Introduction to Programming

An Introduction to NetBeans IDE

What is Java?

JAVA is a *high-level programming language* developed by Sun Microsystems. Java was originally called OAK, designed for handheld devices, but was unsuccessful. In 1995 Sun changed the name to **Java** and modified the language to take advantage of the burgeoning World Wide Web.
Lack of standardization of computer operating systems (Windows vs MAC OS vs LINUX) and hardware platforms (Apple vs PCs) has led to many problems for programmers. Often programs were written one way for one system and then re-written for other systems, or programs were written for just one system or the other. Because **Java** compiles bytecode (binary language code), Java programs can be understood by all hardware and operating systems. As long as the computer has the software to translate the bytecode (e.g., Windows, MAC and LINUX machines), the Java program can be run without being re-written. This portability is what makes Java a popular programming language, well suited for applications and the Internet.

Java is an **object-oriented programming language** similar to C++ and Turing. Object-oriented programming is a way of writing programs with the common theme that the program is a collection of objects that work together to perform the task at hand. Each object is a collection of **data** and **methods** that manipulate the data to perform required tasks. You will learn more about object-oriented programming as the course progresses.

Getting Started with NetBeans

You will be using a software development environment called NetBeans to create your java programs. You will need to download and install the package that includes the both the NetBeans platform SDK as well as Java SE.
Once you have accomplished these two tasks continue on ahead with the remainder of the content.

Basic Output in Java

The first Java program you created in the video tutorial above was a “Hello World” program. Let's take a closer look at it.

```
public class Main{
    /*
    *Created by J. Smith
    *on January 5th
    *to display Hello World on the screen
    */
    public static void main(String[] args){
        System.out.println("Hello World");
    }
}
```

System.out.println ("Hello World"); is responsible for outputting words to the screen.

The function: System.out.println( );

This function is part of the Input/Output library of the Java language. It is used to output text to the screen. **System** is actually a class set up by Java for us to use; **out** is a variable of the System class; and **println()** is a method supported by the System class. You will learn more about these terms later in the course.

The function **System.out.println( );** takes letters (**strings**) or values (**numbers**) as a parameter and displays it on the screen (anything inside of the ( ) is displayed). You can display more than one string or numeric by joining them together using the "+" operator.

Example
System.out.println("Hello " + "how are you today");
This code will print out the phrase "Hello how are you today" on the screen by joining the two strings together into one line of output.
System.out.println("Your age is " + 21);
This code will print out the phrase "Your age is 21" on the screen by joining the string and the numeric together into one line of output.

Printing special characters using the System.out.println(); function:
Certain ASCII characters that are difficult to represent in a string either because they are not viewable (like an enter key) or because they are part of the formatting of strings in Java (like a " or a ' or a \ ) may be displayed using an **ASCII character code** or **control code**.
Shown below is a chart listing of some common **control codes:**

Control Code	***What it means***
\n	new line
\r	return character
\t	tab
\b	backspace
\"	double quote
\'	single quote
****	backslash

Example

If you wanted to print the phrase **Go "LEAFS" Go** (quotes around LEAFS) on the screen, you would use the code:

```
System.out.println("Go \" LEAFS \" Go");
```

If you wanted to print GO on one line, LEAFS on another line and Go on a third line you could use the code:

```
System.out.println("Go \n LEAFS \n Go");
```

resulting in screen output of:

```
Go
LEAFS
Go
```