Summary of Intro Concepts

Objectives

- Understand what a programming language is.
- Send messages to objects asking them to do something using a series of java statements.
- *Understand what an object or class is.*
- *Use a variable to store data or object and explain what an object variable is.*
- *Explain how casting converts between different types and why you might need it.*

Programming concepts

- Programming is a way of describing to a computer a task that you want it to perform.
- An algorithm is a program that's intended to be used over and over again by lots of people, to solve a general problem. There may be many implementations of an algorithm with small variations, but in the end they all do the same basic thing.
- Computers are good at doing calculations very fast, storing infromation, and looking up stored information. If we want to get the computer to do more complex and creative tasks, we need to write a program (or use an existing program) which puts together basic calculations and information to do something more complicated.

- A programming language is a special kind of language that serves as a middle ground between human language and the actual instructions understood by the computer. Similar to the language used in a recipe, programming languages have a stricter, more explicit structure that allows us to translate it into something the computer can replicate.
- When you compile your program, it is translated into a stream of binary code (on and off signals) which can be interpreted by your computer as instructions and data. Everything you have on your computer is encoded as a series of binary digits (bits) in memory.
- Binary can be used to store numbers (in base 2) as well as text, using encoding schemes like ASCII and Unicode.

- Programming is about manipulating information and doing useful things with that information. We can do this more easily by naming variables and creating objects.
- In this course, we choose to use the Java programming language. Like most programming languages, Java has the ability to do basic comparisons and mathematical operations.
- Java uses types to define where each piece of information you manipulate can be used. Some of the primitive (non-object) types Java has include int and double. Other more complex types, like String, are objects.
- Objects are ways of storing information about a particular type of thing together in one place. similar to the way objects group together stuff in the physical world. Objects have methods, pieces of code that manipulate information stored in the object

- The characteristics and capabilities of each type of object are defined in classes.
- We can create and initialize objects with statements like:
  - new Class();
  - new Class(value1,...);
  - new World();

- *Variables* can be used to store primitive types like ints and doubles, as well as references to objects.
- Variables are an extremely useful part of your code for several reasons:
  - Holding information: Variables can store information temporarily for you so you don't have to write a long and complex piece of code all on one line!
  - Making your code more legible: Use. Names. That. Make. Sense. ... PLEASE! ...
  - Specifying the type of something: Creating a variable forces you to think about what the type of something is. Also, when you declare a variable to be an int, or double, or String, it will keep its type. Java is *strongly typed*, meaning that it *really* cares that you don't mix and match types without telling it, and will warn you if you make a mistake.

- We can declare variables to store and name values and objects we care about:

```
Type name = null;
Type name = new Class([value1,...]);
World earth = new World();
```

- You must declare a variable before you can use it, using the syntax *type name;* or *type name = value;*
- It may only be declared once
- You can re-use a variable once it has been declared
- A variable's type can include any of the primitive types (which start with a lower case letter) *e.g.*, int, double, char, boolean or class types (easy to identify because these types always start with an uppercase letter). This could include classes from Java's library, such as String or classes created by you or someone else such as World

- We can send objects a message to do something by calling their methods using the following syntax:

  objRef.method();

  objRef.method(value1,...);

- We create a method using the syntax *visibility returnType name([type name, type name, ...]);* Once we write this method we can use it over and over again. The parameters (in parentheses) can be passed to the method to change its behavior and tell it what information to manipulate more specifically.

- Variables are either primitive types like int or double, or objects, like String. Object variables have a reference stored in the variable, which points to memory somewhere else. Primitive types actually have the value stored in the variable. In Java, object memory is created when you use new, or for some special cases like Strings when you specify a literal value like "hello".
- Sometimes we need to convert a variable to something of a different type. For some primitive types, like int and double, Java does some of this automatically. For example, in the statement double a = 1/2; we get the result 0.0, because Java does integer division, throwing away the fractional part and giving us an int, and casts back to a double after doing the math. If we cast one of the operands to a double like 1/(double)2, the division operation will be floating-point division. An alternate way to fix this would be to use a decimal literal like in 1/2.0 instead of doing explicit casting.

- Java provides a library of useful code organized into packages. We will have used the Math library, String library and there will be many more to come.
- When writing code, you should always comment what you do, and follow the Rules for Writing Clean Code.
- In this module, you hopefully learned some basics of how to write simple programs, and how Java types and variables work.