Unit Assessment

11.1A - Computer Systems

Instructions for the student

1. You have **20 minutes** to answer, according to summative assessment specification (do not talk, do not copy, and do not ask for help from a teacher or other students).

2. If you need more space for answers, use the back of the sheets in the booklet, also indicating the respective question number.

For any question that begins with the “describe” or “explain the answer” should provide a complete, logical , fully explained answer.


PHP
HTML5
JAVA
HTML5
.com
http:
Swift
JS
http:
Python
C#
Arduino
Ruby
CSS3
Swift

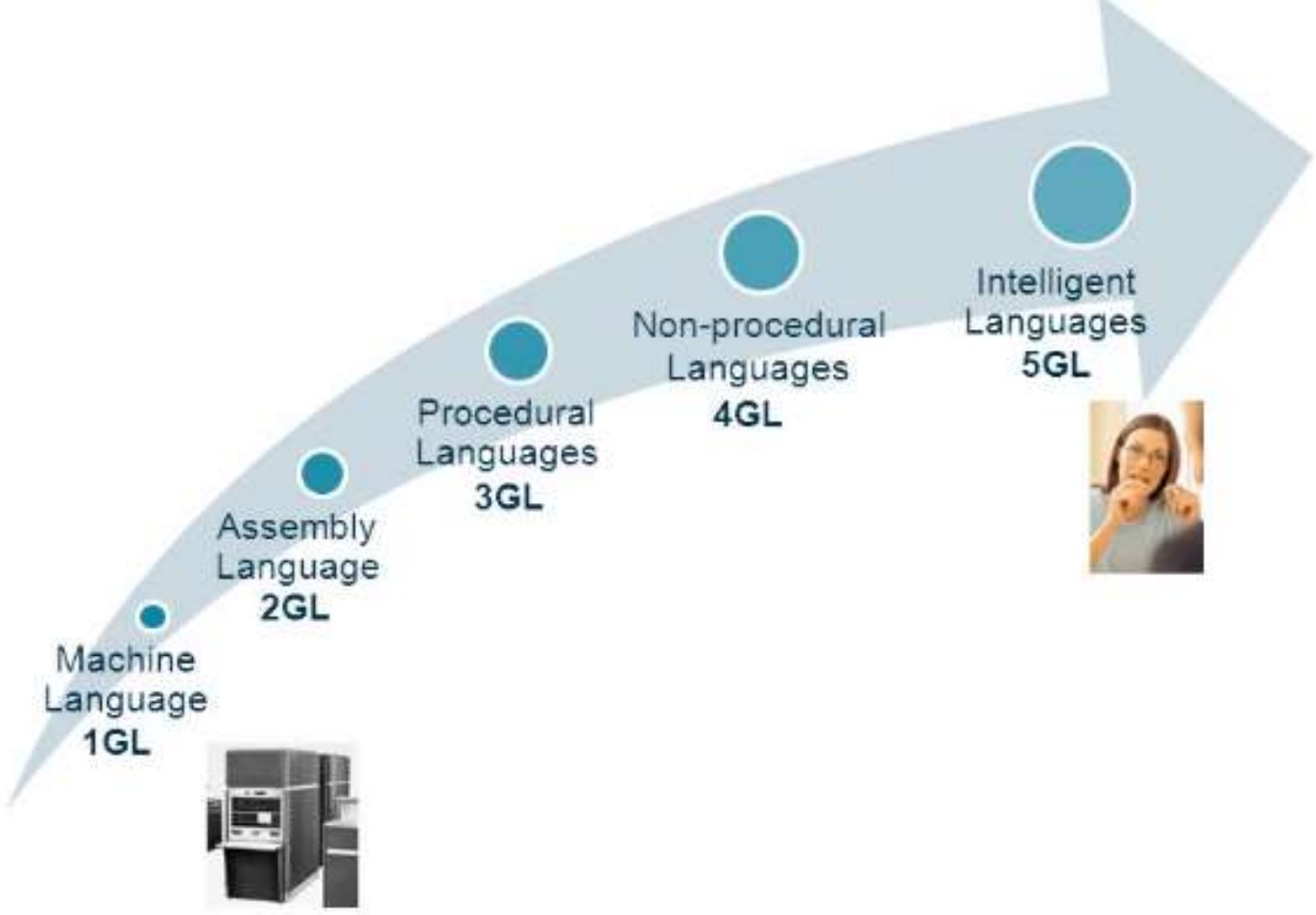
Classification of Programming Languages. Generations of Programming Languages.

Learning objectives

- ➔ distinguish between generations of programming languages
- ➔ classify programming languages into low-level and high level
- ➔ advantages and disadvantages of high-level languages
- ➔ advantages and disadvantages of low-level languages

Assessment criteria

- classify different generations of PL.
- divides programming languages into high and low levels.
- define advantages and disadvantages of low level PL
- define advantages and disadvantages of high level PL


Machine Language
1GL
Assembly Language
2GL
Procedural Languages
3GL
Non-procedural Languages
4GL
Intelligent Languages
5GL

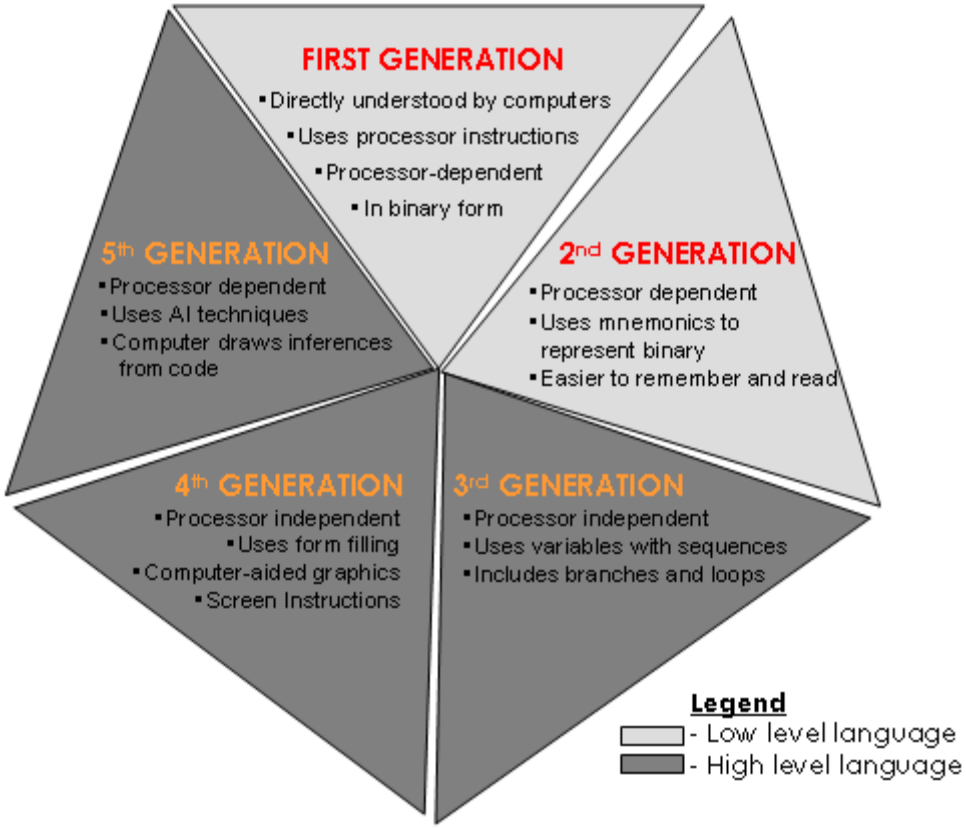
Activity 1. Group work

1st group: First generation
2nd group: Second generation
3rd group: Third generation
4th group: Fourth generation
5th group: Fifth generation

LINK

Make a presentation
You should say:

- About generation
- Advantages and disadvantages
- Give example (source code and name of program language)


FIRST GENERATION
▪ Directly understood by computers
▪ Uses processor instructions
▪ Processor-dependent
▪ In binary form
2nd GENERATION
▪ Processor dependent
▪ Uses mnemonics to represent binary
▪ Easier to remember and read
3rd GENERATION
▪ Processor independent
▪ Uses variables with sequences
▪ Includes branches and loops
4th GENERATION
▪ Processor independent
▪ Uses form filling
▪ Computer-aided graphics
▪ Screen Instructions
5th GENERATION
▪ Processor dependent
▪ Uses AI techniques
▪ Computer draws inferences from code
Legend
- Low level language
- High level language

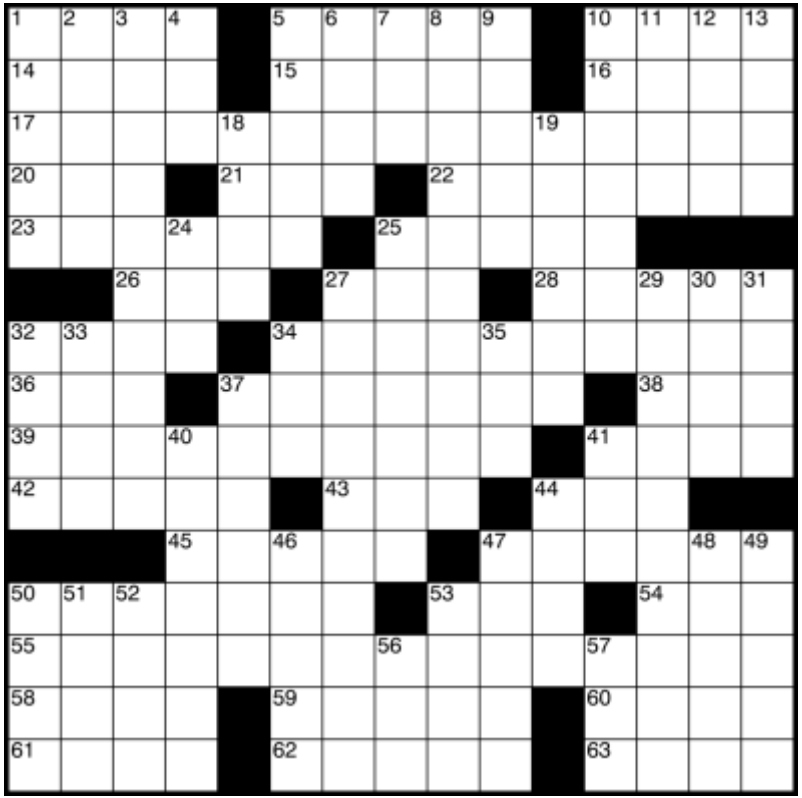
Let’s watch video

https://youtu.be/1OukpDfsuXE?si=BaeyaW2dpqS_uo3Q

Activity 2. Half a Crossword

descriptors:

- ➔ creates question related with term;
- ➔ defines the term by the question;
- ➔ follow by the right order of creating questions.



What is a computer according low and high levels of programming languages

Programming Languages Levels: Machine Language

- Only language understood by computers
  - Does not resemble human language
  - Hardware specific
- Difficult to read and write
- Sequence of bits

Programming Languages Levels: Machine Language Example

55	89	e5	53	83	ec	04	83	e4	f0	e8	31	00	00	00	89
c3	e8	2a	00	00	00	39	c3	74	10	8d	b6	00	00	00	00
39	c3	7e	13	29	c3	39	c3	75	f6	89	1c	24	e8	6e	00
00	00	8b	5d	fc	c9	c3	29	d8	eb	eb	90				

Machine code to calculate the greatest common divisor of two integers for the x86 (Pentium) instruction set.

Programming Languages Levels: Assembly Language

- Hardware specific
- Mnemonic symbols
  - Represent machine code instructions
- Example: Add two numbers
  - ADD R1, R2, R3
- Requires translator
  - Assembler
  - Maps mnemonics to machine instructions and data

Programming Languages Levels: Assembly Language Example


```
        pushl   %ebp                    jle     D
        movl    %esp, %ebp              subl    %eax, %ebx
        pushl   %ebx            B:      cmpl    %eax, %ebx
        subl    $4, %esp                jne     A
        andl    $-16, %esp      C:      movl    %ebx, (%esp)
        call    getint                  call    putint
        movl    %eax, %ebx              movl    -4(%ebp), %ebx
        call    getint                  leave
        cmpl    %eax, %ebx              ret
        je      C               D:      subl    %ebx, %eax
A:      cmpl    %eax, %ebx              jmp     B
```

Assembly language code to calculate the greatest common divisor of two integers for the x86 (Pentium) instruction set.

Programming Languages Levels: High Level

- Computer architecture/hardware independent
  - Portable
- Resembles natural/human languages
  - Easy to read, write, maintain
- Require translator
  - Compiler
  - Maps statements to machine instructions and data

Programming Languages Levels: High Level Language Example

```
int  gcd (int a, int b)  {
     while (a  !=  b) {
               if (a > b) a = a – b;
               else b = b – a;
      }
      return a;
}
```

High level language (C language) code to calculate the greatest common divisor of two integers.

Programming Languages Levels: Abstraction

What is abstraction?

- Focus on high level rather than implementation details.

What does it mean to high level languages?

- Program statements do not deal with specific computer architecture.
- Programming efficiency.

What sort of statements are used in high level languages?

- Variables
- Arrays
- Expressions
- Procedures, subroutines, functions

Activity 4. Quizlet Live

1. Go to www.quizlet.live
2. Type six digit code to join game
3. Type your name(First name+Last name)
4. Wait until teacher start game

LINK

Descriptors:

- divides languages into high and low;
- define advantages low level languages;
- define advantages high level languages;
- define disadvantages low level languages;
- define disadvantages high level languages