Declarations:

Declarations provide information about the name and type of data objects needed during program execution.

Two ways of declaration

- implicit declaration
- explicit declaration

Implicit Declaration or default declaration:-

They are those declaration which is done by Compiler when no explicit declaration or user-defined declaration is mentioned.

Example:

```
$abc = 'a string';
$abc = 7;
```

In 'Perl' Compiler implicitly understand that $abc = 'a string' is a string variable and $abc = 7; is an integer variable.
Explicit declaration of data object:

float A, B;

It is an example of 'float A, B', of C language. In explicit we or users explicitly defined the variable type. In this example it specifies that it is of float type of variable which has name A & B.

A declaration basically serves to indicate the desired lifetime of data objects.

Declarations of Operations:

→ Compiles needs the signature or a prototype of a subprogram or function so it can determine the type of argument is being used and what will be the return type

* Before the calling of Subprogram, Translator need to know all these informations*

Eg in C language

float Sub (int x, float y)
It declares `Sub` to have the signature

\[
\text{Sub: int x float } \rightarrow \text{ float}
\]

Purpose of Declarations:

1. **Choice of Storage Representation:**

   As translators determine the best storage representation of data types that why it needs to know primarily the information of data type and attribute of a data object.

2. **Storage Management:**

   It makes to us to use best storage management for data object by providing its information & these information as tells the life time of a data object.

   **For Example:**

   In C Language we have many options for declaration for elementary datatype

   \( (i) \) Simple declaration: like `float A,B;`

   It tells lifetime is only at the end of execution.

   *As lifetime of every data object can be maximum to
end of execution time.
but Simple declaration tells the single block of memory
will be allocated.

(ii) Runtime Declaration:

C language and many more languages provide the feature of Dynamic Memory Allocation by keywords malloc and calloc.
So in this, special blocks of memory is allocated in memory and their life time is also different.

3) Polymorphic Operations:

In most languages, some special symbol like + to designate any one of the several different operation which depends on the type of data or argument is provided.

In this operation, has some name like as we discussed +. So in this case, operation symbol is said to be overloaded because it does not designate one specific operation.

Ada allows programmer to overload subprograms.
ML expands this concept with full polymorphism whose function has one name but variety of implementation depending on the types of arguments.

4) Type checking:

Declaration is basically for Static type checking rather than dynamic.

Computer Science Lectures By ER. Deepak Garg