

Keywords

Low Level Language, High Level Language, 1st Generation, 2nd Generation, 3rd Generation, Machine Code, Language Translator, Assemble, Compiler, Interpreter, Device Driver, Debug

Translators and Facilities of Languages

TASK - Define the keywords.



Generations of Programming

Objectives

BEGINNER:
Describe the different generations of programming language.

ADVANCED:
Describe the differences between Low Level and High Level Languages.

EXPERT:
Evaluate the benefits of programming in both Low and High Level languages and state which translator is needed for each and why

1st Generation

Machine Code
Directly Executable by the processor
The Generation that “computers understand”
Difficult to program in, hard to understand, hard to find errors (hard to debug)

2nd Generation

Assembly Code that uses mnemonics.
Easier to program in than 1st Generation but still difficult. One Assembly Language instruction translates to one Machine Code Instruction. Needs to be translated into Machine Code for the computer to be able to execute it. Uses an Assembler. It is most commonly used to program Device Drivers.

3rd Generation

Easier to understand (programmer)
Easier to find errors, easier to de-bug
Uses English-Like Keywords
One instruction translates into many machine code instructions
E.g.'s Java, Basic, Pascal, C+
Translated using Compiler and Interpreter

4th Generation

Known as a Declarative Language
Facts and Rules are stated
Describes what computation should be performed and not how to perform it
Examples include:
SQL
Expert Systems
Artificial Intelligence





Identify the Generation

Objectives

BEGINNER:
Describe the different generations of programming language.

ADVANCED:
Describe the differences between Low Level and High Level Languages.

EXPERT:
Evaluate the benefits of programming in both Low and High Level languages and state which translator is needed for each and why

Code

```
Dim Num1, Num2, Tot as Integer  
Num1 = Console.ReadLine()  
Num2 = Console.ReadLine()  
Tot = Num1 + Num2  
Console.WriteLine("Total is: " & Tot)
```

```
01010101010100101010100101010  
10101001111100100010000101010  
0101
```

```
LOAD r1, c  
LOAD r2, d  
ADD r1, r2  
DIV r1, #2
```

Generation

High Level (3rd Generation)

Low Level (2nd Generation)

Low Level (1st Generation – Machine Code)