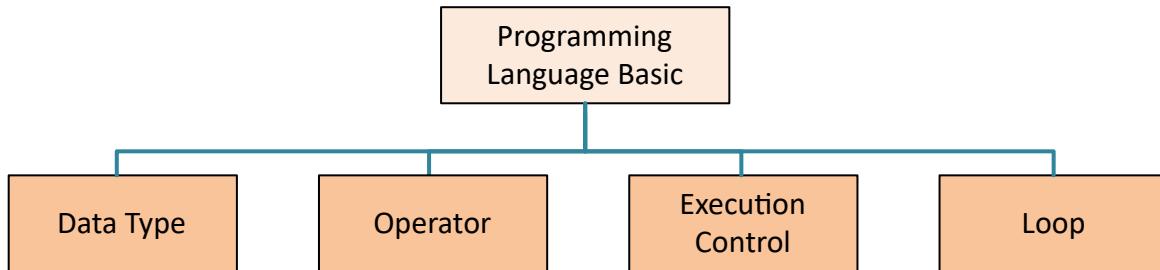
- variable naming convention
- start from letter or `_`, followed by combination of letter and number or `_`, `a~z`, `A~Z`, `0~9`
- cannot start by number
- cannot use special characters
- cannot use java reserved words

Comments

- Single line comments
- multiple lines comments
- Java Doc
-

$$v = \frac{4}{3}\pi r^3$$

Data type



Change variable name for windows: Alt+shift+R;

Primitive Data Types

Refer to: `DataTypeTest.java`

- array of int
- loop through an array with for each syntax
- array copy
- object array

Refer to: `ArrayTest.java`

Right-click [project name]:properties > Java Code Style:Formatter > Configure Workspace Settings>
Active profile: jwang > [Edit] > comments >

General settings

- Enable Javadoc comment formatting
- Enable block comment formatting
- Enable line comment formatting
 - Format line comments on first column
- Enable header comment formatting
- Preserve white space between code and line comments
- Never indent line comments on first column
- Never indent block comments on first column
- Never join lines

Table of Contents

Operator

- OperatorTest.java
 - mathOperator
 - doubleOperator
 - comparisionOperator
 - logicOperator

Table of Contents

- Reorganize project and package
 - highlight your java source code name drag to different project
- You see a Cup by your eye,
never see Cup in your mind in the real world.
never see Cup defined in our Cup.java class.

Table of Contents

if-else

- IfElseTest.java
 - ifElse()
 - ifElseif()
 - ternary()
 - number2Week()
 - isPerfectNumber(6)
 - prime(); method return true/false

(with for loop)

- print even number from 1 - 20
- print odd number from 1 - 20
- understand what % does

Add document for your constructor, methods and class

menu: Project > Generate Javadoc

[Table of Contents](#)

Loop

- ForLoop1.java; simple for loop
- ForLoop2.java; step by 2
- ForLoop3.java; negative step
- ForLoop4.java; forEach
- ForLoop5.java; break loop on condition
- ForLoop6.java; continue on condition
- ForLoop7.java; infinite loop, Scanner
- ForLoop12.java; nested for loop, print right triangle
- ForLoopTester.java
 - simpleFor
 - stepBy2
 - stepByNegative2
 - forEach
 - breakOnCondition
 - continueOnCondition
 - prime

- findPrimeNumbersBetween(40,50)
- perfectNumber.java
- findGCD
- rollDices
- infiniteLoop

[Print Homework](#)

- WhileLoop1.java;
 - whileTest
 - doWhileTest
 - breakOnCondition
 - continueOnCondition
 - dice (probability)
 - infiniteLoop
- GuessNumber.java;
- RollDice.java; Math.random()
- RollDice2.java; static method

[Table of Contents](#)

Simple Math

- SimpleMath.java; add(), sub(), mul(), div(). method overloading
- MathTest.java; built in Math functions
- LCM.java; Least Common Multiple
- GCF.java; Greatest Common Factor
- GCD1.java; Greatest Common Divisor
- GCD1.java; find GCD by while loop
- Prime.java; find prime number by definition
- PerfectNumber.java; find perfect number
- Fraction.java; Fraction math: add, sub, multiply and divide

							6 = 1 x 6
							6 = 2 x 3
							6 = 3 x 2
							6 = 6 x 1



[Table of Contents](#)

Array

- ArrayTest.java
 - intArrayTest()
 - elementTest()
 - forEachTest()
 - changeIndividualElement()
 - arraycopyTest()
 - objectArrayTest() (fix compiler error on line-77)
 - multidimensionalArrayTest()

[Table of Contents](#)

Array List

- ArrayListTest.java
 - review create string array
 - createIntegerArrayList (difference between array and ArrayList)
 - api doc > ArrayList, Arrays
 - array to ArrayList
 - creatObjectArrayList
 - practice
 - create float ArrayList display under certain condition
 - create a student list, write it to a file use forEach loop

Table of Contents

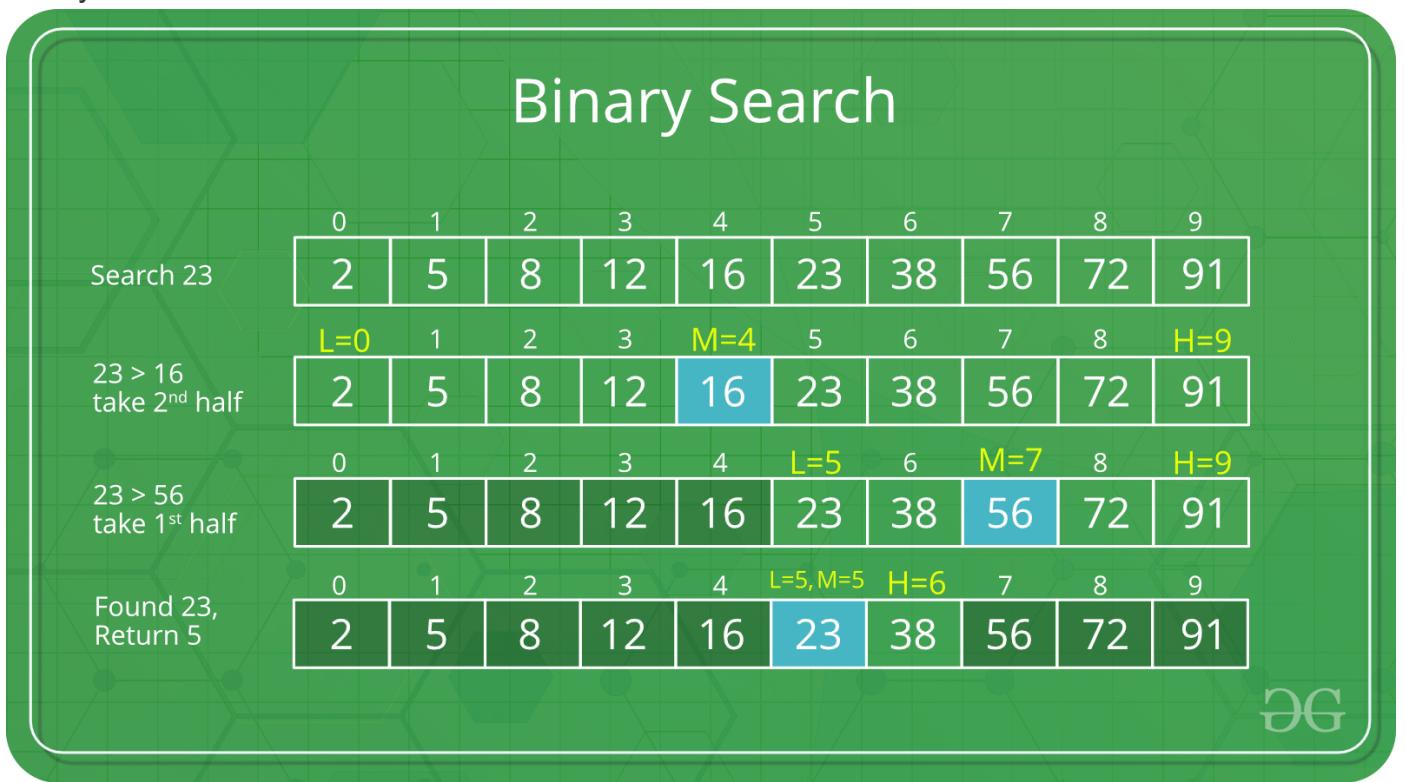
Switch

- SwitchTest.java
 - weekdayConverter
 - day2word
 - number2Weekday

Table of Contents

Data Structure

- QueueTest.java
- StackTest.java
- binary search



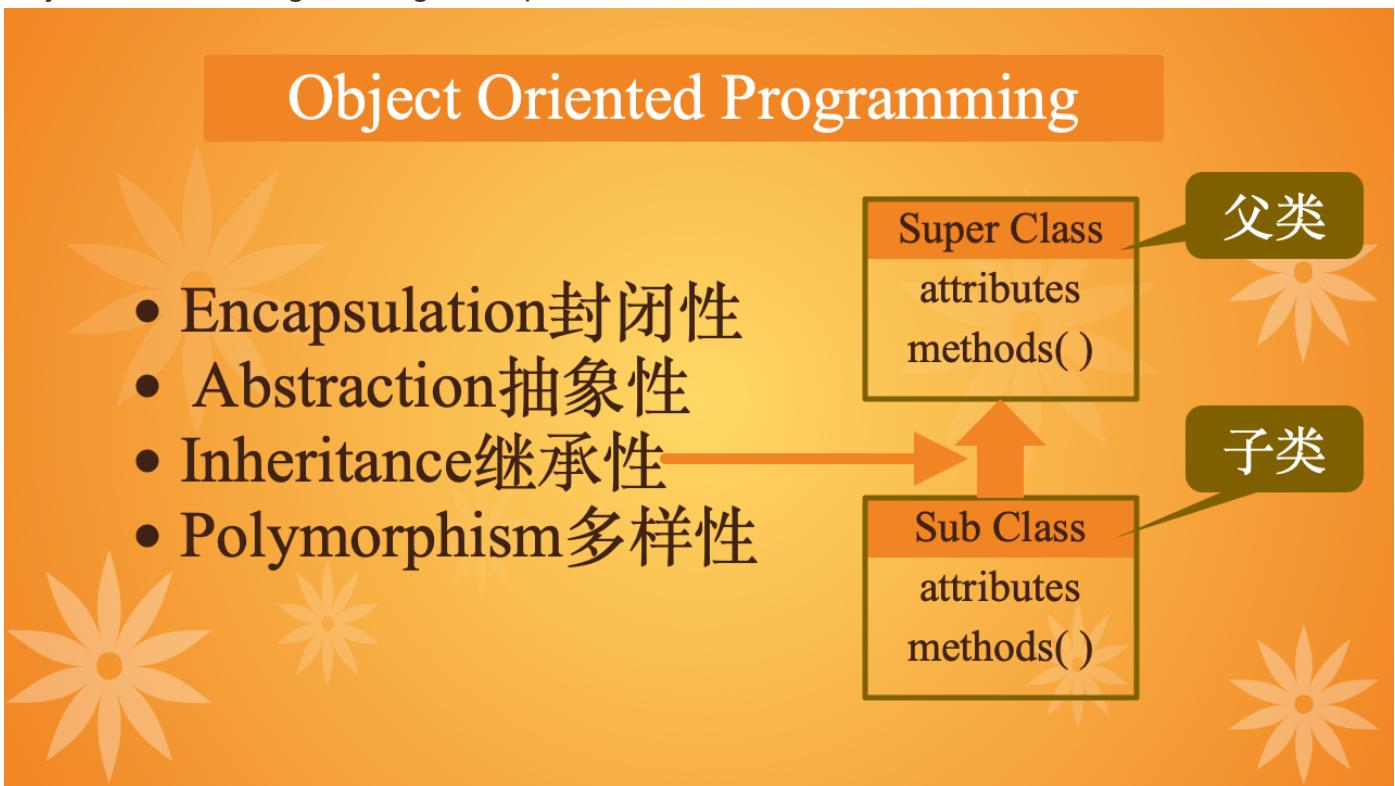
- sort (sortTest.java)

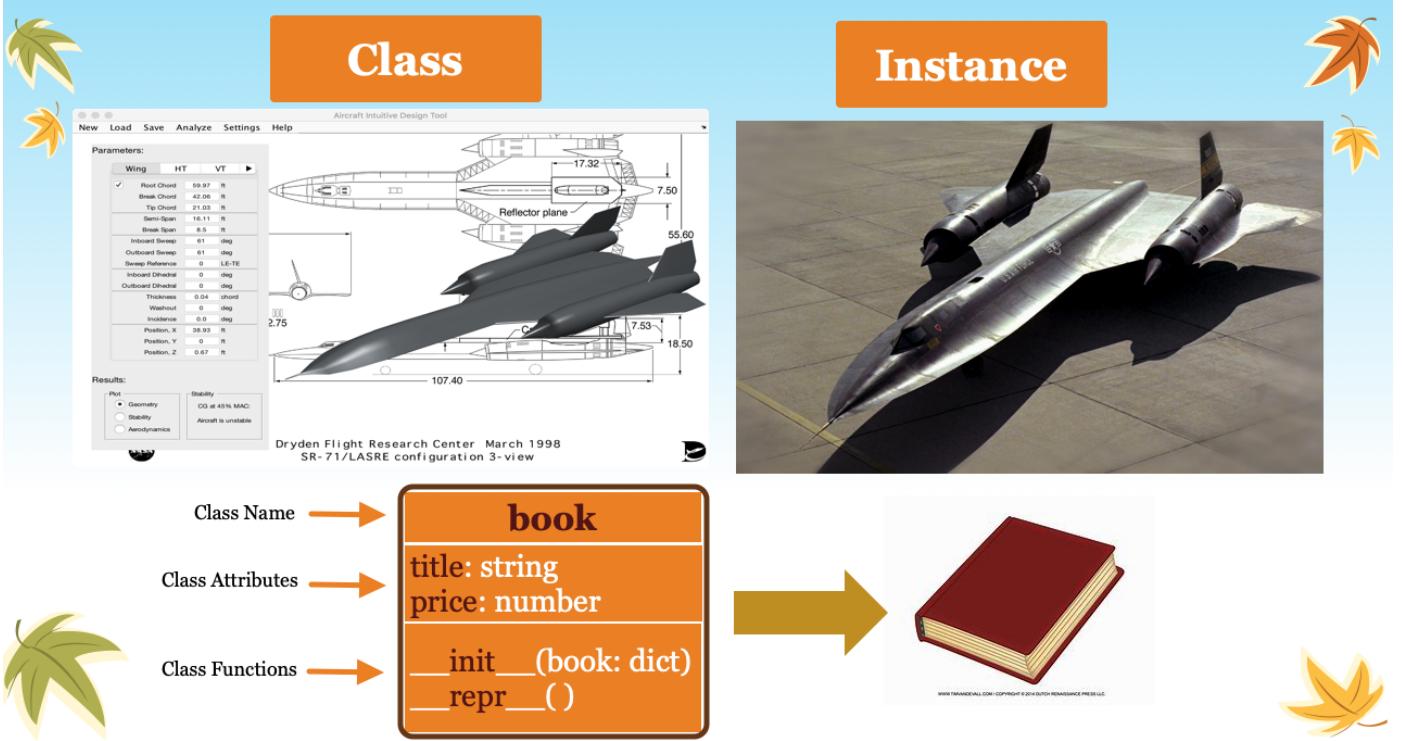
Exception

- Understand Exception Hierarchy
 - uncheckedArrayUsage
 - tryCatchTest() > add try-catch
 - area(1.0f)
 - area(-2.0f)
 - add throw new Exception
 - practice: rectangleArea() throw Exception test.

OOP

- Object Oriented Programming concept





- Abstraction

Cup.java

- Inheritance

SuperClass1.java

SubClass1.java

- Encapsulation

- Polymorphism

what's your name?

Student

Teacher

Employee

Manager

- Interface (variable, method signature)

- Abstract class (abstract method, concrete method)

- Class

- Constructor

Computer.java(String brand, int battery)

- Getter/Setter

- right-click > source > generate constructor using fields

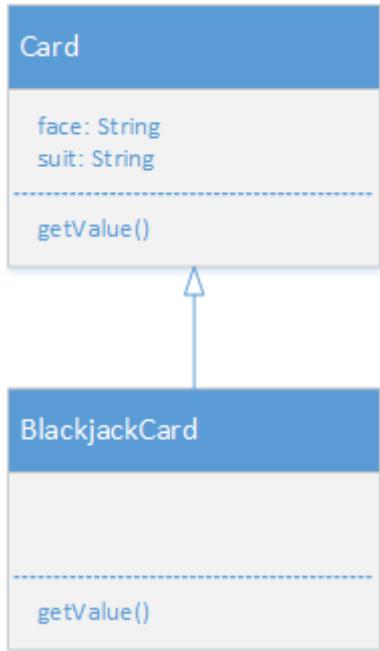
- right-click > source > generate getters and setters

- Method overloading (add(int, int), add(float, float))

- Method overriding (Card.java, Blackcard.java, getValue())

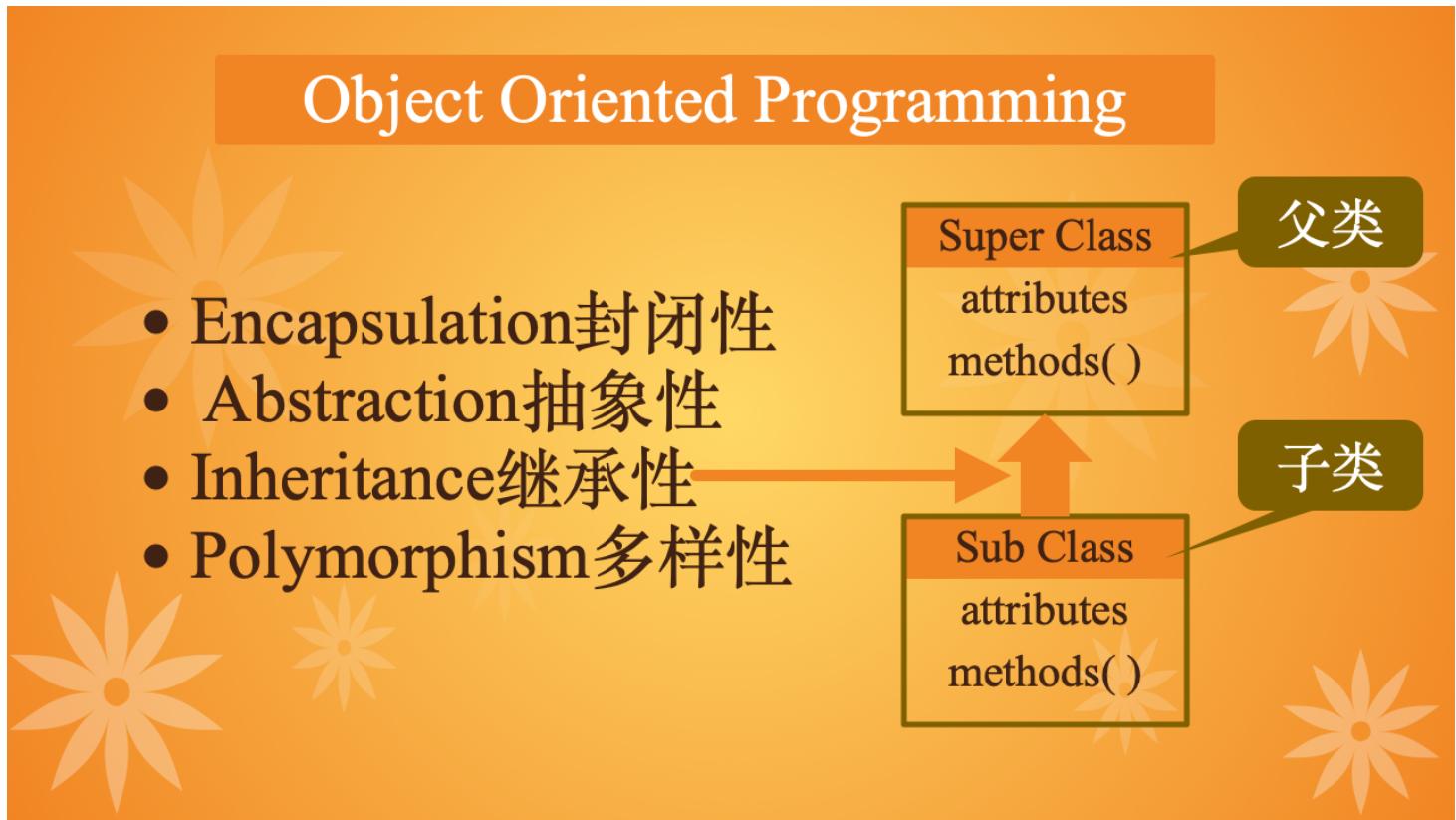
- Super class/sub class (protected vs. package default)

- `toString()` method
- `Card.java`; basic card class



- `BlackjackCard.java`; sub card class
- Why Java always start from class? Java is oop.

What is OOP? Abstraction, inheritance, encapsulation, polymorphism



- Abstraction

You have been doing it every day since you were 3 years old. You never see dog, you always

see individual a dog.

Same thing in our class, we abstract Cup from one real cup to our class cup.

```
workspace: /workspace/java
project: blackjack
package: com.huaxia.blackjack
refer Cup.java
```

Perspective windows

1. Package Explore
2. Navigator (Package==file folder)

Create new project ==> huaxia

create new package ==> com.huaxia.playground

drag the cup to that folder

create toString() method for Cup

3. what we used in Cup.java

```
package : file folder
class : abstraction, design, type
public : can be used in everywhere
float : decimal number
static : Class level attribute or function, can be used by class name
```

String: Java build in class

System.out.println(): Java build in function

public function => return type, name, parameters signature(finger print)

overloading (overriding)

cast

filled = 0.8f;

field = (float)0.8;

play around with main:

change signature,

remove

```
public String toString()
```

change float to int

add constructor

[Table of Contents](#)

File Access

- FileTest.java
 - readFileTest (create class, read a file, print content to console)
 - writeFileTest
 - appendToFileTest (api doc > FileWriter) (write "Hello world" to a file)
- Write_CSV_File.java
- Read_CSV_File.java
- GsonExample1.java; write Staff object to json file staff.json
- GsonExample2.java; read staff.json to build Staff object
- Review
 - search Scanner, Break, Continue
 - read content from a file, parse each line to a word, display on console
 - write hello world to a file
 - get user name from console, write hello and his name to a file
 - write even number between 0 to 20 to file
 - write odd number between 10 to 30 to a file
 - append integer 1 to 10 to a existing file
 - create a string array, write each element to a file
 - create a string array with names, write only name < your name to a file
 - read content from a file, parse each line to a word, write to another file

[Table of Contents](#)

All topics

- Hello World! (HelloWorld.java)
- OOP concept > Abstraction (Cup.java; Computer.java)
- System print() (LearnForLoop.java right-angle triangle)

- main method (MainMethodTest.java)
- Generate JavaDoc
- Package perspective, Navigator window, Console Window, Outline
- Data type (DataType.java)
- Array Test (ArrayTest.java)
- String Test (StringTest.java)
- Operator (OperatorTest.java)
- Scanner (ScannerTest.java)
- For-loop (ForLoopTest.java)
- Function (over loading, overriding)
- File (FileTest.java)
- While-loop (WhileLoopTester.java)
- ArrayList (ArrayListTest.java)
- GuessNumber.java (while loop, if-else, scanner)
- Switch (SwitchTet.java)
- Card.java
- CollectionsTest.java
- create HelloWorld.java
- Use scanner ask user name, and then say hello to him~~
- create any class you can think of~~
- how to find println document in java api doc

Java API Document

- Use Scanner ask user name, and save the hello to him in file
- create a Card class with face and suit fields(Card.java)
- create simple math functions add, sub, mul, div
- create array hold 52 cards (Card2.java)
- use Scanner get user input do simple math
- Find System.out.printf document in javadoc api online
- (ForLoopTest->forEach():add condition skip Ford)
- Eclipse Search button on toolbar, menu Search
- add javadoc in java source code
- generate javadoc
 - Reorganize project package
 - Use Eclipse generate Constructor, toString, getter-setter
 - Use Code Style Formatter to avoid comments change
- Refactor Rename (someone demo this)
- MainMethodTest.java

- DataTypeTest.java
- ArrayTest.java
- Reorganize project package
- Use Eclipse generate Constructor, toString, getter*setter
- Use Code Style Formatter to avoid comments change

[Table of Contents](#)

Applications

- Snake-Game pull from javahuaxia github

[Table of Contents](#)

String.format()

```
public static void main(String[] args) {  
    String name = "John";  
    String sf1 = String.format("name is %s.", name);  
    String sf2 = String.format("value is %f.", 3.1415926);  
    String sf3 = String.format("value is %5.3f.", 3.1415926);  
    System.out.println(sf1);  
    System.out.println(sf2);  
    System.out.println(sf3);  
}
```

Homework: do math

[Table of Contents](#)

- use text editor to write java program

```
(base) Johns-MacBook-Pro:java wangqianjiang$ javac Test.java  
(base) Johns-MacBook-Pro:java wangqianjiang$ java Test
```

Java no longer have the limit of file name must match class name
Make some mistake, and compile, such as x = 4

myTest.java

```
class Test1{
    static public void main(String[] args){
        System.out.println("Hello, World!");
    }
}

class Test2{
    static public void main(String[] args){
        int x = 5;
        int y = 6;
        System.out.println("x + y = " + (x + y));
    }
}
```

compare Java with Python [firstPy.py](#)

```
a = 4
b = 5
print("%d + %d = %d" %(a, b,(a + b)))
```

[Table of Contents](#)

References

[Java api document](#)

[Tutorial point](#)

[Java Tutorial](#)

[Breakout room](#)

[Java class type](#)

Homework 1-1

1. write md file to take notes about all stuff we have learned today.

Table of Contents