Internship Report – Day 10

Today, I learned about how a compiler works and how it changes programming code into a language that a computer can understand. The session started with the basics of tokens and expressions, which are the small parts of a programming language. I understood that symbols like letters (a-z, A-Z), numbers (0-9), and special characters (+, -, ., etc.) are used to write code. A token is the smallest unit of a program, like a word in a sentence. These tokens come together to form sentences, which in programming are called statements. A group of statements makes a paragraph, which is similar to a block of code in a program.

I also learned about the compilation process, which is important because it translates human-written code into machine language that a computer can run. This process has different steps. The first step is Tokenization, where the program is broken into small parts called tokens. These tokens are like building blocks that help the program understand each part correctly. The next step is Syntax Analysis ,where the compiler checks if the code is written in the correct order. If the code structure is wrong, the compiler gives an error.

The third step is Semantic Analysis, where the compiler checks if the code makes sense. Even if a program is written correctly, it might still have semantic errors if it tries to do something that is not possible, like dividing a number by a word. The last step is Code Generation, where the compiler translates the correct code into machine language so that the computer can execute it.

Learning about the compiler is very useful because it helps in finding and fixing errors in a program. If I understand how a compiler works, I can write better and faster code. I also learned that there are two types of errors syntax errors, which happen when the code is not written properly, and semantic errors, which happen when the logic of the code is wrong. Knowing these differences will help me debug my code easily.

During the session, my teacher also explained the importance of compounding skills and knowledge. Learning to code is not just about writing programs—it also helps in developing logical thinking and problem-solving skills. The more I practice, the better I will get at programming.

Along with learning about compilers, I also practiced JavaScript programming by solving two problems using functions. These exercises helped me understand how to sort arrays, find duplicate values, and use JavaScript methods.

The first problem was to sort an array in ascending order. I used the sort() function in JavaScript to do this. This function takes an array and arranges the numbers from smallest to largest. I also learned that using `(a, b) => a - b` makes sure that numbers are sorted correctly, as JavaScript sometimes treats numbers like words, which can cause errors. Sorting is useful for organizing data, like arranging scores in a game or sorting a list of names in order.

The second problem was tofind duplicate values in an array. I solved this problem using the filter() method along with Set(). The filter() method helps find numbers that appear more than once, and Set() removes extra copies, keeping only unique duplicates. I learned that filter() is useful for picking certain values from an array, indexOf() can check if an element appears more than once, and Set() is a great way to remove repeated values. This method is helpful in real-world applications like cleaning up messy data, finding repeated records in a database, or identifying duplicate entries in a list.

Today’s session gave me a clear understanding of how a compiler works and why it is important in programming. It also helped me understand how a computer reads and processes code. By learning about tokens, syntax, and semantics, I now have a better idea of how errors happen and how to fix them.

Practicing JavaScript functions for sorting and finding duplicates also improved my coding skills. I now understand how sorting works, how to extract specific values from an array, and how to remove duplicates efficiently. These skills will be very useful in real-world coding, where working with data is important.

Overall, today’s session was very informative, and I feel more confident in both understanding programming concepts and writing JavaScript code. I will continue practicing and improving my skills so that I can write better and more efficient programs in the future.