

Control Flow

So far we have done...

- Printing
- Working with different types of data / values
- Creating and using variables
- Doing operations on variables
- Writing comments

Now we will learn about control flow. Control flow changes how the program runs. So far all of our programs run top to bottom. We will learn how to change that.

More Operations

Before we get into control flow, let's learn about one more operation.

Not only can we do math with numbers, we can also do math with strings. We can add strings together (This is the only operation we can do with strings).

```
var1 = "Hello" + "World"
System.out.println(var1)
System.out.println("Hello" + "World")
// The output to both will be "HelloWorld"

// We can also add strings and numbers together
System.out.println("Hello " + 1)
// The output will be "Hello 1"
```

Inputs

So far lets try to also get some input from the user.

```
// You have to write this at the top of your file to get input
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        // Create a scanner object
        Scanner scan = new Scanner(System.in);

        // Get input from the user
        String name = scan.nextLine();
        int age = scan.nextInt();
        double height = scan.nextDouble();
        boolean happy = scan.nextBoolean();

        // Print the input
        System.out.println(name);
        System.out.println(age);
        System.out.println(height);
        System.out.println(happy);
    }
}
```

Like shown above you need to write `import java.util.Scanner;` because Java doesn't let you take input easily. However people have written code that we can use making it easier for us. There are many different imports out there to solve many different problems.

Conditional Expressions

There are more operations we can do with numbers. We can compare numbers. We can compare if two numbers are equal, greater than, less than, etc.

```
// Works with any type of number
int num1 = 5;
int num2 = 10;

// Equal to
System.out.println(num1 == num2); // false

// Not equal to
System.out.println(num1 != num2); // true

// Greater than
System.out.println(num1 > num2); // false

// Less than
System.out.println(num1 < num2); // true

// Greater than or equal to
System.out.println(num1 >= num2); // false

// Less than or equal to
System.out.println(num1 <= num2); // true

// Remember variables are just placeholders for values
System.out.println(3 == 5); // false
```

Our New Order of Operations

1. Parentheses
2. Multiplication, Division, Modulus
3. Addition, Subtraction
4. Greater than, Less than, Greater than or equal to, Less than or equal to, Equal to, Not equal to

If Statements

Lets now take control of our programs. The first thing we can do is control if code runs or not. The first way to do this is with the `if` statement.

```
// If statements
if (true) {
    System.out.println("This will always run");
}
```

```
if (false) {
    System.out.println("This will never run");
}

if (3 == 5) {
    System.out.println("This will never run");
}
```

Every `if` statement has a parenthesis after with a boolean inside. If the boolean is true, the code inside the `if` statement will run. If the boolean is false, the code inside the `if` statement will not run.

Else Statements

Now lets say we want to do something `if` the code inside the `if` statement didn't run. We can do this with the `else` statement.

```
// Else statements
if (false) {
    System.out.println("This will never run");
} else { // This will always run if the code above doesn't run
    System.out.println("This will always run");
}
```

While Loops

Now lets say we want to run some code multiple times. We can do this with the `while` loop.

```
// While loops
int i = 0;
while (i < 5) {
    System.out.println(i);
    i = i + 1;
}
```

Just like `if` statements, `while` loops have a boolean inside the parenthesis. The difference is everytime it is true, the code inside will run and keep repeating until the boolean is false.