Hyperion Development

Task: Introduction to Java

www.hyperiondev.com

Introduction

Welcome to your training in Java Programming!

For any queries regarding the course, help needed understanding the tasks, or general comments, please contact us at help@hyperiondev.com.

Why Java?

Java is a powerful and widely used programming language. It is a standard for IT teaching in South African high schools, universities, and even in other countries. Java syntax (the way the code is written) is very similar to other languages such as C++, C, Python and JavaScript.

Get excited to learn such a popular and fun programming language!

A brief history of Java:

James Gosling, Mike Sheridan, and Patrick Naughton initiated the Java language project in June 1991. Java was originally designed for interactive television, but it was too advanced for the digital cable television industry at the time. The language was initially called *Oak* after an oak tree that stood outside Gosling's office.
Later the project went by the name *Green* and was finally renamed *Java*, from Java coffee. Gosling designed Java with a C/C++-style syntax that system and application programmers would find familiar.[1]

Not all languages can boast like Java:

[1] Java (programming language)

One of the main reasons Java is so popular is its *platform independence,* which means that Java programs can be run on many different types of computers. A Java program runs on any computer with a *Java Runtime Environment,* also known as a *JRE,* installed. A JRE is available for almost every type of computer – PCs running Windows, Macintosh computers, Unix or Linux computers, huge mainframe computers, and even cell phones.

Java is inherently *object-oriented* (you'll learn more about this later), which means that Java programs are made up of programming elements called *objects.* Simply put, an object is a programming entity that represents either some real-world object or an abstract concept.

The Java language itself is very simple. However, Java comes with a library of classes that provide commonly used utility functions that most Java programs can't do without. This class library, called the *Java API,* is as much a part of Java as the language itself. In fact, the real challenge of learning how to use Java isn't learning the language; it's learning the API. The Java language has only 50 keywords, but the Java API has several *thousand* classes – with tens of thousands of methods you can use in your programs.

The instructions for your first Java Task are below. Before you get started we strongly suggest you start using Notepad++ to open all text files (.txt) and jGrasp to open Java files (.java). Do not use the normal Windows notepad as it will be much harder to read. Simply right click a file and then select > Edit with Notepad++ to be able to read it. If you cannot see 'Edit with Notepad++' then you do not have it installed. Do not continue unless you have installed JGrasp and Java on your computer. If you have any issues in this process, please contact us.

Instructions

First read example.java, open it using jGRASP (Right click the file and select 'Open with jGRASP").

- In this folder there is a file called example.java
- Open this folder using the JGRASP program. You can either do this by right clicking on the example.java file, going to the 'Open with' menu, and selecting JGrasp.exe. Alternatively, you can run the JGRASP program on your computer, go to the top left corner of the program, select File->Open and navigate to example.java on your hard drive and double click on it to open it.
- Once example.java is open in JGRASP please read all of its content very carefully. This will help you understand the basic structure of a Java program.
- There is a compulsory task at the end of the example.java document. The instructions of what to do to complete the task can be found here. You must complete this exercise to continue onto the next task of this course.

Compulsory Task

Follow these steps:

- Create a text file in this folder called 'Task 1'.
- Open the example.java file in jGRASP.
- Use jGRASP to compile the program.
- Use jGRASP to then run the program.
- Copy the output of the program and paste it into your text file.

Things to look out for

1. Make sure that you have installed and setup all programs correctly. You have setup **Dropbox** correctly if you are reading this, but **jGRASP or Java** may not be installed correctly.
2. If you are not using Windows, please ask your tutor for alternative instructions.

Still need help?

Just write your queries in your comments.txt file and your tutor will respond.

Task Statistics

Last update to task: 18/01/2016.
Authors: Riaz Moola and Jared Ping.
Main trainer: Umar Randeree.
Task Feedback link: Hyperion Development Feedback.