|  |  |  |  |
| --- | --- | --- | --- |
|  | | | |
|  |  | [ [Home](http://people.uncw.edu/cferner/index.html) ][ [Classes](http://people.uncw.edu/cferner/Classes/ClassesMain.html) ][ [Research](http://people.uncw.edu/cferner/Research/ResearchMain.html) ][ [Links](http://people.uncw.edu/cferner/LinksMain.html) ][ [Biography](http://people.uncw.edu/cferner/BioMain.html) ] |  |
| [Up](http://people.uncw.edu/cferner/Classes/csc434/Assignments.html)  [CS Dept.](http://www.uncw.edu/csc/)     [http://people.uncw.edu/cferner/Classes/csc434/glogo50.gif](http://www.uncw.edu/) |  | **CSC 434**  **Assignment 1 (Fortran) - Due 2/9/2017**  Write a Fortran program that computes π.  A good tutorial to start with fortran is <http://www-teaching.physics.ox.ac.uk/Unix+Prog/hargrove/tutorial_77/>.  The above tutorial will be a good place to start with your first Fortran program.  Note that spacing is very important.  The documentation explains this, but it is easy for Fortran novices to miss this.  The first 6 characters on each line are for line numbers.  Fortran statments should begin at the 7th position (or further) on each line.  There is a Fortran 77 compiler by GNU on the machine babbage.cis.uncw.edu.  To create a Fortran program and compile it, create a file with the extension .f, and use the command gfortran <program>.f -o <program> .  **There is also a GNU Fortran plugin for Eclipse called Photran.**  You should be using Fortran 77 constructs.  In particular, I want you to use the original DO loop, whose syntax is:                **DO <label> I = <expr1>, <expr2> [ , <expr3> ]            ...    <label> CONTINUE**     Do not use the form that has a **while** or an **enddo**. Same thing with the **if** statement.  Use the form that does not have an **endif**.***Basically, I want you to write a program in Fortran 77.  I do not want you to write an object-oriented fortran program.***  You are going to approximate π by using two different methods.  There is a plethora of ways to approximate π.  See [http://en.wikipedia.org/wiki/Approximations\_of\_π](http://en.wikipedia.org/wiki/Approximations_of_%3F) of more information.  **Part I**  Create a function that computes π. using the Monte Carlo method.  You program will compute π by using random numbers.  If we draw a unit circle (radius = 1.0) inside a unit square (2x2), then the ratio of the area of the circle to the area of the square is π/4. Consider the following image:  **7**http://people.uncw.edu/cferner/Classes/csc434/unitCircle.png  The area of the circle is π\*12 and the area of the square is 22 = 4.  The ratio of the area of the circle to the area of the square is then π/4.  We can estate this ratio by generating random points inside the square and keeping track of the percentage of those points that are within the circle to those that are not.  In fact, we don't even need the entire circle.  The same ratio applies if we consider only one quandrant of the graph, such as below:  http://people.uncw.edu/cferner/Classes/csc434/unitCircle2.png  So to compute and estimate of π, one only needs to generate a large number of random points where x ∈ [0,1] and y ∈ [0,1].  The point (x,y) is within the circle if √(x2 + y2) ≤ 1.0.  The ratio of points that are within the circle to the total number of points generated times 4 will be an approximation of π.  The more random points you use the better the approximation.    Run your program using more and more random points to see how close you get get your estimate to π. You will probably need on the order of at least a million points to get a decent approximation. You should compute the error between your estimate and a published version of the value of π = 3.1415926535.  **Part II**  Write a second function to approximate π using the Newton's approximation:  http://people.uncw.edu/cferner/Classes/csc434/NewtonPi.png  Do the same thing as above where you try to get a better and better approxmination of π and compute the error.  You will not (and should not) run the program with a million iterations!  You will only need 20 or 30. *Please use the right most version of this formula that does****NOT****use factorials, especially factorials of factorials!*  **Documentation:**  It is expected that you will follow standard practices of documentation of your program.  That means that classes and methods should have header information include: Author, date written, list and description of parameters or data members (where appropriate), type and description of return values (where appropriate), a general description of the purpose of the class or method.  You should also put in comments for code for which its meaning is not obvious.  That does not mean to put in comments that a loop will loop through some values, but rather put in comments for things that it would take study or searching documentation to understand.  You should also put in comments for constructs that are not common in every language (e.g. use of regular expressions).  And finally, any part of the program that asks the user for input should be proceeded with an appropriate prompt.  An appropriate prompt is one that tells the user what to enter and in what form.  For example, "Enter two floating-point numbers separated by whitespace on one line."  **Deliverables:**   * The source files for the program * Explain the features you used in the program. In particular, explain what the language constructs, libraries, and functions do. * Also, tell me why I asked you not to use factorials.  There are multiple answers to this. * Submit the files to the assignment drop box on Blackboard.  You can get there by logging into UNCW SeaPort, then clicking on the link for the course. |  |
| This page was last updated: **January 26, 2017**  **Email: http://people.uncw.edu/cferner/Classes/csc434/mailme.png**  [ [Home](http://people.uncw.edu/cferner/index.html) ][ [Classes](http://people.uncw.edu/cferner/Classes/ClassesMain.html) ][ [Research](http://people.uncw.edu/cferner/Research/ResearchMain.html) ][ [Links](http://people.uncw.edu/cferner/LinksMain.html) ][ [Biography](http://people.uncw.edu/cferner/BioMain.html) ] | | |  |