**Coding Circle Java (translated from Coding Circle Python by Rohan Parikh)**

**Lesson 2**

* Boolean expressions: true and false
* You can compare values against each other to get a boolean result
* Comparative operators and their meanings (x and y can be any type of value)
  + x == y; x equal to y
  + x != y; x not equal to y
  + x > y; x greater than y
  + x < y; x less than y
  + x >= y; x greater than or equal to y
  + x <= y; x less than or equal to y
* Examples:

System.out.println((5 == 6))

output: false

System.out.println((“Hello” != “Goodbye”))

output: true

int x = 5

System.out.println(( 5 >= x))

output: true

**TRY IT**

See if 5.0000001 is greater than 5

Conditional Execution

Flow of control is a term we use that basically means that code runs from left to right, top to bottom, in the order it is written. But, we can write programs that change their behavior and change the flow of control, or the order the code executes, depending on the conditions.

We use if statements to run a block of code if a condition is true. It will not run if the condition is false. Syntax of an if statement:

if(condition) {

code\_to\_execute; //if condition is true

}

Example:

if(true) {

System.out.println(“This is true.”); //will print

}

if(false) {

System.out.println(“This is false.”); //will NOT print

}

if(2<3) {

System.out.println(“This is true”); //will print

In an alternative execution, there are two possibilities. One that happens if the condition is true, and the other that happens if the condition is false. It is not possible to have both execute. Syntax:

if (condition) {

code\_runs\_if\_true;

}

else {

code\_runs\_if\_false;

}

Example:

candies\_taken = 4;

if (candies\_taken < 3) {

System.out.println(“Enjoy!); //will run if candies\_taken is less than 3

else {

System.out.println(“Put some back.”); //will run if candies\_taken is greater than 3

}

Chained conditionals allow you to check several conditions. Only one block of code will ever run.

Use if/else if/else syntax (number of else if’s can be as much as you want.

candies\_taken = 4;

if (candies\_taken < 3) {

System.out.println(“Enjoy!); //will run if candies\_taken is less than 3

else if(candies\_taken < 6 {

System.out.println(“Put some back.”); //will NOT run because it encounters the first if statement first, will only run if candies\_taken is more than 3 and less than 6

}

else {

System.out.println(“You’re grounded.”); //will NOT run unless candies\_taken is 6 or more.

**Important:** Each condition is checked in order. The next else if is only checked if the first if or the else if before it is false. *Even if more than one condition is true, only the first true branch executes*.

**TRY IT**

Set did\_Homework = false. Check if it is true, then print out “You can play a video game”, if it is false print out “Go get your backpack.”

*Recommendation* Name your variables with descriptive names. Naming a variable 'a' is easy to type but won't help you figure out what it is doing when you come back to your code six months later, even with commenting.

Logical Operators

Logical operators allow you to combine two or more booleans. They are && (and), || (or), and ! (not).

Example:

System.out.println((true && true)); //prints true

System.out.println((false || true)); //prints true

System.out.println((!true)); //prints false

cleaned\_room = true;

took\_out\_trash = false;

if(cleaned\_room && took\_out\_trash) {

System.out.println(“Let’s get ice-cream”);

}

else {

System.out.println(“Get to work!”);

}

**Important:** In Java, short-circuiting is when a statement like an if statement doesn’t read all of the conditions. For example, if the statement uses the && operator, and it finds that the first condition is false, it will automatically know not to run the code inside the statement, despite what the second, third, etc. condition is.

This applies to || statements as well; if it finds the first condition is true, it will automatically run the code in the statement, even if all the other conditions are false. This means when writing statements with logical operators, be careful as to the order in which you write your statements, to prevent the code from doing something you don’t want because of shortcircuiting.

**TRY IT**

Check if the room is clean or the trash is taken out and if so print “Here is your allowance”

Nested Conditionals

You can nest conditional branches inside each other as well. Just be careful with your brackets!

Syntax:

if(condition) {

run\_this;

}

else {

if(condition2) {

run\_this2;

}

else {

run\_this3;

}

}

However, avoid nesting too deep, it can be difficult to read.

Catching exceptions using try and catch

You can also put code into a try/catch block. If the code has an error in the try block, it will stop running and go to the catch block. If there is no error, the try block completes and the catch block never runs.

Example + syntax:

try {

System.out.println(“Before”); //will print

int x = 5/0; //causes exception to be thrown (error to occur)

System.out.println(“After”); //will NOT print

}

catch (ArithmeticException e) {

//ArithmeticException is a type of exception that can be thrown where you have a divide by 0, etc. and e is the object of the exception

System.out.println(“Sorry, the universe doesn’t work that way.”); //only runs if there is an ArithmeticException in the try block

}

catch (Exception e) {

//This catch block handles generic exceptions, meaning exceptions that are caught in the try block but not by previous specific exception types; only executes if previous catch blocks do not run

System.out.println(“Exception occurred”);

}

The example would output:

Before

Sorry, the universe doesn’t work that way.

**TRY IT**

Try taking the square root of a negative number. If there is an error, print “What did you think would happen?”

**PROJECT**

We are going to build an application that recommends a car based on the user’s budget.

1. Ask the user what their car buying budget is and store in a variable called budget.
2. Try to convert the budget into an integer and store back in the variable called budget.
3. If step 2 fails print out “Please be realistic, you can’t buy a car on rainbows and love.”
4. If their budget is greater than 75000, tell them to buy a Tesla.
5. If their budget is less than 500 tell them they are better off riding the bus (it will be way more reliable than a $500 car)
6. Otherwise, tell them to buy a Toyota Corolla or something (try using boolean operators instead of an else statement).
7. Regardless of what their budget is, let them know they can get all their car shopping done at “[your name here] Auto Depot.”