

Yolia Simon - Assignment report:

Program 1: Cone calculations

In this program, I used `#include <math.h>` to allow me to use mathematical functions such as 'atan' in my calculations later on. I defined π as a constant equal to 3.14159265 using the variable 'const' and the 'double' data type because 'double' is used to store precise floating-point numbers. I then used a 'scanf' function to ask the user to enter a numerical value for the radius and height of the cone. I used the format specifier %f because it is used for floats. To check if the user inputs are positive, I used an if statement that says 'if the radius/height is equal or less than 0, print an error message and return(1)', where return(1) exits the program. After this, I defined variables for slant height, opening angle (θ), volume and surface area using the 'float' data type. I specified the equations for each variable:

$$\begin{aligned} \text{slant height} &= \sqrt{r^2 + h^2} & \text{opening angle} &= \tan^{-1}(r/h)(180/\pi) & \text{volume} &= \frac{1}{3}\pi r^2 h \\ \text{surface area} &= \pi r(r + h) \end{aligned}$$

Finally, I printed the results of each calculation using a 'printf' function and using the format specifier %.2f to round to 2 decimal places.

Improvements that could be made: Instead of the error message exiting the entire program, you could write some code to make it so that the program allows the user to try entering a correct input again – similar to try and accept statements in Python.

Program 2: Exam marks

In this program, I used `#include <math.h>` to allow me to use mathematical functions in my calculations later on. I created two functions to calculate the mean and standard deviation depending on the input that I can call later in the code:

$$\text{mean} = \frac{\text{total}}{N} \qquad \text{standard deviation } (\sigma) = \sqrt{\frac{\sum(x_i - \mu)^2}{N}}$$

Creating these functions makes the code less bulky when doing repetitive calculations. I opened the exam_marks.txt file in r+ (read and write mode) because using r+ doesn't delete the contents of the file or create an entirely different file. I created an error message to determine whether or not the file is being read. The error message returns (1) which exits the program. I defined 'int' variables to store the physics marks, maths marks and the combination of physics and maths marks. I then read the physics and maths marks from the file. I used the format specifier %*[^:] because it scans everything until the colon, deletes what's been scanned and then reads everything else. This allowed me to skip over all the words in the file and retrieve only the numbers. I used the format specifier %d because it is used for decimal integers. To store the physics and maths marks in the 'combined' array, I created a 'for' loop that loops over 5 times and after each iteration, it assigns values from the physics and maths arrays to the combined array. After this, I defined 'double' variables for the means and standard deviations of the physics marks, maths marks and combined marks. I called the functions I previously created to calculate the standard deviations and means. Then, I closed the file and reopened it in a (append mode) so I could append all the results (rounded to 2 decimal places using the format specifier %.2f) to the end of the file. I also created another error message in case there are issues opening the text file for appending. I used the 'fprintf' function to append the results to the end of the file and prevent the results from being printed to the console. I closed the file and printed a confirmation message to the console to let the user know that the results have been appended.

Improvements that could be made: To make the code cleaner, you could create a function that reads the data. You could also define some constants for fixed numbers like the number of students. To avoid repetitive code, you can create functions.