Bonus assignment week 3–4: Taylor Sequences

Course 'Imperative Programming' (IPC031)

1 Assignment

Similar to the case with computing the square root of a value, many programming languages have built in ways to compute the value of the sinus and cosinus function. These are again approximations of the true values. Two well-known, yet rather inefficient, ways to approximate these functions are the so-called Taylor Sequences (assume $x \in [0, \pi]$):

$$sinus\ (x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^{2n+1}}{(2n+1)!}$$

$$cosinus\ (x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \ldots = \sum_{n=0}^{\infty} \frac{(-1)^n \cdot x^{2n}}{(2n)!}$$

Design and implement the functions:

- `void sinus (double x, double eps, int max_no_steps)` and
- `void cosinus (double x, double eps, int max_no_steps)`

that get as first formal parameter the x value ($x \in [0, \pi]$) of which the sinus (cosinus) needs to be approximated with precision `eps`. At each iteration i, a message is printed that shows i, the current value of the approximation, and the absolute difference between the current approximation and `sin (x)` in case of `sinus` and `cos (x)` in case of `cosinus`. If the approximation is still not sufficiently close to `eps` after `max_no_steps` then the function stops as well.
Obviously, your implementation of `sinus` and `cosinus` should use only the arithmetical operations that are used in the Taylor Sequences. Furthermore:

- Instead of a factorial function ($n!$), your implementation should exploit the fact that `0! = 1` and `(n+1)! = (n+1) * n!`, so you obtain successive factorial values by means of multiplication.
- Instead of a power function (x^n), your implementation should exploit `x^0 = 1` and `x^(n+1) = x * x^n`, so you obtain successive power values by means of multiplication.

To compare the intermediate results with the built in functions `sin` and `cos`, you need to include the `cmath` library:

```
#include <cmath>
```

2 Products

As product-to-deliver you only need to upload to Brighspace "`main.cpp`" that you have created with your solution regarding the bonus assignment.

Deadline

Bonus assignment: Friday September 27, 2024, 23:59h

Important notes:

1. check that you have actually submitted your solution in Brightspace.
2. the deadline is firm, and it is impossible to upload a solution after expiry. In that case you fail the assignment.
3. you can upload solutions to Brightspace several times, but only the last submission is stored.
4. identify yourself and your lab partner in every uploaded document. The identification consists of your first and last names, student numbers, and number of (sub) assignment. By identifying yourself, you declare that the uploaded work has been created solely by you and your lab partner.

5. your work will be judged and commented upon. We expect that you obtain the feedback, read it, and use it to for the next exercises.
6. it is essential that you only submit your own solution, never copy somebody/something else's solution, and never share your solution—in particular: **AI tools (including but not limited to Github Copilot or ChatGPT) are not permitted**, solutions from previous year cannot be reused, and finally, you and your lab partner take joint responsibility for the assignment you submit.