/\*

\* Course

\* Term

\* Assignment

\* Author: Trevor Barnes

\* Date

\*/

package barnestr;

Import Java.util.Scanner;

public class ClassName {

Scanner sc = new Scanner(System.in);

Int input = sc.nextInt();

}

public class SwitchDemo {

public static void main(String[] args) {

int month = 8;

String monthString;

switch (month) {

case 1: monthString = "January";

break;

case 2: monthString = "February";

break;

(…)

case 12: monthString = "December";

break;

default: monthString = "Invalid month";

break;

}

System.out.println(monthString);

}

} (Substitute for if-then-else)

A variable is anything that can change

(Doesn’t have a fixed value)

An object is an instance of a class

(Variables can be assigned a reference to the object

Math.PI, Math.abs(double, float, int, long),

Arc Cosine = Math.acos(double) (…)

Math.min(double, float, int, long) (…)

Math.toDegrees(double) (…)

Math.pow(base(double), power(double))

Math.sqrt(double)

for (int I = 0; I < maxCycles; i++) {

(behavior)

}

&& means AND, || means OR, == checks to see if the values are EXACTLY equivalent, ! means NOT

public int getSomething() {

return something;

}

void methods do not return anything, only “mutate”

public class Die {

private int numSides;

private int currentValue;

JOptionPane.showMessageDialog(null, “Hello World”)

JOptionPane.showInputDialog(null, “Are you alive?)

public class Die {

private int numSides;

private int currentValue;

public Die(int sides) {

this.numSides = sides; // Number of sides on die

roll(); // Rolls die to decide currentValue

}

public int getNumSides() {

return numSides;

}

public int getCurrentValue() {

return currentValue;

}

public void roll() {

currentValue = ((int) (Math.random() \* numSides)) + 1;

}

}

Arrays do not have methods

int[] nums = new int[size];

nums.length

String[] words = new String[102];

words[0] = "key";

words[1] = "pedal";

words[2] = "string";

words[3] = "piano";

words[4] = "sharp";

words[5] = "flat";

words[6] = "tune";

for(int i=0; i<words.length && words[i]!=null; ++i) {

System.out.print(words[i].length() + " ");

}

Array Lists have methods unlike Arrays

private ArrayList<ParkingLot> lots = new ArrayList<>();

lots.size()

lots.add(something) (Adds something to the end)

lots.add(location(int), something)

lots.clear()

lots.get(int)

lots.indexOf(something) (Returns the location of the first occurrence of something)

lots.isEmpty() (Returns true if the Array List is empty)

lots.set(location(int), something else) (Replaces an element at the location in the Array List with something else)

Local variables are declared inside a method or a block whereas instance variables inside a class but outside a method.

Class (static) variables change with the class

Instance variables do not change with the class. They belong to their own instance of the class and every instance of that object has its own copy.