Cairo University

Faculty of Computers and Artificial Intelligence

CS213: Object Oriented Programming

Assignment (1)

Task (2) & Task (3)

Course Instructor:

Dr. Mohammad El-Ramly

Team Members:

- Esraa Emary Abd El-Salam 20230054
  - esraaemary33@gmail.com
- Mohammed Atef Abd El-Kader 20231143
  - mohamed45452020@gmail.com
- Nagham Weal Mohamed Elsayed 20231189
  - naghamw63@gmail.com

Table Of Content

Cairo University

Faculty of Computers and Artificial Intelligence

CS213: Object Oriented Programming

Assignment (1)

Task (2)

Course Instructor:

Dr. Mohammad El-Ramly

Introduction

Artificial intelligence (AI) has become an integral part of modern society, influencing nearly every sector and aspect of our lives. From revolutionizing healthcare and automating industries to enhancing customer experiences and driving scientific advancements, AI offers immense potential to solve complex problems efficiently. Its ability to process vast amounts of data, identify patterns, and make intelligent decisions is transforming the way we live and work.

One area where AI is having a particularly significant impact is programming. Traditionally, coding required manual effort to design, develop, and debug applications, but AI tools have introduced new levels of automation and intelligence into the programming process. AI-powered programming assistants, such as prompt-based models, are changing how developers approach their work.

Prompt AI, a recent innovation in the field of artificial intelligence, allows programmers to interact with intelligent systems by providing natural language prompts to generate code, solve problems, and enhance productivity. This approach makes coding more accessible, reducing the need for deep expertise in complex algorithms or syntax. AI tools like GitHub Copilot, ChatGPT, and others assist developers in writing clean code, debugging, and even explaining code structures—all based on prompts provided by the user.

By leveraging prompt AI, developers can streamline workflows, automate mundane coding tasks, and explore new levels of creativity. Whether it's generating boilerplate code, suggesting optimizations, or automating testing, prompt-based AI brings a higher level of efficiency and support, allowing programmers to focus on problem-solving and innovation rather than manual coding efforts.

In this document we used Chat GPT and Copilot for generating functions for the header provided and we will talk about this experience in detail.

First Code's Explanation (Chat GPT)

ChatGPT is an advanced AI language model developed by OpenAI, designed to understand and generate human-like text in a wide range of contexts. It can engage in meaningful conversations, providing detailed responses and helpful insights based on the input it receives. Whether it's answering questions, explaining complex topics, assisting with coding, or generating creative content, ChatGPT offers a versatile platform for users across various fields. Its ability to respond to prompts makes it especially useful for tasks like brainstorming ideas, solving technical problems, or simplifying complex concepts. With its intuitive interface, ChatGPT allows users to interact naturally, using plain language, which makes it accessible even to those without deep technical knowledge. This blend of intelligence and ease of use makes ChatGPT a powerful tool for enhancing productivity, creativity, and learning in everyday tasks and professional settings alike.

In this assignment we used Chat GPT for Prompt AI, and it helps very well. At first, we put the header file, which is provided in assignment in Chat GPT and we told him to generate functions for these prototypes in header and he did it but some of them works properly and some when we enter some input it output wrong answer so we asked chat GPT to edit more times and it takes much time from us.

One of these functions was (getRoot) function. at first the ai forgot to call the function in main then we asked him that you have an error and you forgot to add this function in main, after that, the function doesn't work at all so we fall in a long dialog with Chat GPT asking him that the function doesn't work at all, and after an hour from trying we could get a function that works well for some cases ,so, I told him that ,but he damaged the code and we couldn't get a better version than which works for some cases so we use it now.

Another function was (Degree) function. At first, ai didn't understand what we need as we want when we enter the coefficient of the two polynomials to tell us the degree of each one, but he misunderstood us as he asked us to enter a coefficient

and he till us - if this coefficient exist - the power of the variable(x) which is multiplied with him, and if this coefficient doesn't exist it output out of range. So, we tried with the ai and told the problem, and we got another function which works properly for most test cases.

The Explanation:

The code is an implementation of the class Polynomial supports a variety of operations, such as addition, subtraction, multiplication, evaluation, differentiation, integration, and root-finding using Newton's method.

Here is an explanation for the code from Chat GPT:

- **Constructors and Destructor**
    - Default Constructor: **`Polynomial::Polynomial() : coeffs (1, 0) {}`**
        - This constructor initializes the polynomial as a zero polynomial with one coefficient, 0. The **`coeffs`** vector is initialized with one element set to 0, which means the polynomial is effectively P(x) = 0.
    - Constructor with Coefficients: **`Polynomial::Polynomial (const vector<double>& coefficients): coeffs(coefficients) {}`**
        - Initializes the polynomial with the provided coefficients from the input vector.
    - Copy Constructor: **`Polynomial::Polynomial (const Polynomial& other) : coeffs (other. coeffs) {}`**
        - Creates a new polynomial as a copy of an existing polynomial.
    - Destructor: **`Polynomial::~Polynomial() {}`**
        - Destructor for the polynomial class; it cleans up resources when a polynomial object is destroyed.

- **Assignment Operator**
    - Assignment Operator: **`Polynomial& Polynomial::operator=(const Polynomial& other) {}`**
        - Overloads the assignment operator to copy coefficients from one polynomial to another, ensuring self-assignment is handled properly.
- **Polynomial Operations**
    - Addition of Two Polynomials: **`Polynomial Polynomial::operator+(const Polynomial& other) const {}`**
        - Adds two polynomials by summing their corresponding coefficients.
    - Subtraction of Two Polynomials: **`Polynomial Polynomial::operator-(const Polynomial& other) const {}`**
        - Subtracts one polynomial from another by subtracting their corresponding coefficients.
    - Multiplication of Two Polynomials: **`Polynomial Polynomial::operator*(const Polynomial& other) const {}`**
        - Multiplies two polynomials using the distributive property to compute the coefficients of the resultant polynomial.
    - Equality Operator: **`bool Polynomial::operator==(const Polynomial& other) const {}`**
        - Checks for equality between two polynomials by comparing their coefficient vectors.
- **Output Operator**
    - Output Operator: **`ostream& operator<<(ostream& out, const Polynomial& poly) {}`**
        - Outputs the polynomial in a human-readable format, including proper handling of positive and negative coefficients.

- **Utility Functions**
    - Degree of the Polynomial: **`int Polynomial::degree() const {}`**
        - Returns the highest degree of the polynomial by iterating from the end of the coefficient vector.
    - Evaluate the Polynomial: **`double Polynomial::evaluate(double x) const {}`**
        - Evaluates the polynomial at a given value of xxx using Horner's method for efficient computation.
    - Derivative of the Polynomial: **`Polynomial Polynomial::derivative() const {}`**
        - Computes the derivative of the polynomial, returning a new polynomial representing the derivative.
- **Polynomial Composition**
    - Composition of Polynomials: **`Polynomial Polynomial::compose(const Polynomial& q) const {}`**
        - Composes this polynomial with another polynomial q, effectively substituting q for each xxx in the polynomial.
- **Integration**
    - Indefinite Integral: **`Polynomial Polynomial::integral() const {}`**
        - Computes the indefinite integral of the polynomial, returning a new polynomial representing the integral.
    - Definite Integral: **`double Polynomial::integral(double x1, double x2) const{}`**
        - Calculates the definite integral of the polynomial from x1x1x1 to x2x2x2 by evaluating the indefinite integral at both limits.

- **Root Finding with Newton's Method**
    - Finding a Root: **`double Polynomial::getRoot(double guess, double tolerance, int maxIter) {}`**
        - Implements Newton's method to find a root of the polynomial. The algorithm iteratively improves the guess by using the polynomial's evaluation and its derivative, stopping when the improvement is less than the specified tolerance or when the maximum iterations are reached.
- **Setters and Getters**
    - Set Coefficients: **`void Polynomial::setCoefficients(const vector<double>& coefficients) {}`**
        - Sets the polynomial's coefficients directly by copying the input vector.
    - Get Coefficient: **`double Polynomial::getCoefficient(int degree) const{}`**

Returns the coefficient for the specified degree of the polynomial. If the degree is larger than the highest degree of the polynomial, it **`returns 0`**.

Second Code's Explanation
(Copilot)

GitHub Copilot is an AI-powered coding assistant developed by GitHub in collaboration with OpenAI. It acts as a copilot for developers, helping them write code more efficiently by suggesting lines or blocks of code based on natural language prompts or partially written code. Copilot can understand the context of what the developer is working on, offering relevant suggestions for functions, methods, and even entire algorithms. Its ability to provide real-time code completions and recommendations makes it a valuable tool for speeding up development, reducing repetitive tasks, and improving code quality. By learning from a vast range of open-source codebases, Copilot can assist with a wide variety of programming languages and frameworks. This seamless integration into the coding environment helps developers focus more on solving problems and building features, rather than worrying about syntax or boilerplate code. With Copilot, programming becomes more intuitive, collaborative, and productive, allowing developers of all skill levels to work more efficiently.

In this assignment we used copilot for Prompt AI, and it helps very well. At first, we put the header file, which is provided in assignment in copilot, and we told him to generate functions for these prototypes in header and he did it but some of them work properly and some when we enter some input it output wrong answer, so we asked Copilot to edit more times, and it takes much time from us.

One of these functions was **`(getRoot)`** function. at first the ai forgot to call the function in main then we asked him that you have an error and you forgot to add this function in main, after that, the function doesn't work at all so we fall in a long dialog with copilot asking him that the function doesn't work at all, and after an hour from trying we could get a function that works well for some cases ,so, I told him that ,but he damaged the code and we couldn't get a better version than which works for some cases so we use it now.

The Explanation:

The code is an implementation of the class Polynomial supports a variety of operations, such as addition, subtraction, multiplication, evaluation, differentiation, integration, and root-finding using Newton's method.

Here is an explanation for the code from Copilot:

1. Constructors and Destructor

- Default Constructor: **`Polynomial::Polynomial() : coeffs({0})`**
    - Initializes a polynomial with a constant term of zero. The **`coeffs`** vector stores the coefficients of the polynomial, where each element corresponds to the coefficient of a particular degree.
- Parameterized Constructor: **`Polynomial::Polynomial(const vector<double>& coefficients)`**
    - Initializes the polynomial with a specific set of coefficients provided as a vector. Each index of the vector represents the coefficient of the corresponding degree.
- Copy Constructor: **`Polynomial::Polynomial(const Polynomial& other)`**
    - Creates a new polynomial that is a copy of another polynomial (other). It copies the coefficients vector.
- Destructor: **`Polynomial::~Polynomial()`**
    - Since no dynamic memory allocation is used, this destructor is just a placeholder.

2. Operator Overloading

- Assignment Operator: **`Polynomial& Polynomial::operator=(const Polynomial& other)`**
    - This function allows assignment of one polynomial to another. It ensures that the current object is not assigned to itself and then copies the **`coeffs`** vector from the other polynomial.
- Addition: **`Polynomial Polynomial::operator+(const Polynomial& other) const`**
    - Adds two polynomials by adding their corresponding coefficients. The result is stored in a new vector with size equal to the larger of the two polynomials, handling polynomials of different degrees.
- Subtraction: **`Polynomial Polynomial::operator-(const Polynomial& other) const`**
    - Like the addition operator, but it subtracts the coefficients of the second polynomial from the first.
- Multiplication: **`Polynomial Polynomial::operator*(const Polynomial& other) const`**
    - Multiplies two polynomials. The result is a polynomial whose degree is the sum of the degrees of the two input polynomials. The multiplication is done by multiplying each term of one polynomial with each term of the other.
- Equality: **`bool Polynomial::operator==(const Polynomial& other) const`**
    - Checks if two polynomials are equal by comparing their coefficient vectors.

- Output Operator: **`ostream& operator<<(ostream& out, const Polynomial& poly)`**
    - This is a friend function that allows the polynomial to be printed in human-readable form. It formats the output, skipping zero coefficients, and adds the appropriate " x^n " for each term.

3. Polynomial Operations

- Degree: **`int Polynomial::degree() const`**
    - Returns the degree of the polynomial, which is the highest exponent with a non-zero coefficient.
- Evaluate: **`double Polynomial::evaluate(double x) const`**
    - Evaluates the polynomial at a given value of x using Horner's method, which is an efficient algorithm for polynomial evaluation.
- Compose: **`Polynomial Polynomial::compose(const Polynomial& q) const`**
    - Composes two polynomials. It replaces every x in the current polynomial with the polynomial q, creating a new polynomial as a result.

4. Calculus-related Operations

- Derivative: **`Polynomial Polynomial::derivative() const`**
    - Returns the derivative of the polynomial. It computes the derivative by multiplying each coefficient by the degree of its corresponding term and then reduces the degree by one.
- Indefinite Integral: **`Polynomial Polynomial::integral() const`**
    - Returns the indefinite integral (anti-derivative) of the polynomial by dividing each coefficient by its degree plus one and increasing the degree of each term.

- Definite Integral: **`double Polynomial::integral(double x1, double x2) const`**
    - Computes the definite integral of the polynomial between x1 and x2. It first computes the indefinite integral and then evaluates it at the two bounds (x2 and x1), subtracting the results to get the area under the curve between the two points.

5. Root Finding with Newton's Method

- getRoot: **`double Polynomial::getRoot(double guess, double tolerance, int maxIter)`**
    - Implements Newton's method for finding a root (solution where the polynomial equals zero) starting from an initial guess. The algorithm iteratively improves the guess by applying the formula:

$$x_n + 1 = x_n - \frac{f(x_n)}{f'(x_n)}$$

    - Where $f'(x_n)$ is the derivative of the polynomial at x_n. The method stops when the improvement is smaller than the given tolerance or when the maximum number of iterations is reached. It includes a check to avoid division by zero if the derivative is too small.

6. Utility Functions

- Set Coefficients: **`void Polynomial::setCoefficients(const vector<double>& coefficients)`**
    - Sets the polynomial's coefficients directly by copying the input vector.
- Get Coefficient: **`double Polynomial::getCoefficient(int degree) const`**

Returns the coefficient for the specified degree of the polynomial. If the degree is larger than the highest degree of the polynomial, it **`returns 0`**.

Comparison Result

All test cases have been passed except the following:

- **Chat GPT** -> Pass: 87, Failed: 13

Test Cases	Input	Operation	Expected	Output
9	0 -4 -5	Derivative	0	NO OUTPUT (Empty)
12	0 5 1	Derivative	0	NO OUTPUT (Empty)
15	0 -5 -3	Derivative	0	NO OUTPUT (Empty)
27	0 -9 12	Derivative	0	NO OUTPUT (Empty)
44	9 11 1 6 9 12 -13 5 -1 -6 -2 11 14 -8 2 -10 14 -10 9 9 12	Get Root	0	6.13708e-11
51	2 -13 1 3 -14 -2 11	Get Root	NO OUTPUT	3.22143e-09
52	2 -14 0 -9 -2 -11 0	Get Root	0	NO OUTPUT (Empty)
65	0 -11 12	Derivative	0	NO OUTPUT (Empty)
71	0 11 8	Derivative	0	NO OUTPUT (Empty)
74	0 7 6	Derivative	0	NO OUTPUT (Empty)
83	2 14 -14 -1 1 0 12	Get Root	[-14.937253933193, 0.937253933193772]	0.937254

92	0 7 1	Derivative	0	NO OUTPUT (Empty)
94	9 0 -2 -11 -8 5 9 15 9 12 -1 3 10 11 -1 10 -12 8 -6 10 -2	Get Root	[-0.679150358373029, -0.220914108533742, 0, 0.800687464725281, 12.7991868727897]	[-0.67915, -0.220914, 6.06332e-12, 0.800687]

- **Copilot** -> Pass: 82, Failed: 18

Test Cases	Input	Operation	Expected	Output
5	9 -1 -2 8 15 -2 -6 14 12 12 1 15 0 -7 11 -2 14 8 7 -13 -3	Get Root	[-11.0314953036864, - 0.327602362497162, 0.361037479450929]	[-0.327602, 0.361037]
9	0 -4 -5	Derivative	0	NO OUTPUT (Empty)
12	0 5 1	Derivative	0	NO OUTPUT (Empty)
13	10 0 -8 13 -1 -13 5 15 -6 -13 -13 15 3 10 -9 -5 0 -5 -8 11 -5 0 6	Get Root	[0, 1.36622405076568]	[-7.5375e-18, 1.36622]
15	0 -5 -3	Derivative	0	NO OUTPUT (Empty)
14	0 -1 -4	Derivative	0	NO OUTPUT (Empty)
27	0 -9 12	Derivative	0	NO OUTPUT (Empty)
45	11 0 -10 -11 -7 9 -4 9 -4 -12 -4 -8 -10 7 -9 -8 -13 -7 3 -10 9 7 -9 15 -12	Get Root	0	4.14301e-21
48	3 1 -4 -11 1 -9 -8 -11 9	Get Root	[-0.515729471589257, 0.170915188827179, 11.3448142827621]	[-0.515729, 0.170915]

54	3 0 -7 -14 -13 -6 10 5 7	Get Root	0	2.07552e-17
65	0 -11 12	Derivative	0	NO OUTPUT (Empty)
71	0 11 8	Derivative	0	NO OUTPUT (Empty)
74	0 7 -6	Derivative	0	NO OUTPUT (Empty)
78	3 0 1 -1 10 0 -8 9 15 2 -6	Get Root	0	-1.18329e-30
83	2 14 -14 -1 1 0 12	Get Root	[-14.9372539331938, 0.937253933193772]	0.937254
92	0 7 1	Derivative	0	NO OUTPUT (Empty)
96	9 0 -2 -11 -8 5 9 15 9 12 -1 3 10 11 -1 10 -12 8 -6 10 -2	Get Root	[-0.679150358373029, -0.220914108533742, 0, 0.800687464725281, 12.7991868727897]	[-0.67915, -0.220914, 2.022e-22, 0.800687]

Test Cases Analysis

▪ Efficiency Comparison: GitHub Copilot vs. ChatGPT

When comparing the efficiency of the code generated by GitHub Copilot and ChatGPT, we observe key differences in their approaches to code complexity, performance, and overall optimization.

GitHub Copilot tends to generate solutions that are generally less complicated and have lower time complexity. This simplicity in code structure makes it more efficient in terms of execution speed, especially when dealing with tasks that require fast runtime. For example, when Copilot generated the `(getRoot)` function, the implementation was straightforward and executed in a manner that minimized the use of complex data structures or recursive calls, which can often lead to increased computational overhead. This streamlined approach is highly beneficial for tasks that need to be completed quickly, without sacrificing too much processing power.

However, a downside to Copilot's focus on simplicity is that, in some cases, the generated code may not be fully optimized for accuracy or edge cases, particularly when dealing with more intricate problem sets. The `(getRoot)` function, while efficient in execution, lacked precision and did not handle certain inputs as accurately as desired. This suggests that Copilot's prioritization of time complexity and performance sometimes comes at the cost of correctness, which could negatively impact on the overall reliability of the solution in complex scenarios.

On the other hand, ChatGPT typically produces more sophisticated and layered solutions. While these approaches may involve a higher level of complexity in terms of the algorithms or structures used, they are often accompanied by detailed comments and explanations that guide the user through the logic behind the code. For instance, in ChatGPT's version of the `(getRoot)` function, the code leveraged more advanced techniques to ensure accurate results across a wider range of cases.

Although this led to a more complex implementation, the result was often more robust and reliable. The trade-off, however, is that such complexity can introduce additional time overhead and may not be the most efficient in terms of raw performance. ChatGPT's emphasis on thoroughness may increase development time and lead to code that runs slower compared to Copilot's minimalist solutions.

In summary, GitHub Copilot excels in generating efficient, low-complexity code that is quick to execute and easy to understand, but it may occasionally sacrifice accuracy and robustness. ChatGPT, while producing more intricate and sometimes less efficient code, compensates with accuracy and insightful comments, making it more suitable for tasks that require in-depth handling of various edge cases and detailed explanations. The choice between the two AI tools depends heavily on the user's priority: speed and simplicity versus accuracy and thoroughness.

- **Code Quality Comparison: GitHub Copilot vs. ChatGPT**

Both GitHub Copilot and ChatGPT deliver high-quality code that is well-structured, readable, and follows best practices. They excel in generating organized code with clear variable names, making the output easy to understand and maintain.

The key distinction between the two lies in comments and documentation. ChatGPT provides detailed comments that explain the logic behind the code, offering insight into both the 'how' and the 'why.' This makes ChatGPT's code more accessible, especially for beginners or when collaborating on complex projects. GitHub Copilot, while still producing clean code, typically includes fewer comments, which may require additional effort to document the logic manually.

In summary, while both tools generate solid code, ChatGPT adds value through its thorough documentation, making it more user-friendly for long-term maintenance and clarity.

▪ Analysis of Failures in `(getRoot)` Function

▪ GitHub Copilot Implementation:

1. Handling of Small Derivatives:
   - Copilot includes a check for very small derivatives (`if (abs(y_prime) < tolerance) break;`). However, if the derivative is too small and it breaks out of the loop without returning a valid result, the function may `return x0`, which could be an incorrect or non-real root. This could lead to situations where the output is not zero when it should be, as it does not adequately address the case when a root may be very close to the guest.

2. Early Loop Exit:
   - The condition to exit the loop early (`if (abs(x1 - x0) < tolerance)`) may lead to prematurely terminating the iteration if the new guess is close enough to the old guess without verifying the accuracy of the found root (`if (abs(evaluate(x1)) < tolerance))`. This can cause the function to return a value that is not sufficiently close to zero, or to miss returning zero altogether if it stops iterating too early.

3. Lack of Return Value for Non-Real Roots:
   - If a valid root is not found within the maximum number of iterations, the function returns the last guess (`return x0;`). If `x0` was never close to the true root, this can mislead users into believing a root was found when it was not.

- **ChatGPT Implementation:**
    1. Return NaN for Small Derivatives:
        - ChatGPT uses return `numeric_limits<double>::quiet_NaN();` when the derivative is too small. This indicates a failure to compute a valid root, which is a better indication than returning a potentially incorrect guess. However, it does not differentiate between situations where a valid root exists but is just hard to find due to the iterative process.

    2. Root Detection Logic:
        - The implementation immediately returns x if fabs(f_x) < tolerance, indicating a found root. However, this does not consider the situation where the method may oscillate around a root without converging, which can lead to misinterpretations about the accuracy of the result. If the initial guess is not sufficiently close to an actual root, this could return to a value that is not a valid solution.

    3. Loop Condition and Return Value:
        - After the loop, if no root is found, the function returns NaN. This is more informative than returning a guess, but it may still confuse users if they expect a numerical output (like zero) and receive NaN instead, particularly when the desired root is indeed zero.

Analysis of Failures in derivative Function

Both implementations of the derivative function exhibit similar behavior, and they may fail when the polynomial is constant (no variable term) or have very few coefficients.

1. Handling of Constant Polynomials:
   - The check for polynomial size (`if (coeffs.size() <= 1) return Polynomial({0});`) ensures that if the polynomial is constant (e.g., f(x) = c), it returns a zero polynomial, indicating that the derivative is zero. This is accurate mathematically but may not help in the context of finding roots, as the `(getRoot)` function relies on derivative values to guide the iterative process.

2. Edge Cases Not Considered:
   - For polynomials that are close to zero, both implementations do not adequately manage edge cases that could arise from floating-point precision errors. This can lead to situations where the derivative is evaluated near zero but is treated incorrectly due to these precision limitations.

3. Missing Root Values:
   - If a polynomial's root is supposed to be zero (e.g., f(0) = 0), and the `(getRoot)` function cannot accurately converge to this point due to the reasons mentioned above, the result may be misleading, as both implementations do not provide sufficient checks to return zero when appropriate.

Conclusion

The failures in the `(getRoot)` function from both GitHub Copilot and ChatGPT stem from handling edge cases, the precision of floating-point calculations, and the lack of comprehensive logic to ensure valid root detection. While both implementations are generally robust, they require additional refinement in their iterative approaches and output management to enhance accuracy and clarity, particularly in cases where roots are expected to be zero.

My Evaluation for Two Codes

Final Evaluation

In the final evaluation, ChatGPT has an accuracy of **88%**, while GitHub Copilot achieves **82%** accuracy. Copilot's code generally exhibits better time complexity and simplicity, but it sometimes produces incorrect outputs. Both tools generate clean code, but ChatGPT's comments enhance clarity and understanding. Overall, ChatGPT receives a score of **95%**, while Copilot earns a score of **90%**.

Appendix

- **Test Cases:**
  - https://drive.google.com/file/d/187gCcIWrpJlcwfbWTS5hK8O6RAYQv90I/view?usp=sharing
- **Header Code:**
  - https://drive.google.com/file/d/1-2ttuH1xu8qZqPhUVIiV1F85MqmEoQPl/view?usp=sharing
- **Main Function:**
  - https://drive.google.com/file/d/1bfJLTCqGCSW8OkfFPklZUT0JEnZPNEhR/view?usp=sharing
- **Chat GPT Code:**
  - https://drive.google.com/file/d/1C03PQ7v08pKcmA5zRx-h1I16UKHs4jrh/view?usp=sharing
- **Copilot Code:**
  - https://drive.google.com/file/d/11E-W9Ptohvm_OzD0BJBhVoBw2_unjq6b/view?usp=sharing

Cairo University

Faculty of Computers and Artificial Intelligence

CS213: Object Oriented Programming

Assignment (1)

Task (3)

Course Instructor:

Dr. Mohammad El-Ramly

National Telecommunication Institute (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.

1. Who offers it and where is it located?

The National Telecommunication Institute (**NTI**), NTI is located at 12 El-Hegaz Street, Roxy, Heliopolis, Cairo, Egypt.

2. What does it offer?

NTI offers a variety of programs, courses, and services aimed at training and developing skills in the telecommunications and information technology sectors. Some of them: Train to Hire, Upskilling Programs, International Academies, Creativa Innovation Hubs, Wazeefa-Tech, Post Graduate Diploma, Research & Development, NTI Scientific Labs.

3. When is it open and how long it is online or offline and the conditions for application?

The Ministry of Communications and Information Technology (**MCIT**), in collaboration with the National Telecommunications Institute (**NTI**) and the Information Technology Industry Development Agency (**ITIDA**), offers a Summer Training initiative for first, second, and third-year university students from engineering, computers, information, and AI faculties.

This 4-week program provides 120 training hours, including 30 hours of soft skills and 90 hours of technical training in fields like Networks, AI, Cloud Computing, Data Analysis, and IoT. The initiative is a free scholarship for outstanding students, preparing them for international certifications and enhancing their technical and professional capabilities.

The summer training program is 120 hours in two parts:

I. Soft skills / Freelance skills part 30 hours.
II. Technical part 90 hours.

NTI has 13 technical fields each one will have 90 hours in 3 weeks and some of them have an exam and you pay for it as follows:

- **Big Data Analysis.**
  I. **Personal skills:** Freelancing Skills.
  II. **Prerequisites:** Basic knowledge of programming fundamentals and computer architecture.
  III. **Exam Cost:** 200$.

- **Red Hat System Administrations.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** network fundamentals knowledge and operating systems basics.
  III. **Exam Cost:** 400$.

- **Web Designer.**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** General background in using computers, managing files, Basic knowledge of programming & internet.
  III. **Exam Cost:** no exam.

- **MEAN-Stack Web Development.**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** HTML, CSS, Mid-level proficiency in JS (Familiarity with DOM manipulation and event handling in JS).
  III. **Exam Cost:** no exam.

- **Data Analysis: Skills for Freelance Professional.**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** Fundamental mathematics and statistics, introductory programming knowledge, proficiency in excel, logical and analytical thinking.
  III. **Exam Cost:** no exam.

- **5G.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** Basic knowledge of Mobile networks and industry applications.
  III. **Exam Cost:** 200$.

- **Digital Design using FPGA.**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** Basic knowledge of programming languages.
  III. **Exam Cost:** no exam.

- **Artificial Intelligence (AI).**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** Basic knowledge of programming languages.
  IV. **Exam Cost:** 200$.

- **Cyber Ops Associate.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** Basic Computer Literacy, Basic PC operating systems navigation skills and Basic internet usage skills.
  III. **Exam Cost:** 300$.

- **Internet Of Things (IOT) Applications.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** Basic knowledge of programming & electronic fundamentals.
  III. **Exam Cost:** no exam.

- **AWS Cloud.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** Basic computer knowledge, General IT Technical knowledge, and General OT business knowledge.
  III. **Exam Cost:** 100$.

- **Cisco CCNA.**
  I. **Personal Skills:** Soft Skills.
  II. **Prerequisites:** Basic Computer Literacy, Basic PC operating systems navigation skills and Basic internet usage skills.
  III. **Exam Cost:** 300$.

- **Graphic Design.**
  I. **Personal Skills:** Freelancing Skills.
  II. **Prerequisites:** Basic Computer Literacy.
  IV. **Exam Cost:** no exam.

4. The different learning track.

NTI has many tracks like: Networks, Security and Virtualization, Embedded Systems, AI and IoT, Cloud Computing, Information Technology, Fiber Optics, Wireless and Mobile Communication Networks, Career Development.

Information Technology Institute (ITI)

Founded in 1993, the Information Technology Institute (ITI) joined MCIT as an affiliate in 2005. The development of ICT mortal capacity is its primary focus. ITI has been simplifying its training portfolio for more than 30 years by continuously observing emerging global tech trends and the nation's mega and public systems. It maintained an employment chance of 85 for its grades as a result.

Numerous young Egyptians from diverse governorates and academic backgrounds have benefited from the ITI programs, which have helped them apply their academic knowledge to improve their future employment prospects. Apart from development, additional courses cover media and vitality, media design, game development, security, data analytics, and more.

1. Who offers it and where is it located?

The **ITI** is the entity that provides specialized training and courses in information technology and related areas. It is a renowned establishment that is connected to the Egyptian Ministry of Communications and Information Technology.

2. What does it offer?

The **ITI** provides a range of programs and initiatives focused on developing advanced technical skills, improving job opportunities, and promoting innovation in the IT industry. Listed below are the main services provided:

I. 9-Month Professional Diploma:

I.1. **Target Audience:** Graduates of IT-related fields.

I.2. **Program Focus:** Intensive, industry-relevant training that covers specializations in areas such as:

I.2.1. Software Development.

I.2.2. Cybersecurity.

I.2.3. Artificial intelligence (AI) and Machine Learning (ML).

I.2.4. Data Science and Analytics.

I.2.5. Game Development.
I.2.6. Cloud computing.
I.2.7. Mobile Application Development.
I.2.8. Internet of Things (IoT).
I.2.9. Network and Infrastructure.
I.2.10. Embedded System.

II. Summer Training (Student Training Program):

II.1. **Target Audience**: Undergraduate students in IT and engineering-related fields.
II.2. **Program Focus**: Brief summer internships and hands-on technical workshops across a range of topics, providing students with early practical experience.

III. Scholarship Programs:

III.1. ITI frequently provides scholarships in conjunction with well-known international IT organizations such as IBM, Microsoft, and Oracle for additional training in state-of-the-art technologies and subsequent certification.

IV. Hackathons and Competitions:

IV.1. ITI organizes hackathons, coding contests, and innovation contests to challenge and encourage students and working professionals to find out-of-box solutions to real-life problems.

V. International Certifications:

V.1. ITI collaborates with leading multinational tech firms to provide international certifications across different fields, improving job prospects on a local and global scale.

3. When it opens and how long it is and online or offline?

The Information Technology Institute (ITI) Egypt offers multiple programs with varying schedules and delivery modes (online/offline). Here are the details for some of its major programs:

I. 9-Month Professional Diploma:

I.1. **Application Period:** Typically, applications open once a year, around June to July, with the program starting in September.
I.2. **Duration:** 9 months
I.3. **Mode of Delivery**: Traditionally, this program has been offline (in-person) at ITI's campuses, but due to COVID-19 and changing trends, they have integrated online components or adopted a hybrid model combining both online and offline learning.
I.4. **Format:** Full-time, intensive training program with hands-on projects and collaborative sessions.

II. Summer Training Program (Student Training Program):

II.1. **Application Period:** Registrations generally open around April-May, and the program takes place during the summer months (July-August).
II.2. **Duration:** Typically, 1-2 months, depending on the specific track.
II.3. **Mode of Delivery:** Online, offline, or hybrid, depending on the course and branch offering it.

III. Scholarship Programs:

III.1. **Application Period:** Varies according to the specific scholarship.
III.2. **Duration:** Depends on the course covered under the scholarship (can range from weeks to months).
III.3. **Mode of Delivery:** Can be online, offline, or hybrid based on the sponsoring organization.

IV. Hackathons and Competitions:

IV.1. **Application Period:** Varies, often announced several weeks or months in advance.

IV.2. **Duration:** Short-term events, typically lasting a few days a week.

IV.3. **Mode of Delivery:** Mostly offline, but some events may have online components or be fully virtual.

For specific start dates, you can check ITI's website or follow their social media channels for announcements on program openings and application deadlines.

4. The conditions for application and acceptance and the fees and conditions if any

I. 9-Month Professional Diploma:

I.1. **Conditions for acceptance:**

I.1.1. **Nationality:** Must be an Egyptian national.

I.1.2. **Education:** Applicants should have a university degree (bachelor's or equivalent) in IT-related fields like engineering, computer science, information systems, or communications. Some fields like science or business may also be eligible depending on specialization.

I.1.3. **Military Service:** For male applicants, they must have completed, exempted, or deferred their military service.

I.1.4. **Assessment:** Applicants must pass several assessment tests in topics like programming, English, problem-solving, and logic, and complete an interview process.

I.2. **Acceptance Process:**

I.2.1. **Online Application:** Submit the application via the ITI portal.

I.2.2. **Assessment Tests:** If shortlisted, candidates will undergo a series of written and online tests.

I.2.3. **Interviews:** Shortlisted candidates will be invited for personal interviews to assess their technical knowledge and fit for the program.

I.2.4. **Final Selection:** Candidates are selected based on their performance in tests and interviews.

I.3. **Fees:**

I.3.1. **Tuition Fee**: The 9-Month Professional Diploma is free of charge. ITI covers all training costs for accepted students, as it is funded by the Egyptian government. However, students need to commit full-time to the program.

II. Summer Training Program (for undergraduate students):

II.1. **Conditions for Application:**

II.1.1. **Nationality:** Open to Egyptian university students.

II.1.2. **Academic Background:** Applicants must be undergraduate students in IT-related fields, such as computer science, engineering, information systems, or electronics.

II.1.3. **Age and Grade Level:** Typically, students in their 2nd, 3rd, or 4th year are eligible.

II.2. **Acceptance Process:**

II.2.1. **Online Application:** Apply through the ITI platform during the registration period.

II.2.2. **Selection Criteria:** Selection is based on academic performance and sometimes a test/interview to assess technical skills.

II.3. **Fees:**

II.3.1. **Free:** The Summer Training Program is generally free of charge, like the 9-Month program, as it is government-supported.

III. International Certifications and Specialized Programs:

III.1. **Conditions for Application:**

III.1.1. It depends on the course. In most cases, candidates need to have a suitable background or previous work experience in study (such as software development, data science).

III.2. **Fees:**

III.2.1. Costs may be incurred for international certification programs, but ITI frequently collaborates with companies such as IBM, Microsoft, and Oracle to offer them at reduced prices or through scholarship opportunities.

5. The different learning tracks.

The Information Technology Institute (ITI) provides a range of training programs tailored to different IT fields and new technologies.

Every track is custom-made to fulfill market needs and provide students with specific abilities.

Below is a summary of the various educational paths available, specifically within the 9-Month Professional Diploma program and other ITI projects:

I. Software Development Track.
II. Artificial Intelligence (AI) and Machine Learning (ML) Track.
III. Data Science and Analytics Track.
IV. Cybersecurity Track.
V. Cloud and DevOps Track.
VI. Internet of Things (IoT) Track.
VII. Game Development and Digital Media Track.
VIII. Embedded Systems Track.
IX. Mobile Applications Development Track.
X. Network and Infrastructure Track.

6. Select a track that you like and will write why you want this one.

In today's technology-driven world, software development is crucial, combining creativity with problem-solving and providing various career prospects. I am eager to follow the Software Development path to acquire important skills and develop impactful solutions. Below are the main factors behind my decision

I. **Passion for Problem-Solving:**
   a. I enjoy taking on challenges and solving them with programming. In software development, I can use my logical thinking in a creative way.

II. **High Demand in the Job Market:**
   a. Career opportunities are abundant due to high demand for skilled software developers across various industries.

III. **Creative Freedom:**
   a. I can turn my ideas into reality through creating a website, app, or game, fostering innovation and creativity.

IV. **Collaborative Work Environment:**
   a. I take pleasure in team collaboration with designers and fellow developers, improving my skills through shared experiences.

V. **Flexibility in the Work Environment:**
   a. Having the choice to work from home or in different locations contributes to an improved work-life balance in my profession.

VI. **Building Scalable Systems:**
   a. I aim to develop apps that are efficient, secure, and scalable to meet user demand.

Google STEP Internship Program

1. Who offers it and where is it located?

Google STEP (Student Training in Engineering Program) is offered by Google, a global technology company. The internship is available in various Google offices around the world, including locations in the United States (Mountain View, New York, Seattle), Europe (London, Zurich, Dublin), and Asia-Pacific (Japan, Singapore, Australia). The specific locations may vary each year.

2. What does it offer?

Google STEP offers several benefits:

I. Hands-on software engineering experience working on real-world projects.
II. Mentorship from experienced Google engineers.
III. Skill development through coding and debugging in languages like Python, C++, and Java.
IV. Team collaboration and exposure to large-scale infrastructure.
V. Workshops and technical talks on cutting-edge topics.
VI. Networking opportunities with Googlers and other interns.
VII. Diversity and inclusion support.
VIII. Competitive compensation and possible perks such as free meals and transportation.

3. When it opens and how long it is and online or offline

The Google STEP program is typically open during the summer for a duration of 12 weeks. While the internship is primarily in-person at Google offices, there may be hybrid or remote options available, depending on the year and location.

4. The conditions for application and acceptance and the fees and conditions if any.

Eligibility requirements for the Google STEP internship include:

I. First- or second-year undergraduate students pursuing degrees in computer science or related fields.

II. Interest in software engineering and coding experience.
III. Priority is given to students from underrepresented groups in tech.
IV. There are no application fees, and interns are paid competitive salaries during the program.

5. The different learning tracks.

While STEP focuses broadly on software engineering, there are different types of projects that interns can work on, such as:

I. Web development.
II. Mobile app development.
III. Backend systems and databases.
IV. Machine learning and artificial intelligence.
V. Tools and infrastructure.

6. Select a track that you like and will write why you want this one.

I would choose the Machine Learning and Artificial Intelligence track. I have a strong interest in AI, and I have been building my skills in Python and C++. Working on AI projects would allow me to deepen my understanding of machine learning algorithms, and I would love the opportunity to apply my knowledge to real-world challenges at Google. It also aligns with my long-term goal of contributing to impactful AI-driven technologies.