

Topic: OOP 1 - Introduction to Java SE7 Basics

There are 3 parts to this assignment.

1.1 Hello Cenit

1. Using Notepad++ or another plain code editor, create a Java program called *Hello* to display the text 'Hello Cenit'.
2. Add comments that clearly show what each line of the program does (except the braces). Include any unique features of the main method and the purpose of the modifiers to the class and main method.
3. Compile and run your program to ensure that it works.
4. Ensure that it conforms to the requirements of well written java source code. (Neat and readable appearance, indented properly etc.) As it should be written in a plain code editor (i.e. Notepad or Notepad++), it should not have any IDE generated references to a package or system generated comments that do not apply to this program.
5. Save it as Hello.java

1.2 Initials

Using Notepad++ or another plain code editor, write a Java program called *Initials* to output your initials to the screen in the form of a series of 9 lines, using asterisks to generate the outline of the letters.

Example: The initials **KD**

```

**          ***  *****

**          ***   **          **

**   ***      **          **

**  ***      **          **

*****        **          **

**  ***      **          **

**   ***      **          **

**          ***   **          **

**          ***  *****

```

Compile and run the program to ensure that it works.

Ensure that it conforms to the requirements of well written java source code. (Neat and readable appearance, indented properly etc.) and that comments not relevant to this program or references to a package are removed.

1.3 Arguments

In the programs you have encountered so far, you will have seen that the main method used something like this: `public static void main(String[] args)`

The args part of the program can accept values (parameters) when the program is called (run). Write a Java program called ShowArgs using Using Notepad++ or another plain code editor. This java program should accept two command line arguments when you are running the program. It should then output these arguments.

Note: You will need to research outside of the course contents to complete this part of the assignment. Include comments to show that you understand fully what the program is doing.

Compile and run this .java source code from a command line. Place the command line arguments as words after the program name when you are running it with spaces between them, as in apples and oranges below.

Example:

```
javac ShowArgs.java
```

```
java ShowArgs apples oranges
```

```
apples oranges
```

END OF EXERCISES