OOP- object oriented programming. A program is a collection of objects. Example is University/a system contains objects that represent students, classes, instructors. Design principles : encapsulation, polymorphism, inheritance. Encapsulation-provide means of using a class, but it omits the details of how the class works.

Classes- Class names begin w/ a capital letter. Public class means any one can run the program

* **Class data-** **When designing a class, decisions about the following must be made. what data must be accounted for, what actions need to be performed, what data can be modified, what data needs to be accessible, and any rules as to how data should be modified. Class design typically is done with the aid of a Unified Modeling Language (UML) diagram.**

**Class methods-** Scanner\_Object\_Name.ext() Lione,Int,Double,Float

Keywords in Java : Boolean, break, case, char, do, doyble, else, for, float, if,return, shirt,private

Arithmetic Operators : all binary type. Add+, sub-, mult\*. Div/, modulus%

Assignment : Use an *assignment statement* to store data.The *assignment operator* is the equal (=) sign.The operand on the left side of the assignment operator must be a variable name. The operand on the right side must be either a literal or expression that evaluates to a type that is compatible with the type of the variable

Constants : name is all uppercase. SEPARATE\_WORDS\_LIKE\_THIS.

Name rules for classes, constants, variables, and methods –setter, and getters : Variables may not be any of the Java reserved keywords.

3 types of comments used in Java: //, /\*…\*/, /\*\*…..\*/

System.out.print() vs System.out.println() vs printf

Variables can only hold one value at a time. Local variables do not receive a default value. Local variables must have a value in order to be used.

Variables are programmer-defined names for: Classes variables constants methods

Variables can only hold one value at a time. A variable is a name storage location in memory.

Local variables do not receive a default value.

Local variables must have a value in order to be used.

Make use of packages in your project structure – do not use the default package. o edu.ilstu – Use this for most assignments.

Always use meaningful names for identifiers.

* Key words

Public, class, static,void

public class HelloWorld

{

public static void main(String[] args)

{

String message = "Hello World";

System.out.println(message);

}

}

Class names start with a capital letter.

Variable names and method names start with a lowercase letter. o Separate words with uppercase letters

Example: firstName • Constants should be all caps o Separate words with an underscore (\_) Example: TAX\_RATE

* Primitive variables actually contain the value that they have been assigned

int number = 25;

The value 25 will be stored in the memory location associated with the variable number.

- When a variable references an object, it contains the memory address of the object’s location.

Then it is said that the variable references the object.

String cityName = "Charleston ";

String objects- Objects are created using the keyword new

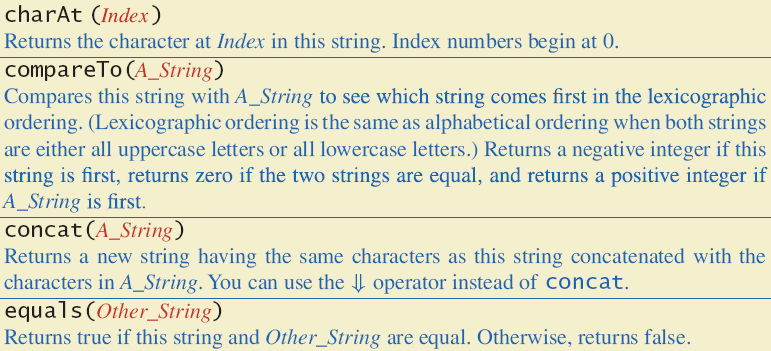
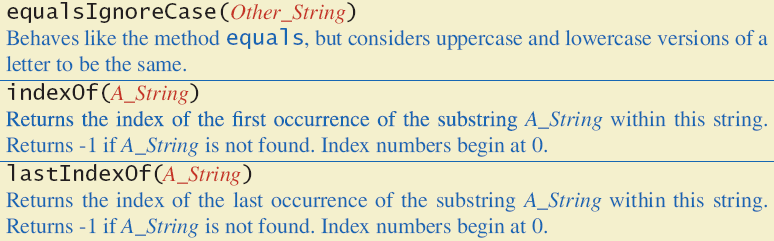
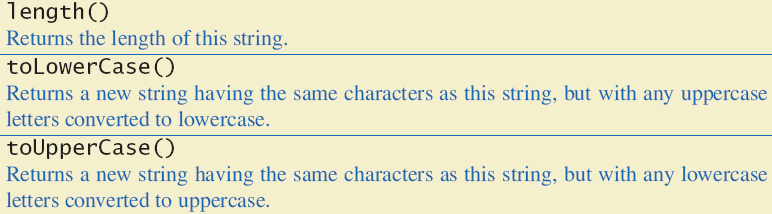
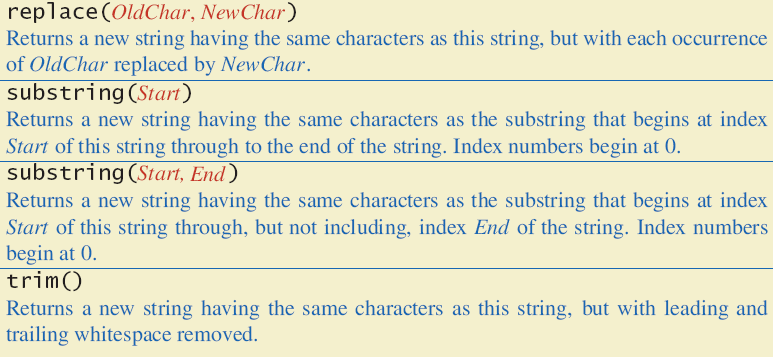
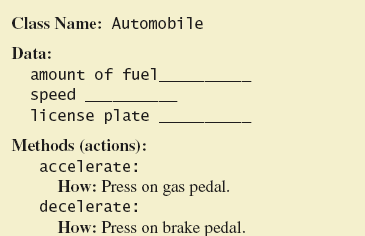
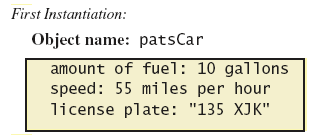
ObjectType objectName = new ObjectType(…);

A String variable can be assigned a String literal.

String value = “Hello”;

Strings are the only objects that can be created without the keyword new

Two strings(any number of strings) are concatenated using the + operator.

* User defined elements
  + args
  + message
* Method calls
  + System.out.println

For the Rectangle example, the accessors and mutators are:

setLength : Sets the value of the length field.

public void setLength(double len) …

setWidth : Sets the value of the width field.

public void setWidth(double w) …

getLength : Returns the value of the length field.

public double getLength() …

getWidth : Returns the value of the width field.

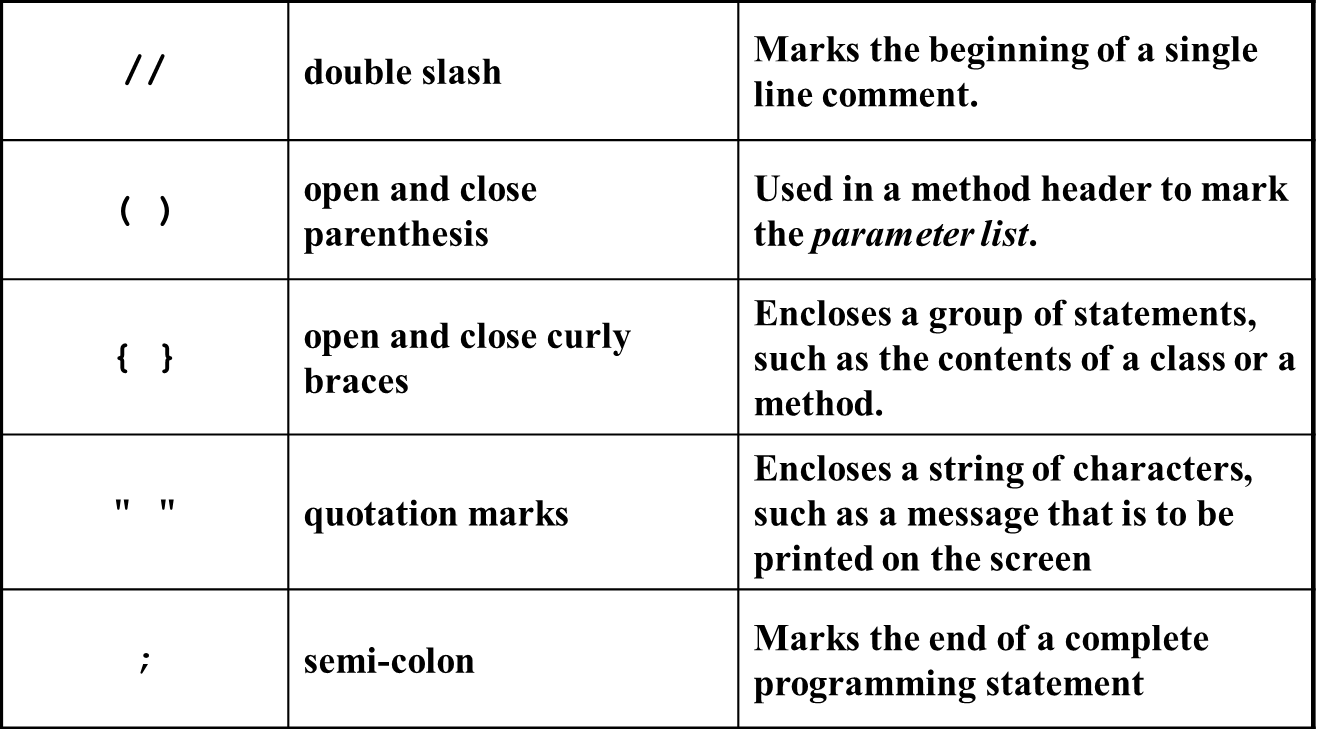
public double getWidth() …

`Other names for these methods are getters and setters.

Instance fields and instance methods require an object to be created in order to be used.

Note that each room represented in this example can have different dimensions.

Rectangle kitchen = new Rectangle();



Rectangle bedroom = new Rectangle();

Rectangle den = new Rectangle();

kitchen.setWidth(10);

kitchen.setLength(20);

invoking a method

No arguments: -- method-name();

With arguments-- method-name(argument-list, ...);

With return-- value ans = method-name(argument-list);

Methods can have multiple parameters.

The format for a multiple parameter method is:

*AccessModifier ReturnType MethodName(ParamType ParamName,*

*ParamType ParamName,*

*etc)* { }

Parameters in methods are treated as local variables within the method.

There are three Java access modifiers:

public: This access modifier states that any other class can access the resource.

private: This access modifier indicates that only data within this class can access the resource.

protected: This modifier indicates that only classes in the current package or a class lower in the class hierarchy can access this resource.

Attributes- The data elements of a class defines the object to be instantiated from the class.

The attributes must be specific to the class and define it completely.

Example: A rectangle is defined by = Length, width.

The attributes are then accessed by methods within the class.

Methods : The class’ methods define what actions an instance of the class can perform

The attributes of a class might be changed, accessed, and calculated. These methods are called accessors and mutators

Methods headers have a format:

AccessModifier ReturnType MethodName(Parameters)

{ //Method body. }

Methods that need to be used by other classes should be made public.

Constructors - Classes have special methods called *constructors*.

Constructors are used to create instances of the class – allocating memory and performing other operations at the time an object is created.

Constructors typically initialize instance fields and perform other object initialization tasks.