Print your first and family Name: ____

Since pizza seems a main staple for programmers, we shall write a program to make ordering easier.
To do this, you must to plan, write, document, and test your own java program that does all the following:

1. Ask the user to enter a Pizza size (**S/M/L/H** for Small*[8"]*/Medium*[13"]*/Large*[21"]*/Huge*[34"]*)
2. take in the input as a ***string***
3. *trim* the string, take only it's first letter, convert it to upper case
4. if the string is not equal to "***M***", "***L***" or "**H**" , set it to "***S***"
5. print the size and
6. ask the user to enter the number of toppings (*0-5*)
7. input the number of toppings
8. if the input is invalid or less than 0, set the number to 0
9. if the input is greater than 5, set the number to 5
10. determine and **print** in the following order:
    - 10.1. the diameter
    - 10.2. the base price *(S: $5; M: $8; L: $13; H: $21)*
11. ・ calculate and **print** in the following order:
    - 11.1. the number of square inches; use radius * radius * PI (from the Math class)
    - 11.2. the price *(add $.89 per topping)*
    - 11.3. the price *(not including tax) per* square inch (*to 1/10 cent*)
    - 11.4. the tax *(9.5%)*
    - 11.5. the Total Price, *including tax*

Some things you must do:

12. use jGrasp to create and compile this project. Use ONLY a text/code editor such as notepad or jGrasp.
13. use javadoc headings for each class and method
14. generate javadocs
15. define and use constants instead of "Magic Numbers" for every occurrence of a number
16. initialize all primitive variables
17. **round off** numbers to the nearest cent.
18. print dollar amounts to **two decimal places**
19. check for valid input
20. include examples for all inputs in your test plan
21. use <Alt> + <PrtScn> to capture modest sized, readable images of it running; paste images into a .rtf file (use file save as to select Rich Text Format) and submit that.
22. submit your .rtf file with the .java and .class file(s) for this exercise

Extra credit (10%): document the use of a command line window with: ***javac***, ***javadoc*** and ***java***

The program must be fully planned *in advance*. It must be well documented *(google javadoc as needed)*, and work efficiently, correctly and to specifications.
Make sure to test your program with both valid and invalid inputs for both size and number of toppings--test using both ***numeric*** and ***non-numeric*** inputs, as well as just pressing *<Enter>*.

Deliverables:

Physical:

You must turn in your project on time with pages **stapled in the following order** (preferably two-sided):

A. Print your name on this assignment Sheet (printed from the web), as a cover sheet.

B. Printed Source Code with, no line wrapping and complete javadoc comments
(include heading blocks for each class and method; you must describe all parameters)

C. Sample Input and Output (printed); Show actual input and ALL values output as well as ALL expected output.

D. a simple test plan including explanations of any discrepancies and reasons for each test.

E. Make sure to test non-numeric and out-of-bounds values.

Electronic:

You must also submit:

A. All ***.java, .class***, input and output and image files, ***zipped*** together. **Do not use *rar* or any format other than *zip*.** Rename the file: "<YourName>_p1.zip".
Submit this single ***zip*** file by going to **Canvas**, select this class, select the **Assignment** link, select ***Project 1*** , select the **submission** tab near the top, right of the screen, find the file, and **Submit.** You can re-submit up to the deadline.

Due:

See Canvas for deadline