High Level Programming Language

How to make decission?
How to build new language?

Why do we need?

Ease of interaction between:

Programmer ⟷ Computer system

New Programming Language

- How to make decission?
- How to build new language?

There could be two perspecives:

- Programmer perspective
- Hardware perspective

Simplicity: easy to learn, code
Easy to debug: precise error message
Portability (Library reuse...)
Compatibility: Legacy

Performance
Power consumption

Making Decission for NewLang

Possibilities:

Function
Method
Class
Constructor
etc.

Operators

How to represent

Operation/Instruction

Derived type

Real number (floating point)
Integer (range of integer)
Char
Bit

Size of data
Different type of data

Where and How to store?

Operand/data

Operand/Data

Where and How to store?

Data Memory

and

Register

How to specify size?

Data type:

char – for character
int - for integer type
float – real number
bit – binary type

Combine these data types:

Many characters: string type or character array, *char arra[10] ;*
Many integers: integer array, *int array[10] ;*

Combine two different types:

Structure, Union, etc.

Allocating Memory: Operand/Data

Where and when to store?

A program goes through multiple phases:

- Write the code
- Compilation
  - Generate token (LEX)
  - Parse them for correct syntax
  - VM/Asm Code generation
  - Assembly to machine
- Execution (line by line,instruction by instruction)

Data Memory

and

Register

Two important informations: How much space and where?

- During compilation itself (static, iformations are available early)
- During program execution (dynamic, informations are available during execution)

Allocating Memory: Operand/Data

Where and **when** to store?

Example: Keyword to tell how much memory: *int, char, float, bool, etc*

Generally these can be done during compilation itself

Char: n bits - 1 byte
Int: m bits - 2 byte, 4 byte, 8 byte, (why not 3 byte or 5 byte?)
Float: l bits - 2 byte, 4 byte, 8 byte etc
Bool: 1 bit - generally just one bit (until you have different kind of logic)

Structure: addition of all the members
Union: Max of (all the members)

Similarly for derived type as well.

Dynamic allocation: Program has to take help of some other programs (OS) to get memory

Example: malloc, calloc, new, constructor, etc

Allocating Memory: Operand/Data

Where and when to store?

Where to store, such that only desired instructions can get to access?

Possibilities:

- Every instructions/function can access
- Only some of them can access
- Only one function can access

- Some can access for sometime
- All can’t access all time
- A function can access only for and during allotted time

Example: scope of the variables

- global or static
- local
- parameters
- private
- public

When it comes to language, you have to decide on place of declaration and definition of these variables.

Allocating Memory

The concept of allocating memory cab be similar for

primary data type

as well as

derived data type

In memory there are two main places:

stack of each function/subroutine

Or

Heap

The registers can also be used!

Making Decission for NewLang

Possibilities:

Function
Method
Class
Constructor
etc.

Operators

How to represent

Operation/Instruction

Derived type

Real number (floating point)
Integer (range of integer)
Char
Bit

Size of data
Different type of data

Where and How to store?

Operand/data

Instructions/Operators

Use of special symbols:

+, -, *, /, ?:, ^, &, |, ~, [ ], ( ), { },

Depending on language requirement these symbols can be used.

Using operators and variables together will lead you to an **Expression**

Next, how to deal with these expressions?

Operator precedence (which one first? Need to be consistent).

Example:

Var = a + b – c * d / -c + a * b

You need to follow mathematical laws. Similarly other laws.

Generally, most programming langs use ( ) to clarify the precedence.

Expression and Function

Many expression combine together can form a function.

So, entire program can have just one single function!

Important thing is: to somehow indicate from where the program should begin.

Example: C has a function:

```
main(  ) {

    Expression.....

}
```

Here main has to be keyword, not just identifier, there can’t be multiple first time entry to the program (at least for C language).

Try building a languge having multiple entry!

Expression and Function

Questions:

- Is it efficient to have just one function?
- What would be the structure of function? How does it look like?
- How doest a function deal with the variables?
-
- Any other important questions?

- Modularity
- Portability
- Code size
- Debugging
- Security
- Other new properties

Function name
Return type
Parameters

Scope

Multiple Functions

Questions:

- Is it efficient to have just one function?
- What would be the structure of function? How does it look like?
- How doest a function deal with the variables?
-
- Any other important questions?

How many? What would be the size, nature etc?

Example: C, by principle it should support infinite number of functions. There is no way you can restrict. Infinite recursion etc.....

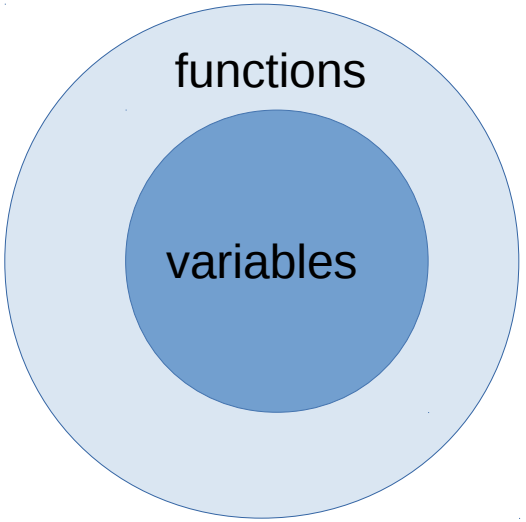
There is *main* fuction and there could be infinite secondary functions!

In C there is no concept of small and big function. Does it really matter?

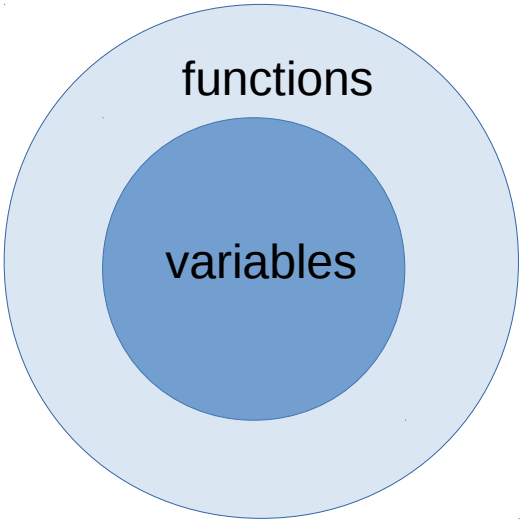
Multiple Functions and Variables

Questions:

- Is it efficient to have just one function?
- What would be the structure of function? How does it look like?
- How doest a function deal with the variables?
-
- Any other important questions?



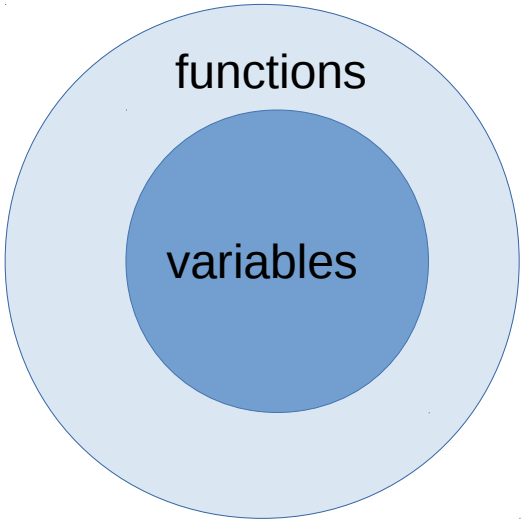
.....



Object Oriented

Idea is to have good relationship between Function and variables

- Class
- Object
- Constructor
- Destructor
- Inheritance
- New concepts on this



The Hack Languges

Java like object oriented
languge

Goal: to design new language

Please design one before you graduate.

Conclusion

Final test: 2nd December, 2019

Syllabus:

- Architecture
- Assmbler
- VM
- Progrmming concept