COMP102 Prac1 – 12 July 2019 [14:00 – 17:00]
Command-line Arguments in Java

Instructions

Log on to a computer in the LAN and attempt the following exercises. You may consult the java documentation for information about classes and methods you do not understand, for example: https://docs.oracle.com/javase/8/docs/api/java/io/File.html . Once you have completed the exercise, you may request to sign the register before leaving the prac.

Introduction

Operating systems generally come with a kernel through which you can specify commands that typically either execute a particular program, request for some information or navigate the file structure. When the kernel is opened it displays a command prompt which indicates that it is waiting for the user to issue a command execution. For example, in linux the command *ls* lists all the files in the current directory while the command ***cd ..*** moves one level up the file structure. The command *javac* executes the java compiler, it however expects you to provide at least one argument specifying the file to compile, for example: *javac Program.java* . After a program is compiled successfully, an executable is generated and this executable can itself be executed on the kernel. Every java program anticipates that it may be executed on the command prompt and it allows for arguments to be supplied after the name of the executable. These arguments are stored as individual strings in an array.

Problem

Your neighbor would like to leave directions to a hidden treasure, in a password protected program. The program, should only allow the user 5 seconds to guess the password as a command-line argument, after which the program deletes itself. He has requested you to write such a program and give him the source code. He will then insert his desired password into the code, compile it then give the executable to his lawyer. Upon his death, the lawyer will give all his family members a chance at retrieving directions to the treasure.

Exercises

1. Save the file Treasure.java on your computer then write comments on each line of code to explain what it does (or its meaning).
2. Open the console, navigate to your saved program (Treasure.java) and compile it using the command: *javac Treasure.java*
3. View the files in the current folder using the command *ls*. Note that a file Treasure.class (executable) has been added.
4. Run the executable using the command *java Treasure*
5. View the files in the current folder using the command *ls -a*. Note that the executable has been removed but a hidden file (.stamp.txt) has been added.

Learning Outcomes

1. Know the syntax of different java application components such as a class, main method, statements and comments.
2. Be conversant with editing, compiling and running a java application on the console and using an IDE.
3. Appreciate that java is case sensitive and requires attention to detail when writing code.
4. Understand the importance of java command-line arguments and how to use them.
5. Consult the java documentation for information about inbuilt classes and methods.