***Doubts solved by Anjali Mam***

**Q. Difference between int main, void main, main?**

**Ans.** Int main will retun 0 value (analogy-Bill of any purchase is simply a return value. Void main will not return 0 value. (analogy-Shopkeeper will not give you receipt of Rs.10 soap). And if you simply write main, automatically it will take int main. But in Dev-C++, you don't need to write int main, you can simply write main().

**Q. Full-stack developer? Which one-JAVA\_Python\_Node.js**

**Ans.** Node.js

**Q. Why '$' sign is accepted when defining variables in Dev C++?**

* **Ans.** First let us list the rules to define a variable
  + Variable should start not start with digits.
  + Variable should not contain whitespaces.
  + Variable should not contain keywords from respective language.
  + Variables are case sensitive.

***NOTE: $ sign accepted as variables is just a compiler error in Dev C++.***

Variable declarations are a fundamental aspect of programming languages, allowing you to define and allocate memory for variables that store data values. The specific rules for variable declarations can vary depending on the programming language you're using, but here are some general guidelines that apply to many programming languages:

1. **Syntax**: Variable declarations typically follow a specific syntax. They usually start with the data type, followed by the variable name, and optionally an initial value.
2. **Data Types**: Different programming languages support various data types, such as integers, floating-point numbers, characters, strings, booleans, and more complex types like arrays and structures. When declaring a variable, you need to specify its data type.
3. **Variable Names**:
   * Variable names should be meaningful and descriptive.
   * They usually start with a letter (uppercase or lowercase) or an underscore (\_).
   * They can contain letters, digits, and underscores.
   * Variable names are case-sensitive, meaning **myVariable** and **myvariable** could be treated as distinct variables.
4. **Initialization**: Variables can be initialized with an initial value at the time of declaration. This value should be compatible with the variable's data type.
5. **Scope**: Variables have a scope, which defines where they can be accessed. Some variables are local to a specific block of code (e.g., inside a function), while others might have a global scope, accessible throughout the program.
6. **Modifiers**:
   * Some languages allow the use of modifiers like **const** or **final** to indicate that the variable's value should not be changed after initialization.
   * In object-oriented languages, you might encounter access modifiers like **public**, **private**, or **protected**, which determine where the variable can be accessed from.
7. **Rules and Conventions**:
   * Some languages have reserved keywords that cannot be used as variable names.
   * Follow any naming conventions established by the programming language community. For example, in Java, variable names often start with a lowercase letter and use camelCase (e.g., **myVariableName**).
8. **Declaration Statements**: Variable declarations are often statements in themselves, but in some languages, they might need to be included within a specific declaration section at the beginning of a code block.

Here are a few examples of variable declarations in different programming languages:

**C++:**

cppCopy code

int age = 25; // Declaration with initialization double temperature; // Declaration without initialization

**Python:**

pythonCopy code

name = "Alice" # Declaration with initialization count = 0 # Declaration with initialization

**Java:**

javaCopy code

int score = 100; // Declaration with initialization char grade = 'A'; // Declaration with initialization

Remember that these are general guidelines, and the exact syntax and rules might differ based on the programming language you're working with. Always consult the documentation or resources specific to the language you're using for accurate information.

**Q. What is TypeCasting?**

**Ans.** Conversion of one datatype into another is called “***TypeCasting***”.

E.g. Int 🡪 float, float 🡪 int, char 🡪 int?