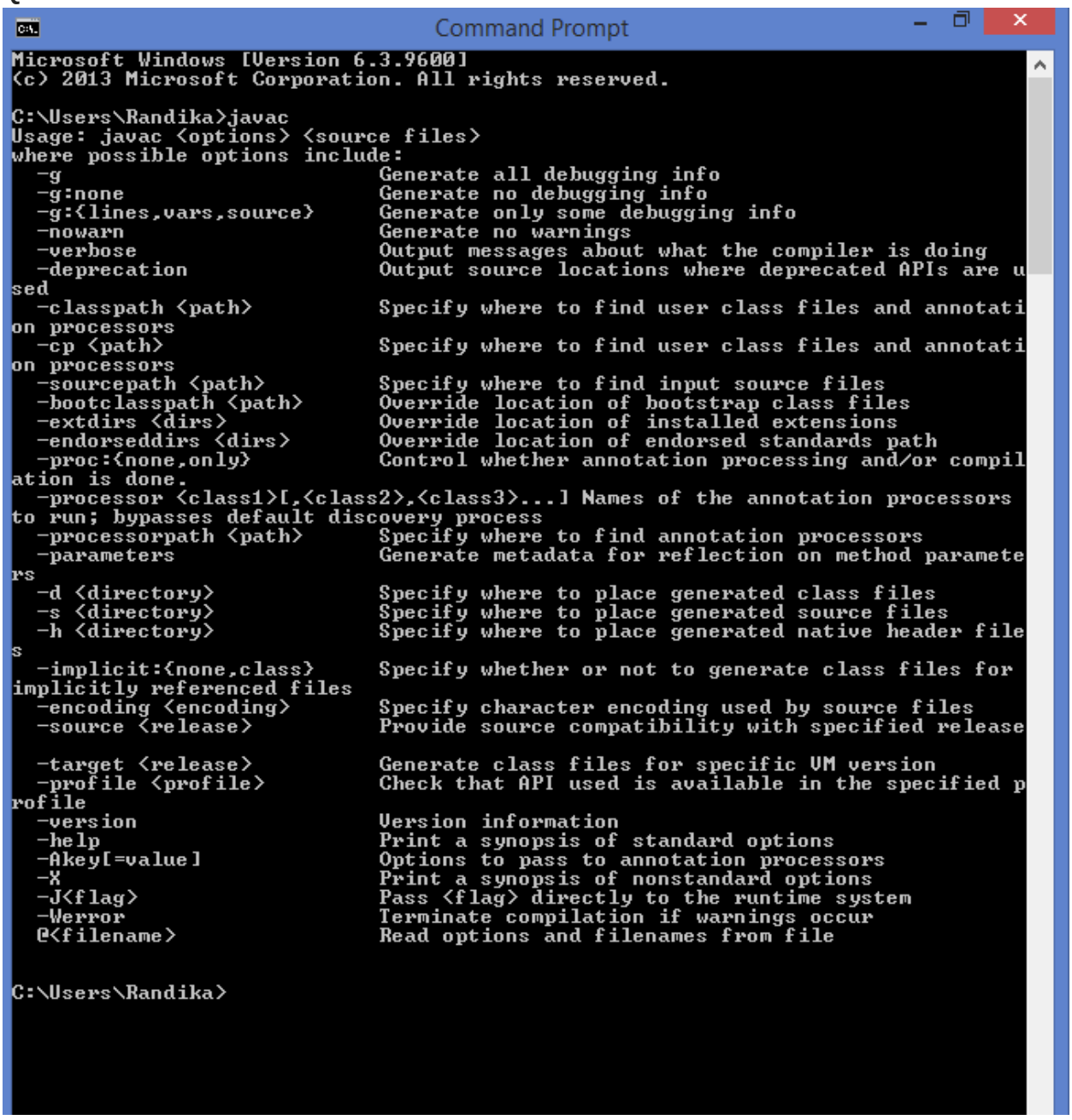
Rapid Application Development

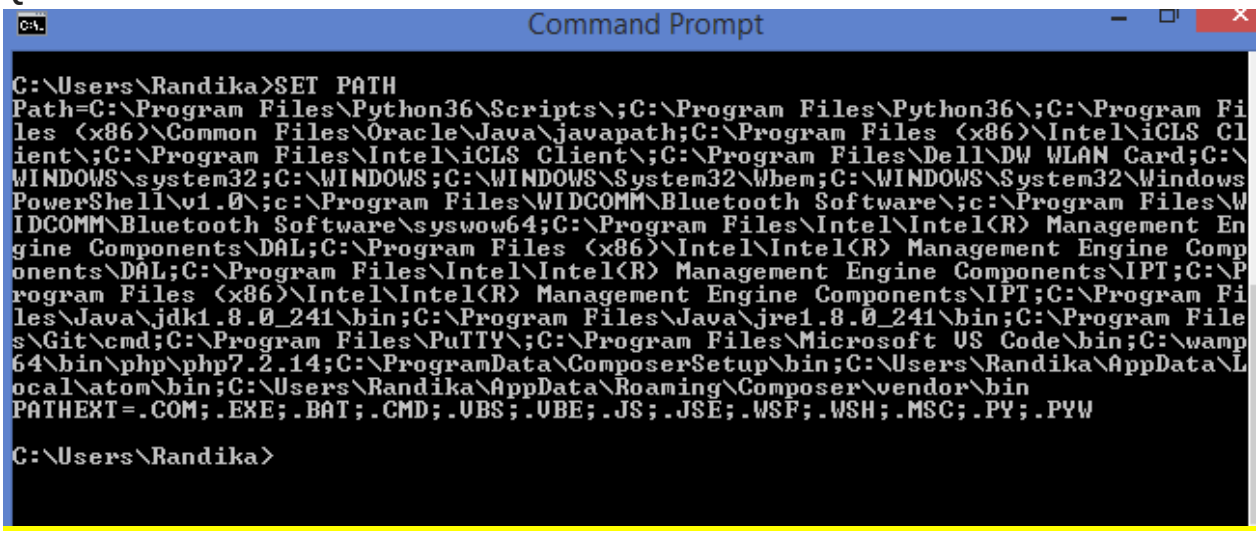
IS 2104

- **W.H.M.GUNATHILAKA**
- **2018/IS/027**
- **18020275**

Q1.

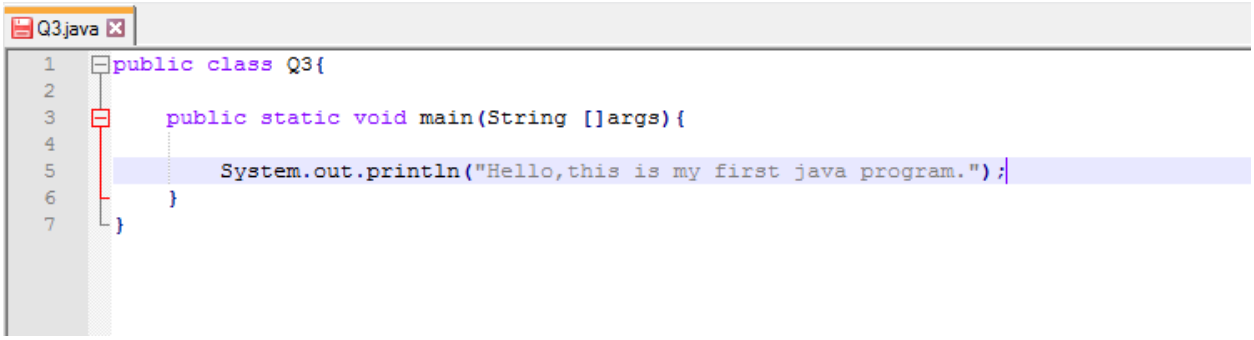

Command Prompt
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.
C:\Users\Randika>javac
Usage: javac <options> <source files>
where possible options include:
-g Generate all debugging info
-g:none Generate no debugging info
-g:{lines,vars,source} Generate only some debugging info
-nowarn Generate no warnings
-verbose Output messages about what the compiler is doing
-deprecation Output source locations where deprecated APIs are u
sed
-classpath <path> Specify where to find user class files and annotati
on processors
-cp <path> Specify where to find user class files and annotati
on processors
-sourcepath <path> Specify where to find input source files
-bootclasspath <path> Override location of bootstrap class files
-extdirs <dirs> Override location of installed extensions
-endorseddirs <dirs> Override location of endorsed standards path
-proc:{none,only} Control whether annotation processing and/or compil
ation is done.
-processor <class1>[,<class2>,<class3>...] Names of the annotation processors
to run; bypasses default discovery process
-processorpath <path> Specify where to find annotation processors
-parameters Generate metadata for reflection on method paramete
rs
-d <directory> Specify where to place generated class files
-s <directory> Specify where to place generated source files
-h <directory> Specify where to place generated native header file
s
-implicit:{none,class} Specify whether or not to generate class files for
implicitly referenced files
-encoding <encoding> Specify character encoding used by source files
-source <release> Provide source compatibility with specified release
-target <release> Generate class files for specific VM version
-profile <profile> Check that API used is available in the specified p
rofile
-version Version information
-help Print a synopsis of standard options
-Akey[=value] Options to pass to annotation processors
-X Print a synopsis of nonstandard options
-J<flag> Pass <flag> directly to the runtime system
-Werror Terminate compilation if warnings occur
@<filename> Read options and filenames from file
C:\Users\Randika>

Q2.

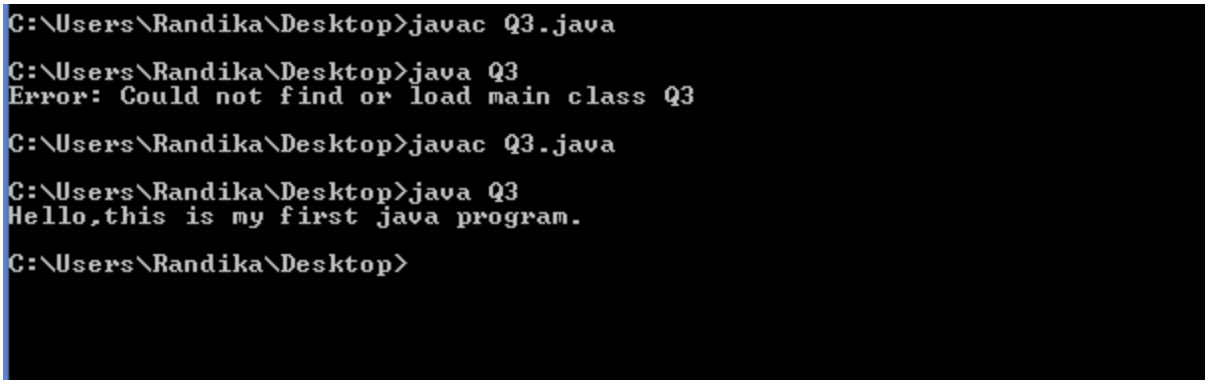


In the 11 th row,provement is there.(\Java\jdk1.8.0…..)

Q3.



Q4.



Q5. ” Java programs are compiled and interpreted”

In programming languages like C and C++, the human readable source code is converted directly into the machine readable binary code or machine code.

But in java,there is an intermediary code called Byte Code. In the compilation of the program,the java compiler converts the source code into Byte code.

In the interpretation process the JVM (Java Virtual Machine)converts the byte code into Binary code or Machine code.

Java is a compiled programming language, but rather than compile straight to executable machine code, it compiles to an intermediate binary form called JVM byte code. The byte code is then compiled and/or interpreted to run the program.

So the java programs are both compiled and interpreted.