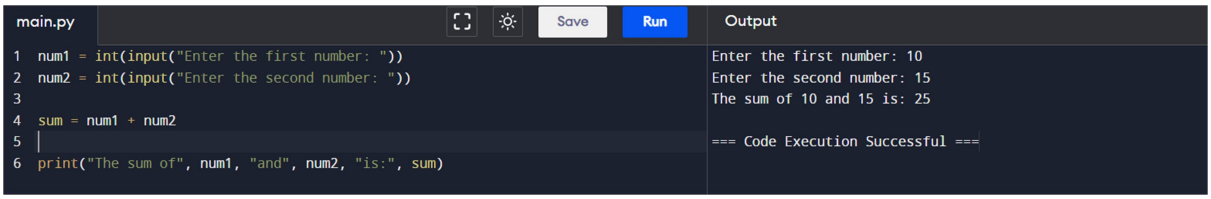
Date: - 8th April 2024 Day 1

Topic:

1. Self-Introduction.
2. Some questions regarding on Software and how it's works.
3. Write first code.
4. IF- Else condition.
5. Sequence, Selection and Iteration.
6. Variable, Constant, Declaration Initialization and assignment.
7. Scope (Global and Local)
8. Expression and Statement
9. Pseudocode.

Answer:

1. I am Deepak Kumar Jha and, I am from Darbhanga Bihar and I have dome my Graduation in Computer science engineering from Swami Vivekanand institute of engineering and technology.
2. Software refers to a set of instructions or programs that enable a computer or other hardware to perform specific tasks.

    - Software works by executing a series of instructions written in a programming language.
    - **Writing Code**: Software developers write code using programming languages like Python, Java, C++, and many others.
    - **Compilation/Interpretation**: Depending on the programming language, the code may need to be compiled or interpreted.
    - **Execution**: Once the code is compiled or interpreted, the resulting executable file or script can be run on a computer or other device.
    - **Input and Output**: Software processes input from various sources, such as user input from keyboards, mice, or touchscreens, as well as data from other software applications or external devices.
3. 

4. 

5.

- **Sequence**: Sequence refers to the order in which instructions are executed in a program. In a sequence, statements are executed one after another, in the order in which they are written. This means that each statement is executed exactly once, and the flow of control moves from one statement to the next.

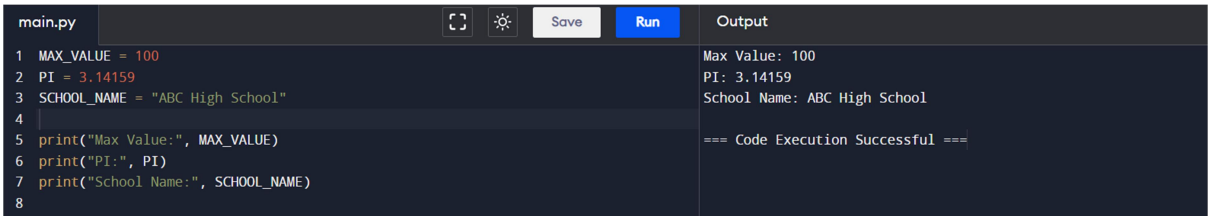
- **Selection**: Selection refers to the ability to make decisions in a program based on certain conditions. This is typically done using conditional statements such as if statements. Based on the evaluation of a condition, the program can choose to execute different blocks of code.

- **Iteration**: Iteration, also known as looping, refers to the repetition of a block of code multiple times. This allows the program to perform the same or similar actions repeatedly without the need to duplicate code. There are different types of loops in programming, such as for loops, while loops, and do-while loops, each serving different purposes and controlling the flow of execution differently.

6.

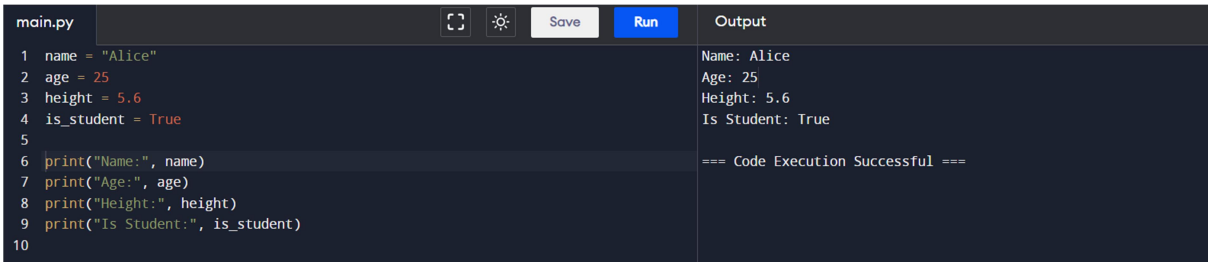
- **Variable**: A variable is a named storage location in a computer's memory that holds a value. Variables can hold various types of data, such as numbers, text, or complex data structures. They provide a way to refer to and manipulate data within a program.

- **Constant**: A constant is similar to a variable, but its value cannot be changed once it is assigned. Constants are used to represent fixed values or quantities that do not change during the execution of a program.



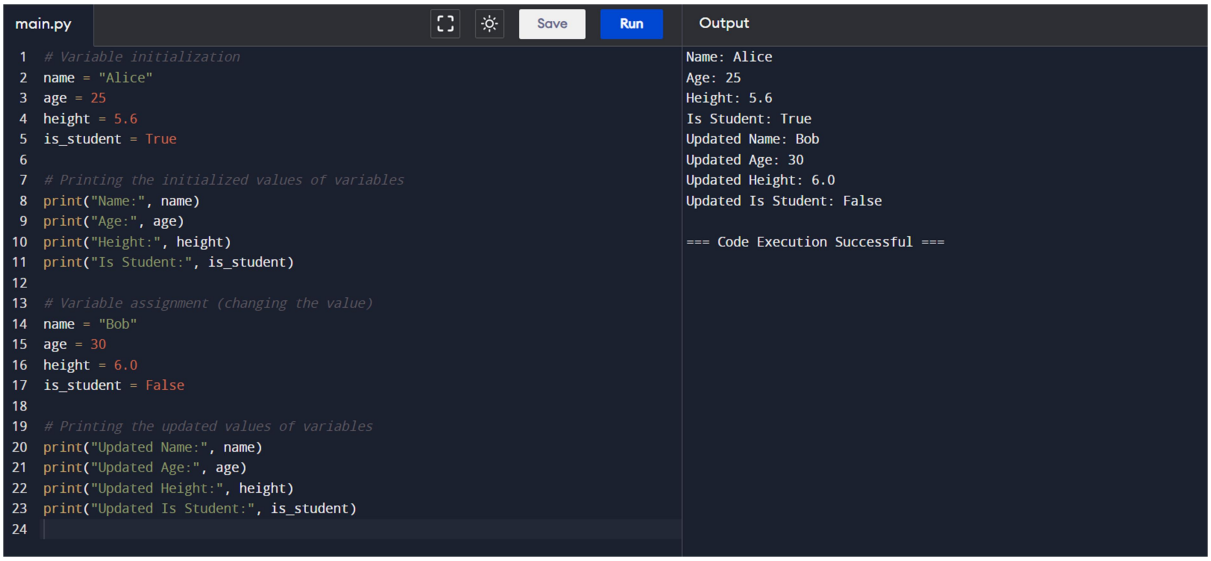
- **Declaration**: Declaration refers to the process of introducing a new variable or constant to a program. When a variable or constant is declared, the programming

language reserves memory space for it and associates a name with that memory location.



- **Initialization**: Initialization is the process of assigning an initial value to a variable or constant when it is declared. Initializing a variable or constant ensures that it has a valid starting value before it is used in the program.

- **Assignment**: Assignment is the process of giving a value to a variable or constant after it has been declared and initialized (if applicable). Assignments are typically done using the assignment operator (=), which assigns the value on the right-hand side of the operator to the variable or constant on the left-hand side.



7.

Global Scope:

- ✓ Variables defined outside of any function or class has global scope.
- ✓ They can be accessed and modified from anywhere within the program.
- ✓ Global variables persist throughout the program's execution.
- ✓ Global variables are declared using the global keyword if they need to be modified inside a function.

Local Scope:

- ✓ Variables defined within a function have local scope.
- ✓ They can only be accessed and modified within the function where they are defined.
- ✓ Local variables are created when the function is called and destroyed when the function exits.
- ✓ Local variables do not affect variables with the same name in the global scope.

8. Kjd

Expression:

- ✓ An expression is a combination of values, variables, operators, and function calls that evaluates to a single value.
- ✓ Expressions can be simple, such as a single constant or variable, or complex, involving multiple operations and function calls.
- ✓ Examples of expressions include arithmetic expressions (3 + 4), logical expressions (x > 5), function calls (math.sqrt(9)), and more.
- ✓ Expressions can be used wherever a value is expected, such as in assignments, function arguments, or conditionals.

Statement:

- ✓ A statement is a complete instruction that performs a specific action in a program.
- ✓ Statements can include expressions but also consist of keywords, operators, and other elements that control the flow of the program.
- ✓ Examples of statements include assignment statements (x = 5), conditional statements (if-else), loop statements (for, while), function definitions, import statements, and more.
- ✓ Unlike expressions, statements do not necessarily evaluate to a value but instead perform some action or control the flow of execution in the program.

main.py Save Run Output

```
result = 3 + 4

if result == 7:
    print("The result is 7")

```

```
The result is 7

=== Code Execution Successful ===
```

9. Pseudocode is a way of representing an algorithm using a mixture of natural language and informal programming syntax. It's not tied to any specific programming language and serves as a bridge between a problem statement and the implementation in code. Pseudocode is used to outline the logic and steps of an algorithm in a clear and understandable manner before translating it into actual code.

main.py

```
Start
Input num1, num2
If num1 > num2 Then
    max = num1
Else
    max = num2
End If
Output max
End
```