**Chapter 2 – Elementary Programming**

**Starting a Java Program**

1. Java programs are built with classes.
2. Braces { } are used to start and end your class.

public class ComputeArea {

<Program Code appears here>

 }

**Main Method**

1. The main method is where your java program starts itself.
2. Commands found in the main method will execute in a top-down fashion.
3. To begin with, all of our code will be placed inside the main method.
4. Braces are also used to start and end your main method

public static main void (String args[]) {

 <program code >

 }

**Output**

1. **System.out.println()** - This command will output whatever text or formula is contained inside the parentheses and put the console at the start of a new line after its output.

2. **System.out.print()** - This command will output whatever text or formula is contained inside the parentheses, but does not start a new line. Subsequent output statements will appear on the same line, immediately following this output.

**Variables**

1. Like in algebra, we can use letters (*x*) or words (*area*) to hold values while a program runs.
2. There only a handful of primitive (or fundamental) data types:
	1. **int** – (integer) whole numbers, no decimals
	2. **double** – (floating-point) decimal numbers
	3. **char** – (character) – Single letter or number is stored in character variables
	4. **boolean** – Boolean variables hold either *true* or *false*.
3. Declaring variables
	1. Before a variable can be used, we must tell the program its name and type.
	2. Often a program will have variable declaration as their first lines of code after the class open brace.
	3. Variable Declaration Syntax: data\_type variable\_name;
		1. *int x;*
		2. *double area;*
		3. *boolean done;*
	4. You can assign a starting value to a variable when you declare it.
		1. *int age = 42;*
		2. *double pi = 3.14*
		3. *char firstInitial = ‘W’;*
	5. You can also declare a variable with a formula
		1. All parts of the formula must have already been declared and assigned a value.
		2. *double area = 4 \* side;*  (assuming *side* is a variable already assigned a value)
	6. Variable Naming Rules
		1. Can consist of letters, digits, underscores and dollar signs.
		2. Must start with letter, underscore or dollar sign. Cannot start with a number.
		3. Cannot be a reserved word (See Appendix A in book for list)
		4. Can be of any length
		5. Variable names and types will be lowercase. (standard naming rules dictate this)
		6. If a name consists of several words, concatenate them into one, first word is lower case, and first letter of second word is upper case. (Examples: *hoursWorked*, *circleArea*)

**Constants**

1. Constants are variables that never change.
2. Their value is assigned when it is declared.
3. Constants are always all upper case letters.
4. Constants use the *final* keyword when they are declared.
	1. final double PI = 3.14159;
	2. final int AGE = 42;

Numeric Variable Ranges

|  |  |
| --- | --- |
| Data Type | Range |
| int | approximately -2 billion to 2 billion |
| double | X-324 to X308 |

**Numeric Operators** (Listed in order of operation, first to last)

|  |  |
| --- | --- |
| Operator | Symbol(s) Used |
| Parentheses | ( ) |
| Exponents | Math.pow(2,3)  |
| Multiplication | \* |
| Division | / |
| Modulus (Remainder) | % |
| Addition | + |
| Subtraction | - |

**Modulus (%) (or mod) -** Used to calculate the remainder (what is left) after two numbers are divided.

Examples

* 11 % 3= 2, since 3 goes into 11 3 times, and there is 2 left over.
* 16 % 9 = 7
* 24 % 5 = 4

**String Class**

1. A String holds text.
2. String is a predefined class in java, unlike the above data types.
3. When declaring Strings, you will note a capital S is used in String.
4. When assigning a value to a String, that value is contained in double quotes.
	1. Ex: String name = “Bill”;
5. To append (add) to a string we use the + character.
	1. Ex: String message = “I “ + “love” + “ you.”; <Same as> String message = “I love you.”;
6. You can use a String with System.out.println(); to output the contents of a String.
	1. Ex: System.out.println(message); <outputs 🡪 I love you. >

**Scanner Class**

1. Scanner class is used to accept input from the user
2. A Scanner is an object that must first be declared before it can be used.
	1. *Scanner input = new Scanner(System.in);* 🡨Creates a scanner object named *input*
3. You have to specify the type of data you will be expecting when using the Scanner class
4. The Scanner class is often used when a variable is declared
	1. *int age = input.nextInt();*
	2. *double radius = input.nextDouble();* 🡨Saves the user’s input into a double variable, *radius*
	3. Types of inputs that work with Scanners
		1. *nextInt, nextDouble* – see above
		2. *next();* - Reads a String (text) that ends with a space, tab or enter(end of line)
		3. *nextLine();* - Reads all of the text from a line and saves it to a String.
5. There is typically a preceding System.out.println() statement requesting specific input.
6. Use of the Scanner class requires an import statement, as it is not part of the default Java library.
	1. Import statements appear at the top of your program, even before your class declaration.
	2. Syntax: *import java.util.Scanner;*

**Incrementing/Decrementing Variables**

1. Three methods of incrementing an integer variable by 1.
	1. x = x + 1; (counter)
	2. x +=1;
	3. x++;
2. Decrementing x by 2.
	1. x = x-2 or x -=2;
3. Multiplying x by 3.
	1. x = x \* 3 or x \*= 3
4. Dividing x by 5
	1. x = x / 5 or x /= 5
5. Finding remainder after dividing by 10
	1. x = x % 10 or x %= 10

**Number Type Conversion (Casting)**

1. We can convert from one data type to another using explicit *casting*.
2. Use the type to convert to in parentheses before the expression.
3. Converting from double to int
	1. double num1 = 3.5;
	2. int num2 = (int) num1; num2 = (int) 3.5 or 3

**ASCII**

1. ASCII – 8 bit code for typical English characters – 256 possible combinations
2. Look on asciitable.com for the character and integer matches.

**Escape Characters**

1. You can display special characters or codes inside of quotes by using the backslash (\) and a code
	1. \t – tab
	2. \n – new line
	3. \” – double quote
	4. \\ - backslash