Constructors

When you create a constructor and create variables, initialise variables directly under the class name

Eg.

// in human.java

public class Human {

    double weight;

    String name;

    double height;

// constructor (type 1)

public Human() {

name = “nameless”

weight = 0;

height = 0;

}

// constructor (type 2)

public Human (String n, double w, double h) {

name = n;

weight = w;

height = h;

}

}

// in main.java

public static void main(String[] args) {

Human.h1 = new Human (); // constructor type 1

Human.h2 = new Human (“Amy, 100.0, 1,6); // constructor type 2

- if you put private keyword in front of the variable, no one else will be able to use those variables

- the keyword static means that there is only one copy

Initialisation

- make use of the new keyword to create a new object

- byte short int long will be initalised to 0

- float, double will be initialised to 0.0

- char will be initialised to \u0000

- boolean will be initialised to false;

- reference data types are all initialised to null --- means that no address is stored in the reference variable

Syntax Template

public class <Name> {

//containers/attributes

private <datatype> <varibaleName> (these are instance variables)

//constructors (for running a piece of initialization code when the object is constructed)

public <name> (<dataType> <variableName>, <dataType>, <variableName>){

// statements

}

// methods

public <dataType> < methodName> (<dataType> <variableName>

<dataType>, <variableName>){

return <Value of dataType>

}

Common Mistakes

- use this.<instance variable name> when the instance variable name is the same as the parameter name

- local variables are variables that only exist within the scope of the method

- instance variables will live as long as the object lives

- the keyword static means that there is only one copy

- since you need to make use of the template to create different objects, within the template never make use of static for your instance variable names as well as methods

- do not specify return type for constructors

- When you use the private keyword in front of an instance variable, you cant touch them directly so to access them you need to create methods (getter methods)

Java Naming convention

primitive data types:

byte, short, int, long

float, double

char, boolean

- start with lowercasee

reference data types:

Human h1

- when creating a new class start with uppercase

Wrapper classes

- to wrap around the primitive data type

|  |  |
| --- | --- |
| **Primitive Data Type** | **Wrapper Class** |
| byte | Byte |
| short | Short |
| int | Integer |
| long | Long |
| float | Float |
| double | Double |
| boolean | Boolean |
| char | Character |