SPPP 2102
BASIC PROGRAMMING

INTRODUCTION TO PROGRAMMING

SEKOLAH PENDIDIKAN,
FAKULTI SAINS SOSIAL DAN KEMUANUSIAAN, UTM

2020

Week 1 - Lecture Outline

This lecture focuses on

1. Computer Systems
2. Computer Programming
3. Computer Languages

Week 1 - Lecture Outline

At the end of this lesson, students should be able to:

- Explain about Computer Systems
- Elaborate on Computer Programming
- Compare the Programming Language Generation

Computer Systems

SEKOLAH PENDIDIKAN,
FAKULTI SAINS SOSIAL DAN KEMUANUSIAAN, UTM

2020

Opening Questions

Take a moment to reflect on your experience with a computer and its system.

Come up with a positive and a negative example.

(eg : what the positive effect to your daily life)

(eg : what the negative impact to copyright issues)

Computer Systems

- A computer is a system made of two major components: **hardware** and **software**
- Computer hardware: physical equipments
- Computer software: the collection of programs that allow hardware to do its job (eg : display text, video on monitor, print out the MS Word documents)

Computer Hardware

- The computer hardware consists of five parts:
  - Input devices (mouse / keyboard)
  - Central processing unit (CPU)
  - Primary storage of main memory (RAM)
  - Output devices (Monitor / speaker)
  - Auxiliary storage devices or secondary storage (Pen-drive)

Now, name one example for each parts of the components.

Computer Software

- Computer software is fall into two broad categories:
  - System software
  - Application software

- System software manages the computer resources; i.e. OS & system utilities (Windows XP / AVG anti-virus)
- Application software is responsible for helping users solve their problems; i.e. Microsoft Office (MS Word)

Computer Software

Software

- Program(s) + Data (input)

Program

- Set of instruction in programming language.
- Deal with computer / hardware to solve the problem / make a calculation

Data

- Refer to input / source that will be process by computer

Computer Software

- Algorithm
  - A set of procedure or step by step process to solve the problem

Computer Programming

SEKOLAH PENDIDIKAN,
FAKULTI SAINS SOSIAL DAN KEMUANUSIAAN, UTM

2020

Computer Programming

- Why do you need to know about programming?

- Programming is used to create the application / software you use everyday
(eg ; to calculate your BMI)

- Application software is the result of the efforts of computer programmers.

- Knowing at least the basics of the history & practices of the programming will help you to **better understand what goes on inside computer**.

Computer Programming

Important keywords:

- Computer programming / programming
  - □ is a multi step process for designing or creating instructions or solution.
- Programming language
  - □ is a set of words (or symbols) & rules used to create instructions for computer to perform.
- Program / Application
  - □ is a list of instructions that the computer must follow in order to perform specific assigned task.

Computer Programming

Important keywords:

- Syntax - set of rules to create program
- Code - computer instructions

Computer Programming

Example :

- Programming language

  Eg : C, *C++, HTML, PHP, Java, Basic, Fotran*
- Program / Application

  Eg : *MS Word, S.M.M, S.M.A, Attendance Record*.
- Syntax –

```
     cout<<"\n\t Skor purata = ";
                cout<<purata;
                cout<<"\n\t Enter for release";
      cout<<endl;
```

- Code – <html><head><body></body></head></html>

Computer Languages

SEKOLAH PENDIDIKAN,
FAKULTI SAINS SOSIAL DAN KEMUANUSIAAN, UTM

2020

Computer Languages

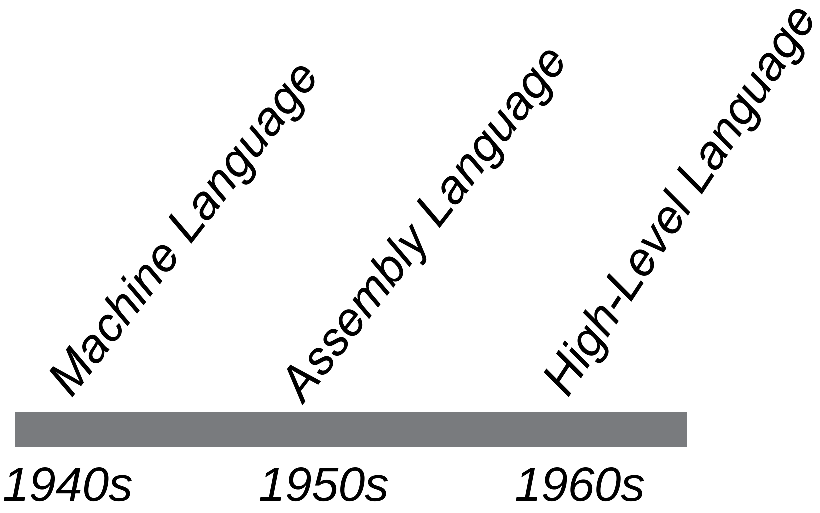
- To write a program for a computer, you must use a computer language.

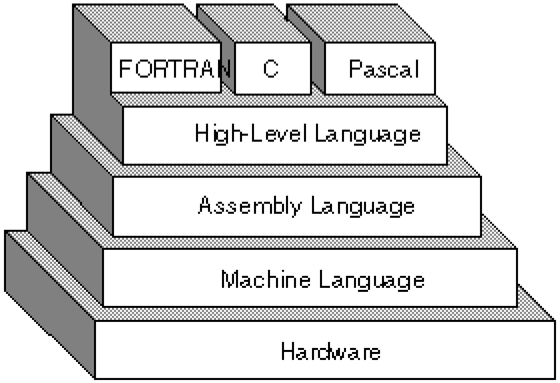
Basically, What is a computer language?

- A computer language is an artificial language that tell the computer what to do.
- It has the same meaning with programming language .

Computer Languages

- Over the year, computer languages have evolved.





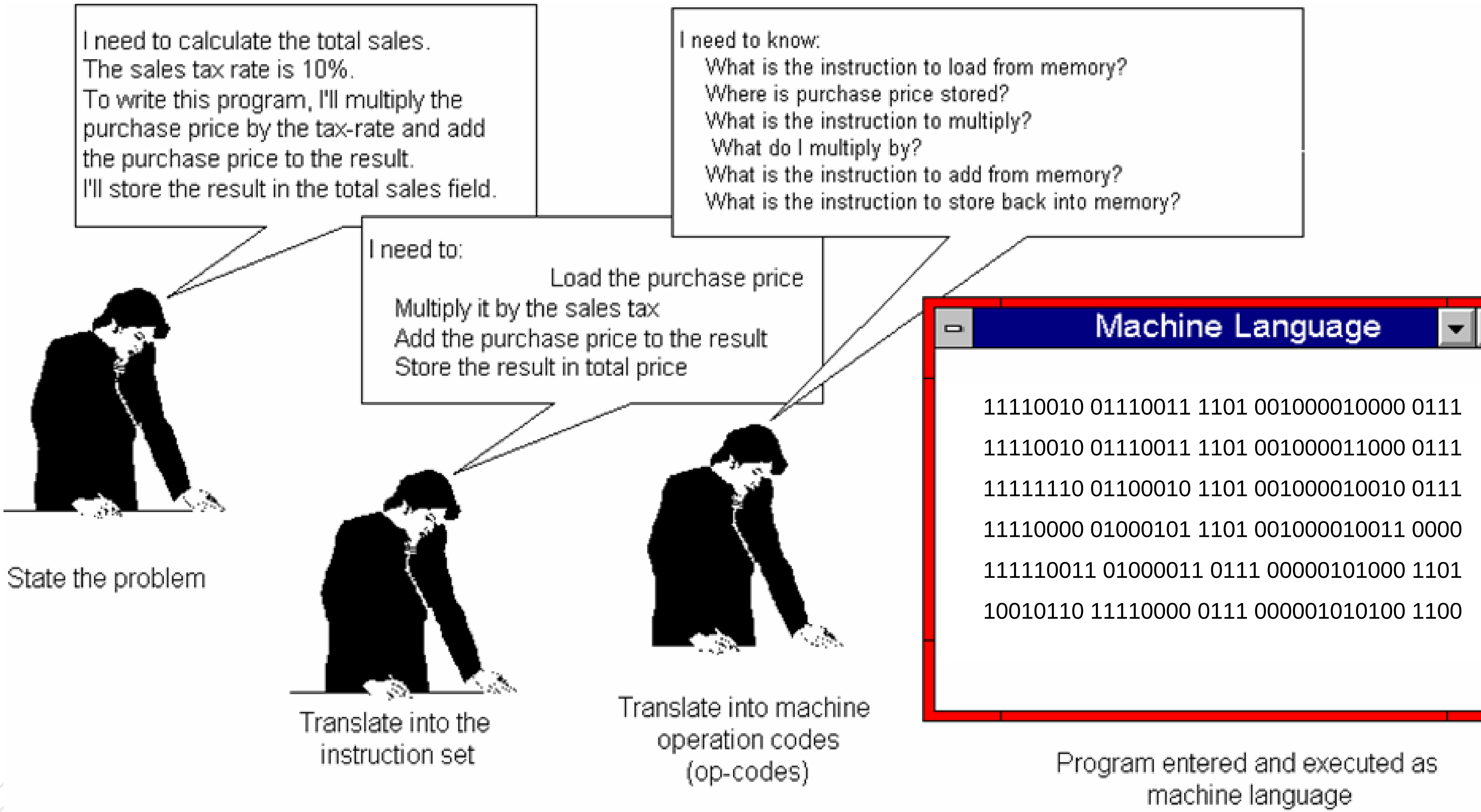
Machine Language

- 1st generation of programming language.
- The only language understood by a computer without translation.
- It is a language consists of 0s and 1s – that directly correspond to the computer's electrical states.
- Also known as binary or machine code.

Machine Language

- **Advantage:**
  - Very fast in processing data. **WHY?**
- **Disadvantages:**
  - Machine (or hardware) dependent
  - Time-consuming
  - Difficult to create program (less user friendly) (100110101011001)

Example: Machine Language Program



I need to calculate the total sales.
The sales tax rate is 10%.
To write this program, I'll multiply the purchase price by the tax-rate and add the purchase price to the result.
I'll store the result in the total sales field.

State the problem

I need to:
Load the purchase price
Multiply it by the sales tax
Add the purchase price to the result
Store the result in total price

Translate into the instruction set

I need to know:
What is the instruction to load from memory?
Where is purchase price stored?
What is the instruction to multiply?
What do I multiply by?
What is the instruction to add from memory?
What is the instruction to store back into memory?

Translate into machine operation codes (op-codes)

```
11110010 01110011 1101 001000010000 0111
11110010 01110011 1101 001000011000 0111
11111110 01100010 1101 001000010010 0111
11110000 01000101 1101 001000010011 0000
111110011 01000011 0111 00000101000 1101
10010110 11110000 0111 000001010100 1100
```

Program entered and executed as machine language

Programming in Machine Language - the programmer does all the translation

Assembly Language

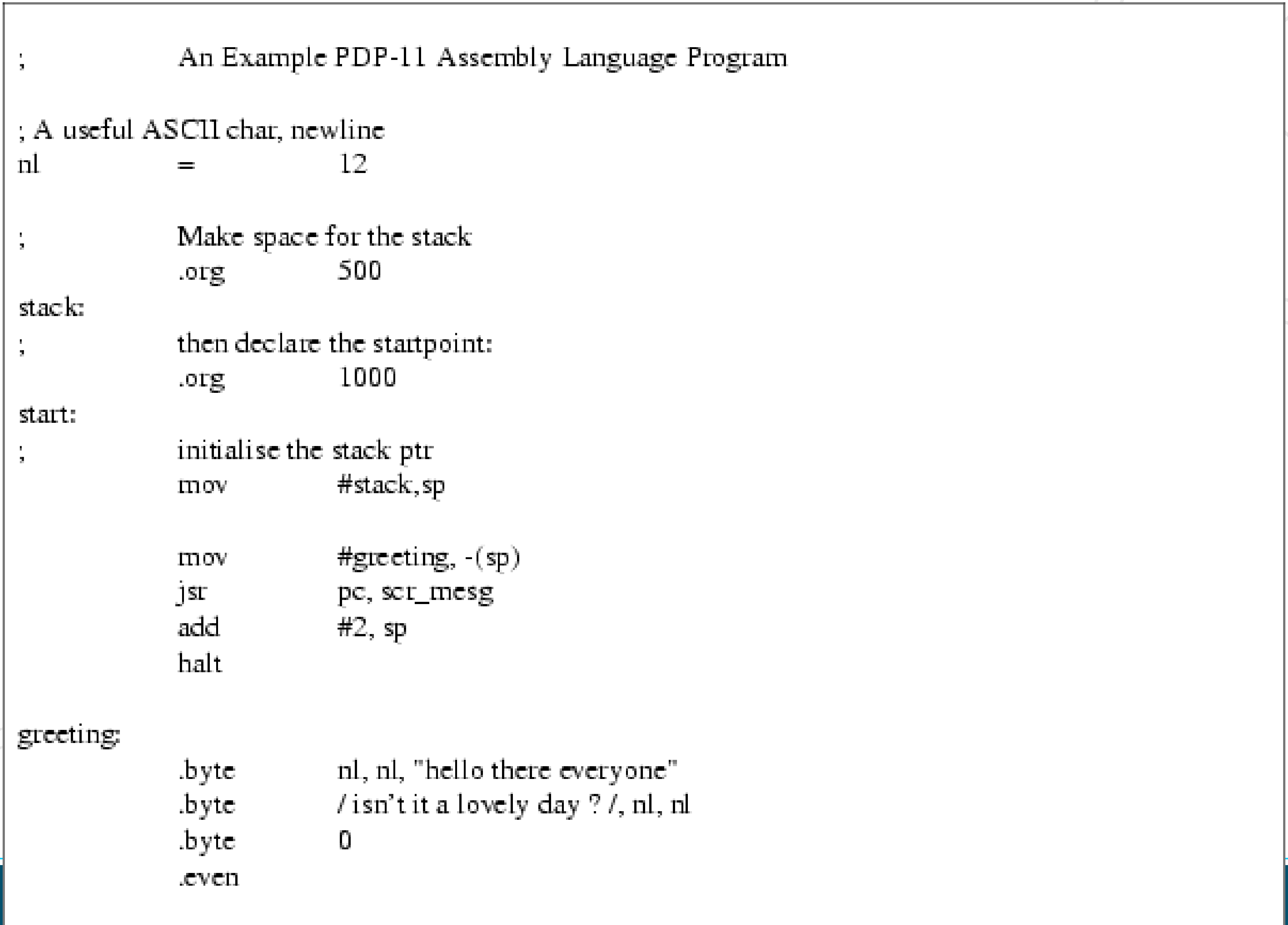
- 2nd generation of programming language.
- Also known as symbolic language.
- Assembly language is a language that allows programmers to use symbol or mnemonics (abbreviations), to represent the various machine language.
- It uses assembler to translate assembly code into machine code.

Assembly Language

- **Advantage:**
  - fast in processing data
  - Program can be write more quickly than in machine language

- **Disadvantages:**
  - Machine (or hardware) dependent
  - Time-consuming

Example: Assembly Language



```
;               An Example PDP-11 Assembly Language Program

; A useful ASCII char, newline
nl              =               12

;               Make space for the stack
                .org            500
stack:
;               then declare the startpoint:
                .org            1000
start:
;               initialise the stack ptr
                mov             #stack,sp

                mov             #greeting, -(sp)
                jsr             pc, scr_mesg
                add             #2, sp
                halt

greeting:
                .byte           nl, nl, "hello there everyone"
                .byte           / isn't it a lovely day ? /, nl, nl
                .byte           0
                .even
```

High-Level Language

- 3rd generation of programming language.
- Also known as procedural language.
- High-level languages use an English-like language instead of symbols and abbreviations.
- High-level languages are designed to relieve the programmer from the details of the assembly language.

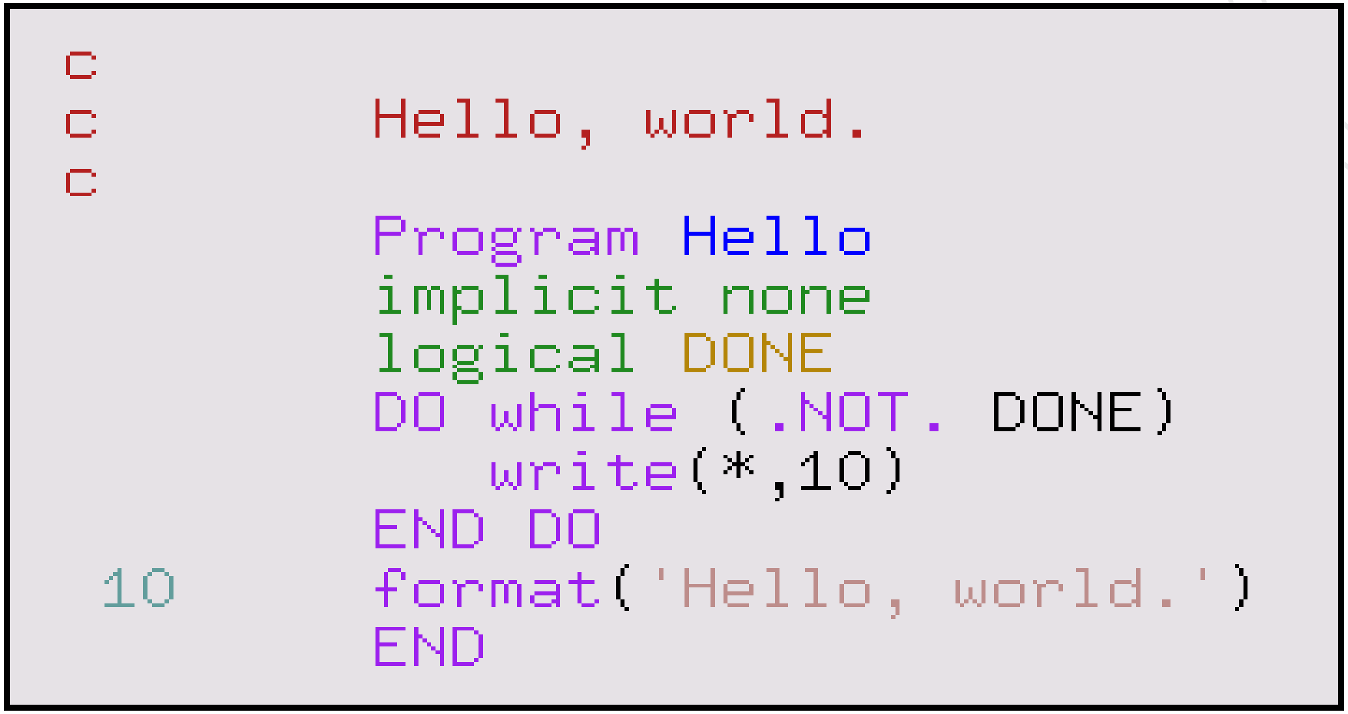
High-Level Language

- Example of high-level languages are C, Fortran, Pascal, COBOL and etc.

- Advantage:
  - Easy to program
  - Machine independent

- Disadvantages:
  - Requires translator (compiler or interpreter)

Example: FORTRAN Program



```fortran
c
c       Hello, world.
c
        Program Hello
        implicit none
        logical DONE
        DO while (.NOT. DONE)
           write(*,10)
        END DO
 10     format('Hello, world.')
        END
```

Example: C Program

```
#include <stdio.h>
main()
   {
     printf(“Selamat Belajar!¥n”);
     return 0;
   }
```

Very High-Level Language

- 4th generation of programming language.
- Also known as object-oriented or non-procedural language.
- It is much more user-oriented and allow programmers to develop programs with fewer commands.

Very High-Level Language

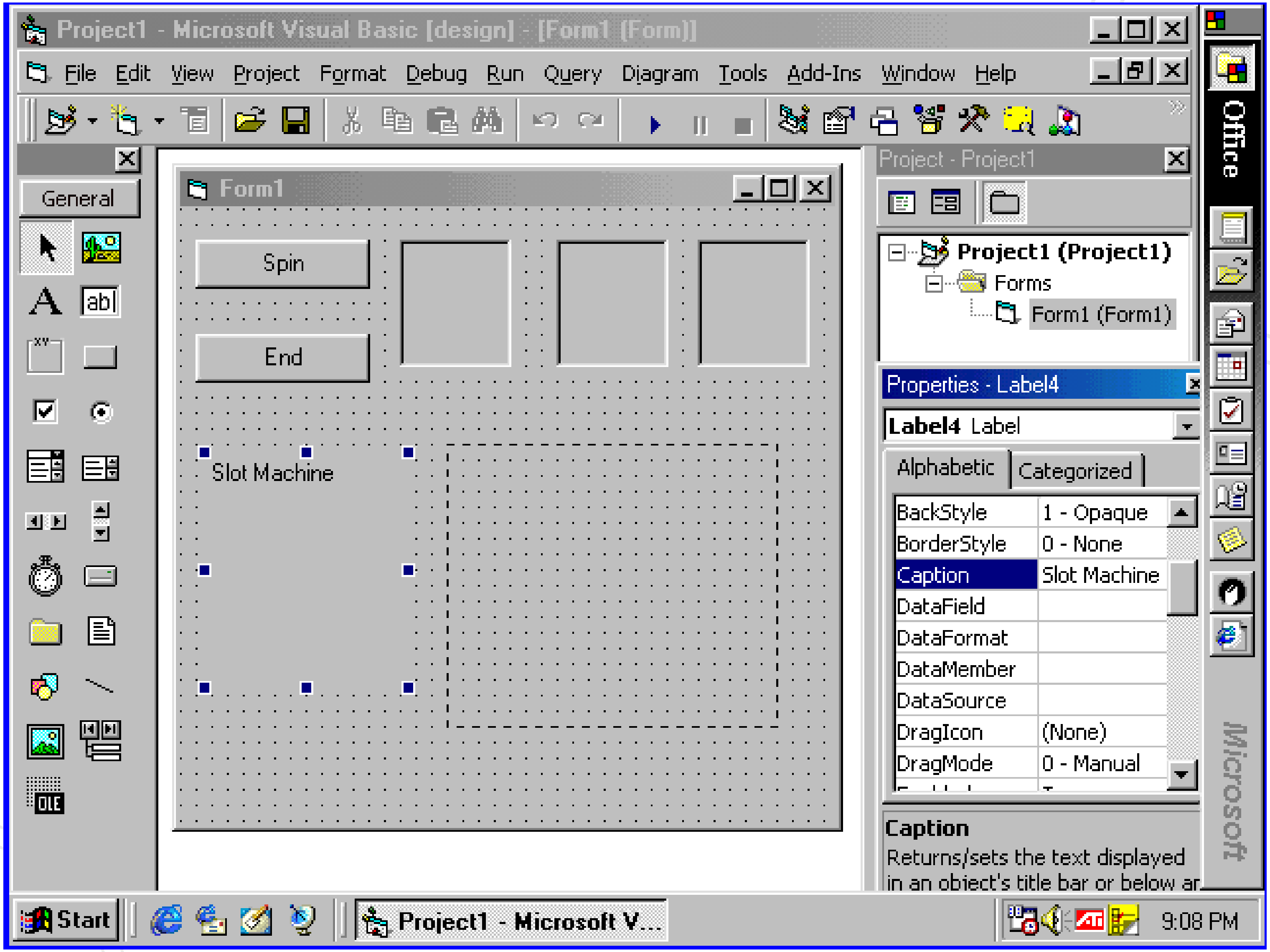
- Some of very high-level languages are also called RAD (*rapid application development*) tools.
- The use of visual in programming was also introduced in very high-level language.
- Example of very high-level languages are C++, Java and Visual Basic.

CLASS ACTIVITY

Find out and explain what are the following 4GL capabilities with an example of the Programming Language.

1. report generator
2. query language
3. application generator

Example: Visual Basic Program



Project1 - Microsoft Visual Basic [design] - [Form1 (Form)]

Properties - Label4

Label4 Label

BackStyle	1 - Opaque
BorderStyle	0 - None
Caption	Slot Machine
DataField	
DataFormat	
DataMember	
DataSource	
DragIcon	(None)
DragMode	0 - Manual

Caption
Returns/sets the text displayed in an object's title bar or below ar

Natural Language

- 5th generation of programming language.

- Natural languages use human language to give people a more natural connection with computers.

- Natural languages allow questions or commands to be framed in a more conversational way.

Natural Language

For example:

I WANT THE LIST OF SPPP2101 STUDENTS TO BE PRINTED AT 2.00 PM 19/01/2020

- Natural languages are part of the field of study known as *artificial intelligence (AI)*.
- AI are technologies that attempt to develop machine to emulate human-like qualities.

History Of Programming Language

1954 – IBM published Fortran

1957 – Math-Matic

1958 – FORTRAN II

1959 – COBOL (Common Business Oriented Language)

1962 – FORTRAN IV

1964 – BASIC created

1970 – Pascal created

1970 – Smalltalk created

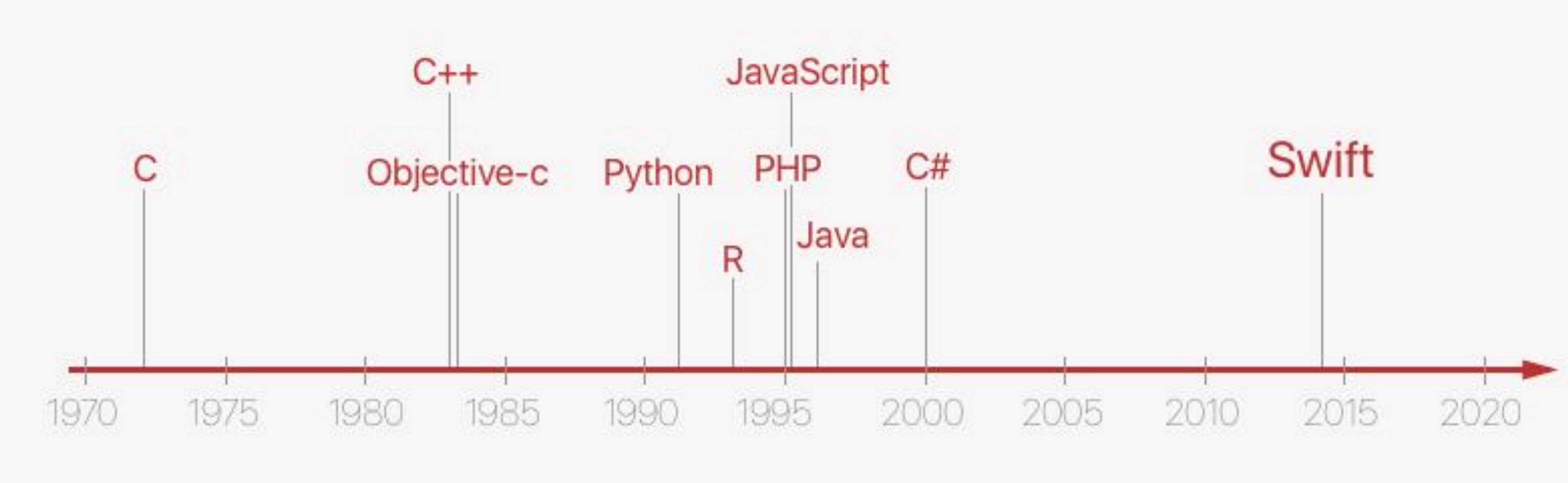
1972 – C created and becoming so popular

1979 – ADA language

1982 – dBase (The first database Programming Language)

1984 – Turbo Pascal created

TIMELINE


C++
JavaScript
C
Objective-c
Python
PHP
C#
Swift
R
Java
1970
1975
1980
1985
1990
1995
2000
2005
2010
2015
2020

THE END