Object-oriented programming in C++

Assignment 1

The Bohr Atom

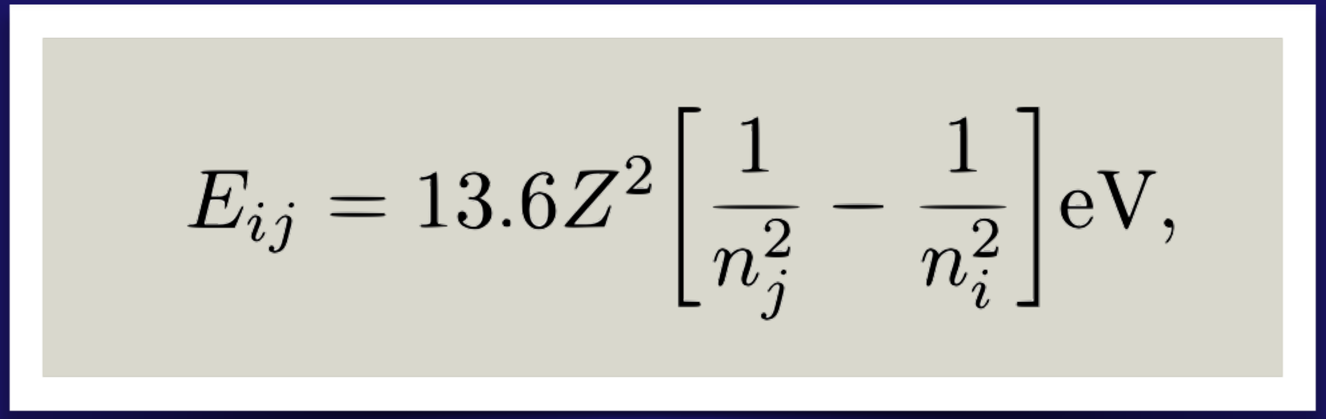
Deadline: February 14th 23:59
Link to the GitHub classroom: https://classroom.github.com/a/QeInPnAM
(It will ask you to associate your GitHub account with your name/student ID)

PHYS30762/63762

Assignment 1 - The Bohr atom

Deadline: February 14th 2359

The Bohr model allows the photon energies of electron transition to be calculated as:



$$E_{ij} = 13.6Z^2\left[\frac{1}{n_j^2} - \frac{1}{n_i^2}\right]\mathrm{eV},$$

where Z is the atomic number, n_i and n_j are the principal quantum numbers (integers) for the initial and final energy states of the electron, respectively.

Write a C++ program to calculate the energy for a transition.

It should:

- Ask the user to enter the atomic number, initial and final quantum numbers, and whether to display the calculated energy of the transition in Joules or electron volts.
- After the calculated value is shown, ask whether to repeat (“y/n”) and stop on “n”.
- Check at each stage for incorrect inputs (either in format or violating some physical conditions).

Assignment 1 - The Bohr atom - test cases

Deadline: February 14th 2359

Here are some examples of how your code should behave that you can use as tests.

How you take the input is up to you, but assuming you take e.g. ‘1 2 1 J’ as input for ‘Z n_i n_j Units’, then:

- An input '1 2 1 J’ would display the result of the calculation in Joules.
- An input ‘1 1 2 e’ would give an error message because $n_i < n_j$.
- An input ‘3 5 1 e’ would display the result of the calculation in electron volts.

At the end of the calculation, ask whether the user wants to calculate another or not, then:

- A response of ‘y’ would repeat the code and ask for new values of the inputs.
- A response of ’n’ would stop the program.

Further instructions

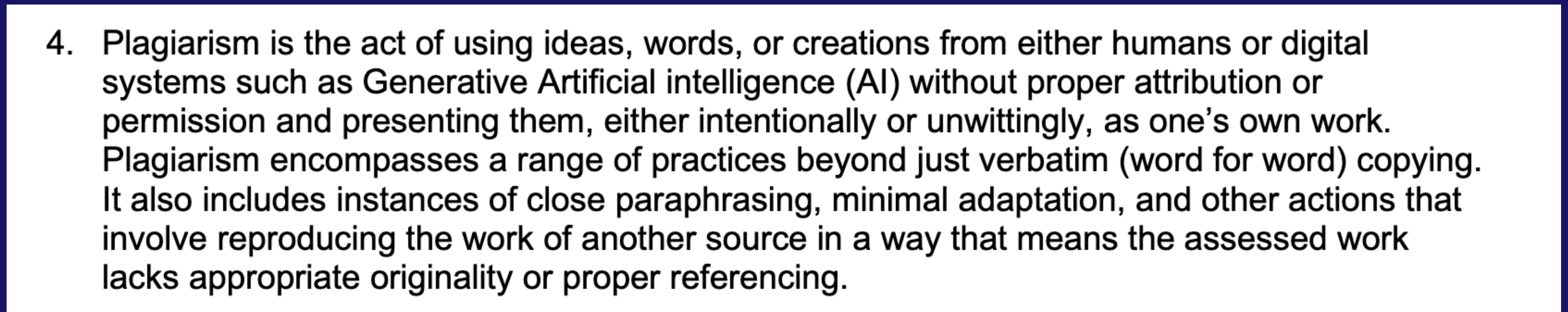
You may use the **skeleton code** provided on Github and Canvas, but you are encouraged to write the program from scratch yourself.

You will need to compile and run your program on the PC Lab computers (Schuster 3.58).

The marking rubric can be found on Canvas. The challenge mark (1/5 points) include a good commit history as well as information about your thought process in the README or similar document.

Assignment submission

Your code must be your own - plagiarism will not be tolerated and will be checked for. This includes Generative AI, where you must disclose its use following the University guidelines:



4. Plagiarism is the act of using ideas, words, or creations from either humans or digital systems such as Generative Artificial intelligence (AI) without proper attribution or permission and presenting them, either intentionally or unwittingly, as one's own work. Plagiarism encompasses a range of practices beyond just verbatim (word for word) copying. It also includes instances of close paraphrasing, minimal adaptation, and other actions that involve reproducing the work of another source in a way that means the assessed work lacks appropriate originality or proper referencing.

https://documents.manchester.ac.uk/display.aspx?DocID=2870

If you use Gen AI, you should use the GAIDeT declaration: https://panbibliotekar.github.io/gaidet-declaration/index.html and detail how you used it. As seen in class, we expect you to understand the code you hand in!

You are free to discuss issues you encounter with each other and with the demonstrators in the lab sessions.

Your code must run successfully in the PC Lab using Visual Studio Code and g++11 with the line provided in the skeleton code by the deadline. You are responsible for this, and if the code doesn't compile with this setup you will lose marks.

Feedback

Marks and feedback will be released 5 working days (+/- 1 day) after the submission deadline.

If anything is unclear, please talk to us in the lab sessions.