

**FCAI**

**CS213**

**Dr,Muhammed Elramly**



Cairo University

|  |  |  |
| --- | --- | --- |
| Name | ID | Task |
| Mahmoud Sayed Mahmoud Younis | 20230372 | Chat-Gpt code, NTI report,free code camp report. |
| Saif Omar Emam Abd-Elhalem | 20230183 | Copilot code , ITI report. |
| Hassan momen Hassan Hosny | 20231047 | AI code analysis and report. |

# Code Report - Task 2

## Brief:

Both codes did the assignment ordered from them with the same prompt to ensure fairness. While one of them performed better in almost every scale, both of them fell in somewhat the same tests, except one did one better than the other.

**AI assistants used in this report were:**  
- Microsoft Copilot  
- ChatGPT  
- Blackbox AI (bonus assistant that fed us with the answers of the test cases in case we got any of them wrong)

## ChatGPT:

### Code:

[GITHUB-LINK](https://github.com/MahmoudSayed538/Assignment_1/tree/main/Assignment_1_T2_T3/chat-gpt_code)

### Explanation:

1. Constructors and Destructor:  
 - The class has three constructors: a default constructor that initializes a zero polynomial, one that accepts a vector of coefficients, and a copy constructor for creating a new polynomial from an existing one.

2. Assignment Operator:  
 - The assignment operator (operator=) allows one polynomial to be equal to another.

3. Arithmetic Operators:  
 - The class implements the addition (operator+), subtraction (operator-), and multiplication (operator\*) of polynomials. These operators create and return new Polynomial objects representing the result of the respective arithmetic operation.

4. Equality Operator:  
 - The equality operator (operator==) checks if two polynomials are identical by comparing their coefficient vectors.

5. Output Operator:  
 - The output operator (operator<<) enables easy printing of polynomial objects in a human-readable format, showing the polynomial in descending order of degree.

6. Utility Functions:  
 - The degree() function returns the degree of the polynomial, which is the size of the coefficient vector minus one. The evaluate(double x) function calculates the polynomial's value at a given point x. The getCoefficient(int degree) function retrieves the coefficient for a specific degree, returning zero if the degree is out of range. The setCoefficients(const vector<double>& coefficients) function updates the polynomial's coefficients.

### Testing the Polynomial Class

The runTests() function is designed to test various functionalities of the Polynomial class. It includes a series of tests that cover:  
- Creation of default and specific polynomials.  
- Basic arithmetic operations like addition, subtraction, and multiplication, with expected outcomes printed to the console.  
- Evaluation of polynomials at specific points.  
- Checking polynomial degrees and coefficients.  
- Handling of polynomials with negative coefficients and large coefficients.  
- Equality checks between identical and different polynomials.  
- Testing edge cases like evaluating at zero and composing with empty polynomials.  
- Mock calculations for definite integrals and polynomial roots, even though the actual root-finding method is not implemented in this code.

## Copilot:

### Code:

[GITHUB-LINK](https://github.com/MahmoudSayed538/Assignment_1/tree/main/Assignment_1_T2_T3/copilot_code)

### Explanations:

1. Constructors and Destructor:  
 - The class features three constructors:  
 1. Default Constructor: Initializes a zero polynomial with a single coefficient of 0.0.  
 2. Coefficient Vector Constructor: Accepts a vector of coefficients to create a polynomial based on user input.  
 3. Copy Constructor: Creates a new polynomial by copying the coefficients from another polynomial instance.  
 - The destructor is defined but does not perform any specific actions, as the class does not manage dynamic memory directly.

2. Assignment Operator:  
 - The assignment operator: (operator=) allows one polynomial object to be assigned to another. It includes a check for self-assignment to prevent unnecessary operations. If the polynomials are different, it copies the coefficients from the source polynomial.

3. Arithmetic Operators:  
 - The class implements three arithmetic operators:  
 - Addition (operator+): Computes the sum of two polynomials by creating a new polynomial with the resulting coefficients.  
 - Subtraction (operator-): Similar to addition, it computes the difference between two polynomials.  
 - Multiplication (operator\*): Multiplies two polynomials together, resulting in a new polynomial whose coefficients are determined by the product of the individual coefficients.

4. Equality Operator:  
 - The equality operator(operator==) checks if two polynomial objects are identical by comparing their coefficient vectors for equality. This allows for straightforward comparison between polynomial instances.

5. Output Operator:  
 - The output operator(operator<<) is overloaded to enable easy printing of polynomial objects. It formats the polynomial in a human-readable way, displaying the coefficients in descending order of degree while omitting any terms with a zero coefficient.

6. Utility Functions:  
 - The class includes several utility functions:  
 - degree(): Returns the degree of the polynomial, which is the number of coefficients minus one.  
 - evaluate(double x): Evaluates the polynomial at a specific value of x using Horner's method for efficiency.  
 - getCoefficient(int degree): Retrieves the coefficient for a specified degree, returning zero for degrees outside the range of the polynomial.  
 - setCoefficients(const vector<double>& coefficients): Updates the coefficients of the polynomial.

### Testing the Polynomial Class:

The runTests() function is designed to validate the functionalities of the Polynomial class. It covers a variety of test cases, including:  
- Creation of both default and specific polynomials.  
- Execution of arithmetic operations like addition, subtraction, and multiplication, with expected outcomes printed for verification.  
- Evaluation of polynomials at various points to ensure correct computation.  
- Checking the degree of polynomials and accessing specific coefficients.  
- Handling special cases like polynomials with negative or large coefficients.  
- Verifying equality between identical and different polynomial instances.  
- Testing edge scenarios, such as evaluating at zero and composing with an empty polynomial.  
- Mock calculations for definite integrals and polynomial root finding, though actual root-finding logic is not implemented in this version.

### Main Function:

The \*\*main()\*\* function serves as the entry point of the program. It invokes runTests() to execute the defined tests. After all tests are run, a completion message is displayed to indicate that the testing process has finished.

# Analysis

**1. Readability :**

ChatGPT(28/30):

* The code is neatly structured with clear function and variable names. It uses comments to explain complex parts, enhancing understanding.

CoPilot(25/30):

* The code is more concise. It avoids redundancies, The code follows a consistent and minimalist approach, using functions effectively. it lacks detailed comments or explanations in some areas, making it harder to understand.

**2. Simplicity:**

ChatGPT(25/30):

* Follows a step-by-step approach with an explicit handling of terms in polynomial operations. The use of for-loops is straightforward.
* The redundancy adds unnecessary complexity, especially in operations like add() and multiply(). Each method involves multiple steps that are not needed and could be condensed.

CoPilot(28/30):

* The code is much simpler and more efficient. By reducing repeated logic and combining steps, it achieves the same outcome with fewer lines of code.

**3. Performance (Speed):**

ChatGPT(25/30):

* Due to the redundancy in the add() and multiply() functions, the first code performs more computations than necessary. The multiple iterations and explicit handling of each term in loops make it slower when dealing with larger polynomials.

CoPilot(28/30):

* The code reduces the number of unnecessary computations by using efficient looping constructs and reusing logic. This results in faster execution.

**4.Correctness:**

ChatGPT(14/20):

* Failing 6 out of 30 test cases (80% pass rate) indicates issues in edge cases or specific logic, reducing the reliability
* failed cases ( 12 , 19 , 21 , 23 , 25 , 28).

CoPilot(16/20):

* With only 5 out of 30 test case failures (83% pass rate), the second code is slightly more reliable, though still needing improvement.
* failed cases (12 , 19 , 21 , 23 , 24).

**Conclusion:**

* With ChatGPT getting an 83% in general performance and CoPilot getting an 88% it’s clear that copilot proved to be superior in certain parts.
* Copilot also added a couple of functions that GPT lacked like integral, definite integral and root finding.
* Copilot was different in handling the output as if the output was only 0 it would print out an empty line.
* Both of them used mostly the same logic at the beginning but each went with their own way of implementing it.
* Generally AI can be a useful tool in coding but it shouldn’t be a primary source for code unless it’s understood and revised with caution.

# Training research-Task 3

## Automotive Embedded Systems

**why this track:**

I chose to write about this specific training track because I have always been amazed by the integration of software and hardware and the endless creativity that it offers, also studying embedded systems can be a fun and creative way to understand computer architecture on a very deep level.

### Who Offers It and Where It Is Located

This track is a part of the Electronics and Embedded Systems category in [DEY-Egypt Makes Electronics (EME)](https://www.nti.sci.eg/dey/eme.html) offered by the NTI (National Telecommunication institute). The initiative offers training in various tracks under categories like:

* + - 1. Cybersecurity.
      2. Devops Engineering.
      3. software Engineering.

**About the NTI :**

The National Telecommunication Institute (NTI), established in 1983, excels in Egypt's education, research, and technical consultation. Aligned with the Ministry of Telecommunication and Information Technology (MCIT), NTI recognizes ICT's crucial role in the Egyptian societal advancement. It plays a key role in Egypt's Vision 2030 for digital transformation, supported by the government and industry collaboration. NTI prioritizes high-quality education, training, and innovative research, offering consultancy and technical services.

**NTI Sites in Egypt :**

1. Nasr City Campus .
2. Smart Village Campus.
3. Knowledge City Campus (New Capital, Egypt).

### What It Offers

* This track introduces a variety of topics to build trainees’ skills and understanding of embedded systems and familiar with the basic concepts and terminology of the target area, the embedded systems design flow. Understanding of the embedded system architecture. Build embedded system solutions, moreover, understand how Microcontrollers communicate with other devices using standard communication protocols like “UART, SPI, and I2C”. To develop software for embedded computer systems using a real time operating system also Write driver for any peripherals for any Microcontroller. Develop an understanding of software process models and the ability to select the suitable model to use in software development. Understand the embedded Linux important topics such as (Qemu, toolchain, bootloader, kernel, and root file system). Understand concepts, issues, principles, and mechanisms in embedded systems security such as embedded security trends, software vulnerabilities, physical attacks, and security policies. Obtain hands-on skills in securing practical embedded systems.
* This track targets persons who hope to become embedded system engineers, Software testing engineers and Embedded Linux Engineers.

### When It Opens, Duration, and Mode (Online/Offline)

Registration is currently closed, and expected to be opened during summer, the Duration of the training is

4 months in an offline study mode.

### Conditions for Application, Acceptance, Fees, and Other Conditions

1. Commitment to submit all required documents on the initiative's electronic platform upon approval of these terms and regulations is a fundamental requirement for scholarship acceptance:
2. Copy of the graduation certificate.
3. Scanned copy of the national ID card.
4. Signed declaration by the trainee.
5. Compliance with the fundamental requirements of each training program is a prerequisite for acceptance into the scholarship.
6. Commitment to the scheduled training hours (6 training hours daily).
7. Adherence to the prescribed attendance percentage of not less than 75%.
8. Commitment to appropriate behavior and professional etiquette with everyone during the training period within the initiative.
9. Commitment to preserving academic materials and refraining from posting them on electronic platforms (intellectual property rights).
10. Commitment to the accuracy of all information stated in the declaration.
11. Commitment to submitting a new signed declaration for each training course the trainee enrolls in.
12. Commitment to providing a notification or sending an apology via email for non-completion of the scholarship, at least 48 hours before the commencement of the training.
13. Only communications through the official email of the initiative (D4m@nti.sci.eg) will be considered.
14. There are no fees once accepted in the initiative.

### Different Learning Tracks

**other training tracks that the initiative offers are :**

1. Network Infrastructure and Enablers.
2. Cybersecurity.
3. Devops Engineering.
4. Telecom Engineering.
5. Software Engineering.

## Machine Learning with Python

**why this track:**

I chose this training track because Machine learning has many practical applications that you can use in your projects or on the job, and now most modern technology function with AI integration, for me I have always hada passion for robotics and embedded systems and machine learning can be a great way to add more functionality to my projects in the future.

### Who Offers It and Where It Is Located

this training is offerd by [freeCodeCamp](https://www.freecodecamp.org/learn)

**About freecodecamp.org :**

freeCodeCamp is a donor-supported charitable organization their message is to help people learn to code for free. We accomplish this by creating thousands of videos, articles, and interactive coding lessons - all freely available to the public. Donations to freeCodeCamp go toward their education initiatives, and help pay for servers, services, and staff.

### What It Offers

In the Machine Learning with Python Certification, you'll use the TensorFlow framework to build several neural networks and explore more advanced techniques like natural language processing and reinforcement learning.

You'll also dive into neural networks, and learn the principles behind how deep, recurrent, and convolutional neural networks work, also you get a free certification in the end.

The training walks you through several courses like :

1. TensorFlow.
2. How Neural Networks Work.
3. Machine Learning with Python Projects

The best thing offered by the training for me is the active learning style it walks you through, meaning that you get to learn by doing projects and following along.

### 3. When It Opens, Duration, and Mode (Online/Offline)

The training is available on freecodecamp.org completely free and online only which makes learning at your preffered pace.

### 4. Conditions for Application, Acceptance, Fees, and Other Conditions

There are no conditions nor fees for applying for the training, however there are some conditions that govern the use of the website :

* + - 1. You must be at least thirteen years old.
      2. You may no longer use the website if the company contacts you directly to say that you may not.
      3. You must use the website in accordance with Acceptable Use and Content Standards.

### Different Learning Tracks

**other training tracks that the website offers are :**

1. Responsive Web Design Certification.
2. Front End Development Libraries Certification
3. Information Security Certification.