Notes 01

Writing a Program:

* Need text editor to write Java program (download notepad++)
  + Written code is called assembly code
* Set up Java Development Kit (JDK)
  + Need to compile and execute code
* Code to print simple text:

**public class firstProg**

**{**

**public static void main(String[]args)**

**{**

**System.out.println("Insert text here");**

**}**

**}**

* Compiling a program:
  + Translate assembly code into Java bytecode for computer to read
  + Install NppExec from Plugin Manager
    - Go to Plugins-> NppExec-> Execute
  + Command to compile program:

**//save the file**

**NPP\_SAVE**

**//change to the current directory**

**cd "$(CURRENT\_DIRECTORY)"**

**//tell notepad ++ where java is**

**C:\Program Files\Java\jdk1.8.0\_05\bin\javac $(FILE\_NAME)**

* Save command as “JAVACOMPILE”
* Executing a program:
  + JVM (Java Virtual Machine) reads the bytecode and translates it into machine code for the computer to understand
  + Command to execute program: (same thing as compile but change the last line to:)

**C:\Program Files\Java\jdk1.8.0\_05\bin\java $(NAME\_PART)**

* + Save command as “JAVAEXECUTE”

Using Escape Characters in Java:

* Some characters (like single or double quotation marks and backslashes) have certain meanings to the computer that don’t allow you to simply type it into text editor as you would for any normal character
  + For example, quotation marks are used to indicate what text should be printed
  + If we want a quotation mark itself to be printed, just typing it regularly as if it were a normal character will result in an error (the quotation mark will not be recognized since it is programmed to have a different function)
* Instead, we need to enter escape sequences for these special characters so that the compiler understands what we want to have printed
* Use escape commands when needed in text, e.g.:
  + Type \” to add “
  + \’ to add ‘
  + \\ to add \

ASCII Art:

* Pictures that are made from keyboard characters on computers
  + Basically, art that is made out of text
  + Formed when methods for printing were still rather primitive, so people needed to use text to create images
* Use multiple lines to make more complex pictures
* Example: simple ASCII art:

( \ \_ / )

( = ‘ . ‘ = )

( “ ) \_ ( “ )

This bunny can be created using the code:

**System.out.println(" ( \\ \_ / ) ");**

**System.out.println("( = \' . \' =)");**

**System.out.println("( \" ) \_ ( \" )");**

* More complex ASCII art:

\_\_\_\_\_\_\_ \_\_

/ ------. / .\_`\_

| / ~--~ \

| | \_\_ `.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_^-----^

| | I=|=======/--\=========================| o o o |

\ | I=|=======\\_\_/=========================|\_o\_o\_o\_|

\| / ~ ~

\ .---. .

-----' ~~''

\_ \_\_

\_\_.--\*\*"""\*\*--...\_\_..--\*\*""""\*-.

.' `-.

.' \_ \

/ .' . \ \_.\_

: : :`\*. :-'.' ;

; ` ; `.) \ /.-'

: ` ; ' -\* ;

:. \ : : : :

; ; `. `. ; ` | '

| `. `. -\*"\*\; / :

| : /`-. `. \/`.' \_ `.

: ; : `\*-.\_\_.-\*""":`. \ ; 'o` `. /

; ; ; \ ;: ;: ,/

| | | /` | , `\*-\*'/

` : : : / / | : . .\_.-'

\ \ , \ : `. : \ \ .'

: \*: ; : |`\*-' `\*+-\*

`\*\*-\*`"" \*---\*