Text

Description automatically generated

|  |  |  |
| --- | --- | --- |
| **PROG 10082** |  | |
| **Faculty of Applied Computing and Technology** | | **Assignment 1** |

***Work Hard on this and do your own work! This will help you with the exam…***

**Part 1: Worth 8 marks**

1. Create a ***project*** in Netbeans called OP1-<yourname> where <yourname> is your name.
2. Create a ***package*** in the project named after you and call it ‘th1’. All of your code will be placed in this package.
3. Write a class called Assingment1
4. Write a method called hello that:
   1. accepts no arguments and returns no values.
   2. Prints out the word “Hello!”
5. Write a method called helloName that:
   1. Accepts one String argument
   2. Returns no value
   3. Prints out “Hello <name>” where <name> is whatever String was passed into the method
6. Write a method called calculateTheAverage that:
   1. Accepts three doubles
   2. Calculates the average of those 3 doubles
   3. Returns the average.
   4. Note that this method does not print anything.
7. In the main method:
   1. create a variable called name of type String and assign it a value of your name. For example, name would hold “Rich” for me
   2. Call the hello method
   3. Call the helloName method, passing it the value stored in your name variable.
   4. Call the calculateTheAverage method, passing it the following values
      1. 15, 18, 20
   5. Remember that calculateTheAverage does not have any print statements in it but that it returns a double.
   6. You will need to ‘catch’ this returned value and print it out.

You program should have an output that looks like this except that it will have your name instead of mine:

A picture containing text

Description automatically generated

**Part 2: Worth 2 marks**

1. Step through the program with the debugger and list what happens in each line of code.
2. For example:
   1. Was a new local variable created? If so, where did the jvm store it?
   2. Did a new method get called? If so, what happened on the stack?
   3. Did a method end? If so, what happened on the stack and what happened to the local variables that were stored on the current stack frame?
3. I have given you a head-start by showing you the code up until the hello method ends. Note that your line numbers may be different than mine
4. **When you have completed part 1 and part 2, save all java source files (.\*java extensions) and the word document to your Assignment 1 dropbox and submit your assignment.**

|  |  |
| --- | --- |
| Current Stack Frame:line number | Things of interest: (Variables in Stack Frame, what did the program do, was a new stack frame created? Destroyed? Was an object instantiated? Was a variable declared? Is it a local or instance? A reference or primitive? |
| 9 | Declared a variable of type String called myName and placed the value of “Puneet Anand” into it. Since this is a local variable, it is stored in the current stack frame |
| 10 | Called the hello method. Since a new method is being entered, a new current stack frame was created on the stack. The JVM then jumped to line 17 which is the beginning of the hello method. |
| 20 | System.out.println (“Hello”) ran. Since a new method (println) was called, a new stack frame was created on the stack and the jvm jumped to the println method. Once the println method ended, that stack frame was destroyed and the jvm ended up back in the hello method. |
| 21 | The hello method ended, so the current stack frame was destroyed and the jvm jumped back to the main method where it left off |
| 11 | Called the helloName method with the value saved in the string myName. Since this is a new method being called, so a new current stack frame is created on top of the stack. JVM would now jump to line 23 where the helloName method begins. |
| 23 | System.out.println(“Hello “ + name) ran. Since the new method println was called, a new stack frame was made on the stack where jvm jumped to the println method and once the println method ended, the stack frame is destroyed and the jvm shifts back to the helloName method. |
| 28 | helloName method ends here, thus the current stack frame is destroyed and the jvm moves back to the main method at where it left off just after line 11. |
| 13 | The method println is called and a new stack frame is created on the stack, however the method calculateTheAverage is then called, creating yet another new stack frame on the stack and the jvm would jump to line 30, which is the start of the calculateTheAverage method. |
| 34 | Created a new double variable called averageValue which holds the average value of the 3 doubles passed as arguments as a double. Since this is a local variable, it is saved on the current stack frame. |
| 36 | Here instead of calling a method to print out a value, there is a return value of averageVal which isn’t directly printed out but can be accessed on the current stack frame and jvm moves back to line 23. |
| 23 | Here the averageVal method ends and that stack frame is destroyed and the println method also ends after printing the return value of averageVal, thus this stack frame is also destroyed. The program reaches it’s end after and current frames are destroyed, jvm exits. |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |