Java is an object-oriented language, which means it relies on classes and objects of classes.

Classes can be created by this declaration at the top of the file:

public class MyClass

“MyClass” is the name of the class. You can name your class whatever you want, although it is best to have a capital letter at the beginning.

Any class that is going to be able to run a program has to have a main method, which tells the computer that this file is the **Driver class** or the class that can run the program.

public static void main(String[] args)

You can add methods to your classes. Instance methods are functions that each object of the class knows how to do, and class methods are function that the entire class knows how to do. Class methods have no knowledge of individual objects, the class does the method as a whole.

public class MyClass

{

public void doSomething()

{

System.out.println(“Hi”);

}

public int getSomething()

{

int x = 0;

return x;

}

}

To create an object of type MyClass, use this declaration:

MyClass object = new MyClass();

You can then use the object to call its methods:

object.doSomething();

There are several words in the method header that you can use to get information about the methods:

public – The method can be accessed by other classes

static – The method is a class method, so the entire class can do it

void – The method doesn’t return anything. This is the type we have primarily used so far.

int – The method returns an integer. An example is getSomething() above.

String – The method returns a string.

There are several types of data that you can use, called primitive types. They can each store a value that can later be accessed and used.

int x = 0;

double d = 0.0;

String s = “0”;

boolean b = false;

ints can store a whole number, doubles store decimals, strings store words made up of characters, and booleans store true or false.

You can use if-statements and else-if ladders to check the values of these variables.

if(x == 0)

{

System.out.println(“x is equal to zero”);

}

else if(x < 0)

{

System.out.println(“x is negative”);

}

else

{

System.out.println(“x is positive”);

}

**OR**

if(s.equals(“0”)

{

System.out.println(s);

}

else if(b == true)

{

System.out.println(“true”);

}

else

System.out.println(“false”);

OR

if(d == 0.0)

System.out.println(“zero”);

You can use some operators to check the values of multiple conditions:

Truth tables are a common way to show the output of certain inputs with these operators.

And (&&):

These operators can be used in if-statements:

boolean a = false;

boolean b = true;

if(a && b) //returns false

{

System.out.println(“f”);

}

else if(a || b) //returns true

{

System.out.println(“t”);

}

else if(!a && b) //returns true

{

System.out.println(“t”);

}

else if(a && !b) //returns false

{

System.out.println(“f”);

}

These can be used in combination, but it can get very complicated:

if(!a || (b && c)) OR if(b&&((!a||c) && !b))

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| True | && | True | = | True |
| True | && | False | = | False |
| False | && | True | = | False |
| False | && | False | = | False |

Or (||):

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| True | || | True | = | True |
| True | || | False | = | True |
| False | || | True | = | True |
| False | || | False | = | False |

Not (!):

|  |  |
| --- | --- |
| !True | False |
| !False | True |

Sometimes, it is helpful to have the user input something into the program. There are several ways you can do this. We focused primarily on one of them.

String s = JOptionPane.showInputDialog(null, “Enter a prompt”);

You have to be careful when you use this way because you have to have an import statement at the very top of your program:

import javax.swing.JOptionPane;

You can also use a Scanner, which we will cover later in more detail

import java.util; //at the top of the program

import java.io;

Scanner in = new Scanner(System.in);

System.out.println(“Enter a prompt”);

String s = in.nextLine();

If you want to call a method several times, you can use loops. There are two types of loops.

The first loop is a for-loop. It has three parts:

for (initialization, condition, iteration)

for (int x = 0; x < 10; x++)

{

//call some method

}

This code will call the method you want 10 times. You can add if-statements to the for loop.

The next type of loop is called a while-loop. It is slightly simpler, and it runs until a condition is false.

while(something == true)

{

//do some code

}

**Common Errors:**

Most errors you will make are syntax errors, which means incorrect spelling or incorrect “grammar” for the computer

* Having too many or too few “{“ or “}”
  + reached end of file while parsing
  + class, interface, or enum expected œ
* Missing a semicolon
  + ';' expected
* Misspelling something
  + cannot find symbol
* Putting a semicolon at the end of a loop or if-statement (very weird things will happen)
  + There will be random errors, usually “cannot find symbol” or “else without if“