principles of programming

Learning Assesment

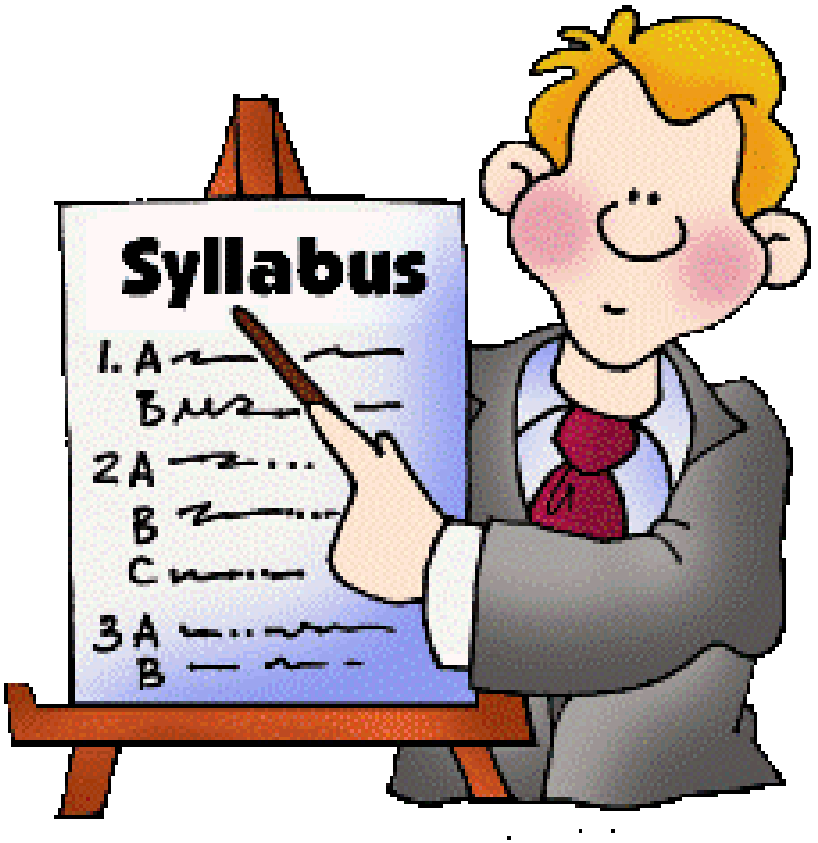
VII. Learning Assessment:				
No.	Assessment Tasks	Assessment day & date	Mark	Weight
1	Homework/Tasks/Assignments	2	20	20%
2	Quiz 1	4	5	5%
3	Midterm Exam	7	20	20%
4	Quiz 2	10	5	5%
5	Final Exam	16	50	50%
	Total			

References and resources

VIII. Learning Resources
1. Textbooks: • Sprancle, M., 2005, Problem Solving and Programming Concepts, 7th Edtion, Prrentice Hall, USA. • Walter Savitch, Problem Solving with C++ , 7th Edition, Addison Wesley.
2. Essential References: • Deitel and Deitel, 2000, How to Program, 5th edition, Inc. and Prentice Hall. • D.S. Malik, Thomson, 2007, C++ Programming: From Problem Analysis to Program Design, Third Edition, Course Technology.
3. Electronic Materials and Web Sites: • www.deakin.edu.au/~agoodman/Ctutorial.html • www.tldp.org/howto/c++programming.howto.html • www.vb-bookmark.com/cpptutorial.html

Course Structure

- Introduction
- Computer Programming
- Introduction to C++
- Input and output
- Variables
- Selections (if, switch)
- Loops (for, while, do-while)
- Arrays (1D, 2D)
- Functions
- Steps to application development



Why people use machines?

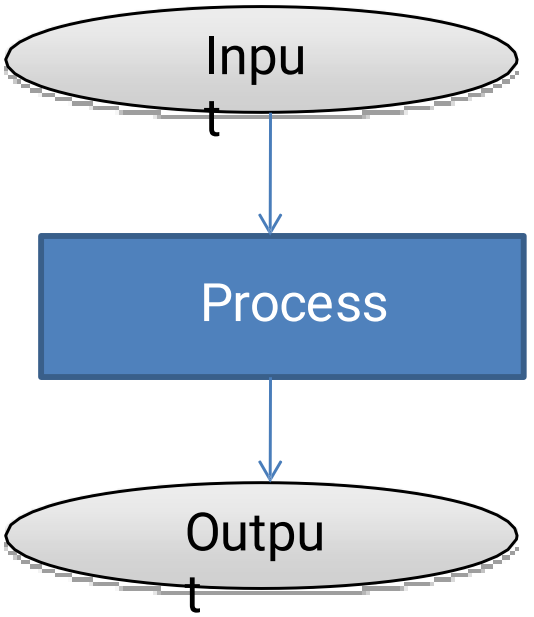
Machines

- Café Machine → make a tea
- ATM machine → money transaction
- Calculator → solve equation

- Each Machine has …
  - Input
  - Output
  - Process

Café Machine

- Input
  - Sugar
  - Water
  - Coffee
  - Milk
- Output
  - Tea
- Process
  - ????

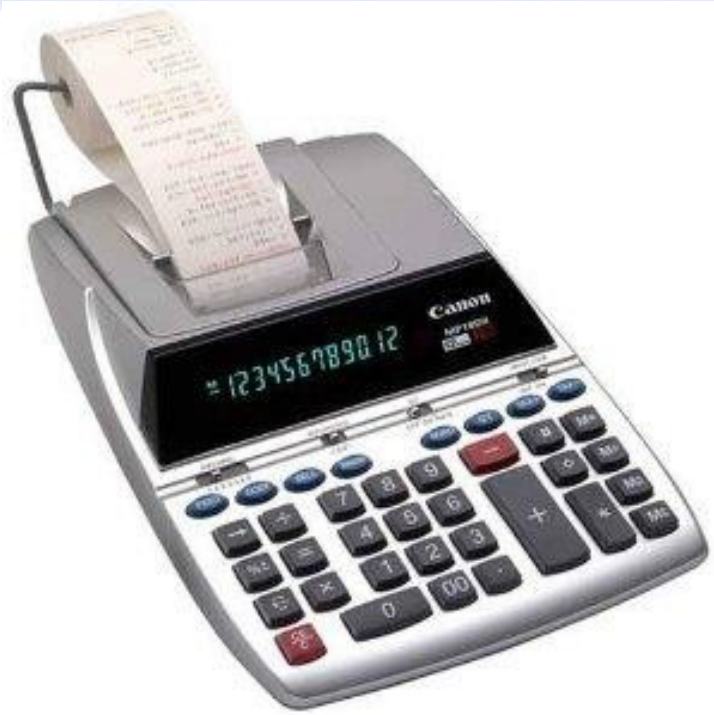


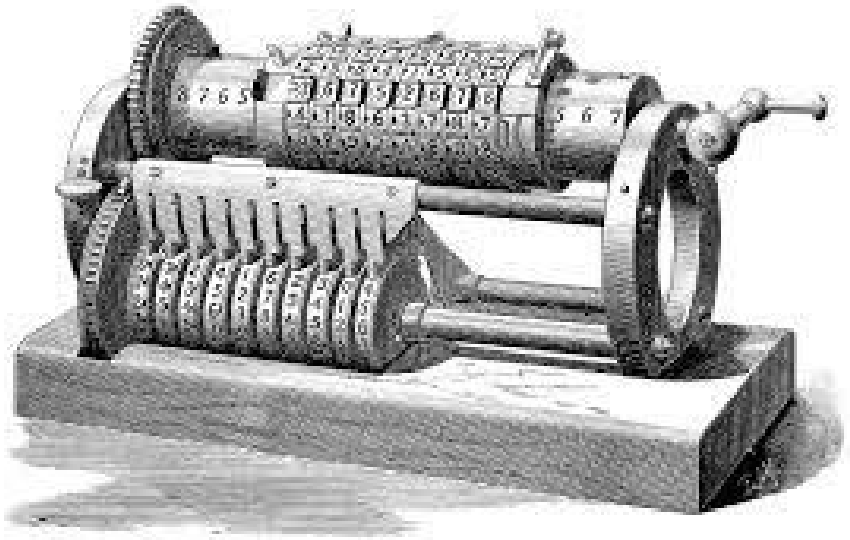
ATM machine

- Input
  - ???
- Output
  - ???
- Proces
  s
  - ???

Calculator

- Input
  - ???
- Output
  - ???
- Proces
  s
  - ???





Process of a Machine

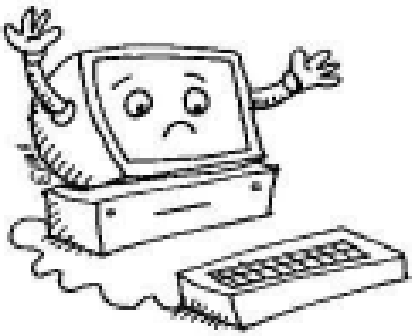
- Work through the instructions given to it
  - Read
  - Write
  - Do some calculations
  - Go to next instruction
- The sequence of instructions is call a **Program**

What is a Programming?

- Programming is the way to give instructions
- Each program has
  - Start
  - Some work
  - End
- Program is given/execute through the machine understood language

A Computer

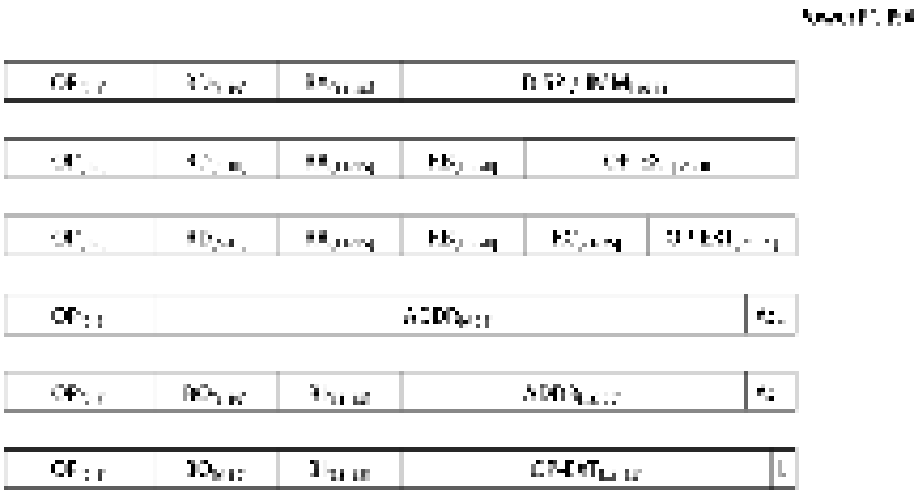
- Machine that can solve problems for people by carrying out instructions given to it
- The sequence of instructions is called Program
- The language machine can understand is called machine language



Machine Language

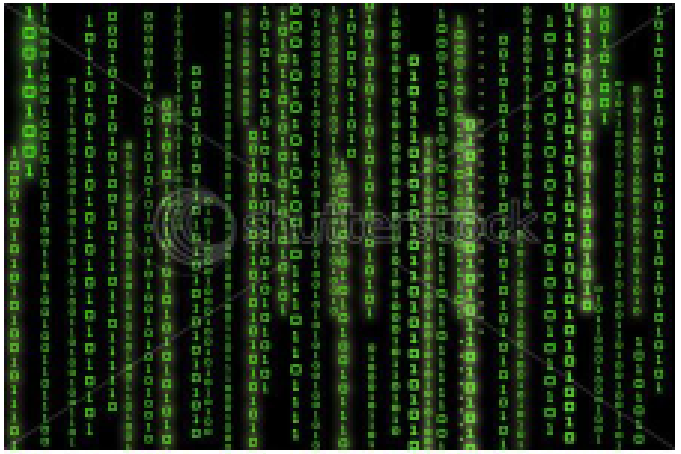
- Machine language is a set of instructions executed directly by a computer's central processing unit (CPU)

```
10100001 10111100 10010011 00000100
00001000 00000011 00000101 11000000
10010011 00000100 00001000 10100011
11000000 10010100 00000100 00001000
```



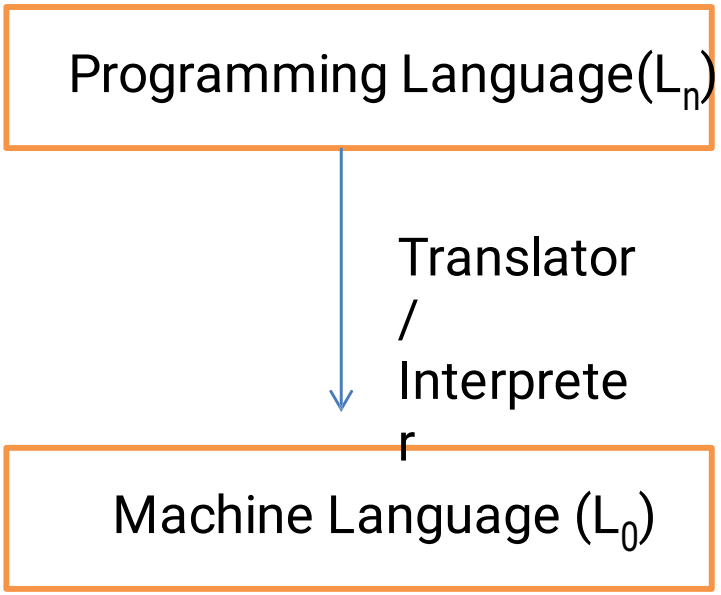
Machine Language contd..

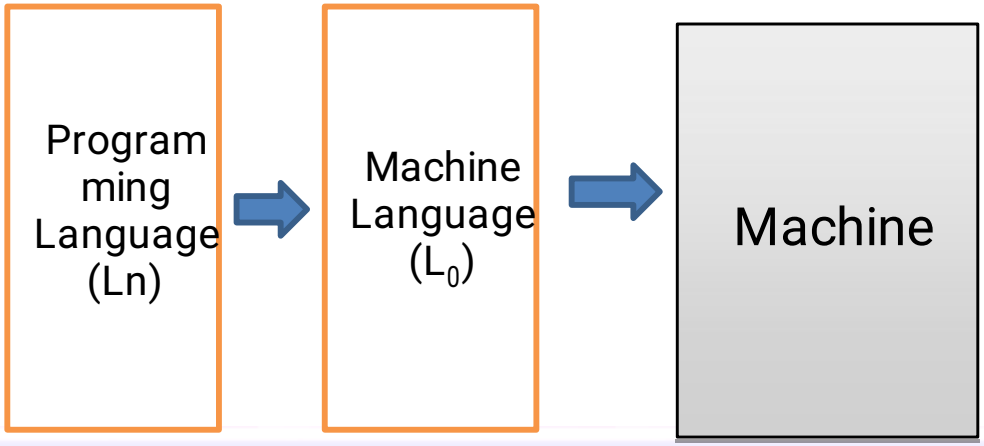
- Advantages
  - Machine can directly access (Electronic circuit)
  - High Speed
- Disadvantages
  - Human cannot identify
  - Machine depended (Hardware depended)



Computer Language(s)

- Artificial
- Human readable
- Ex: C/C++, Java



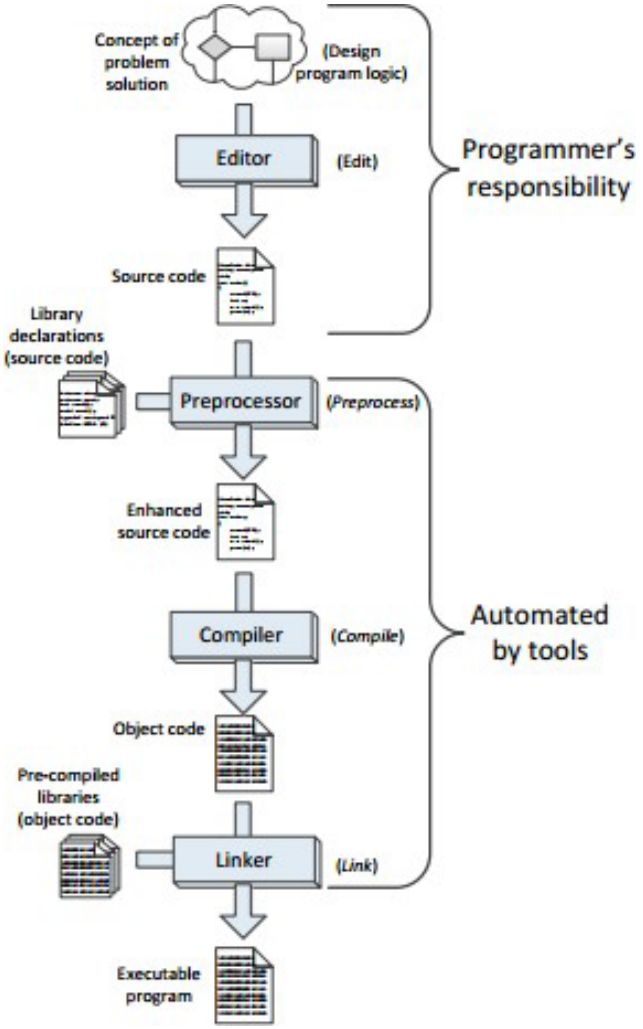


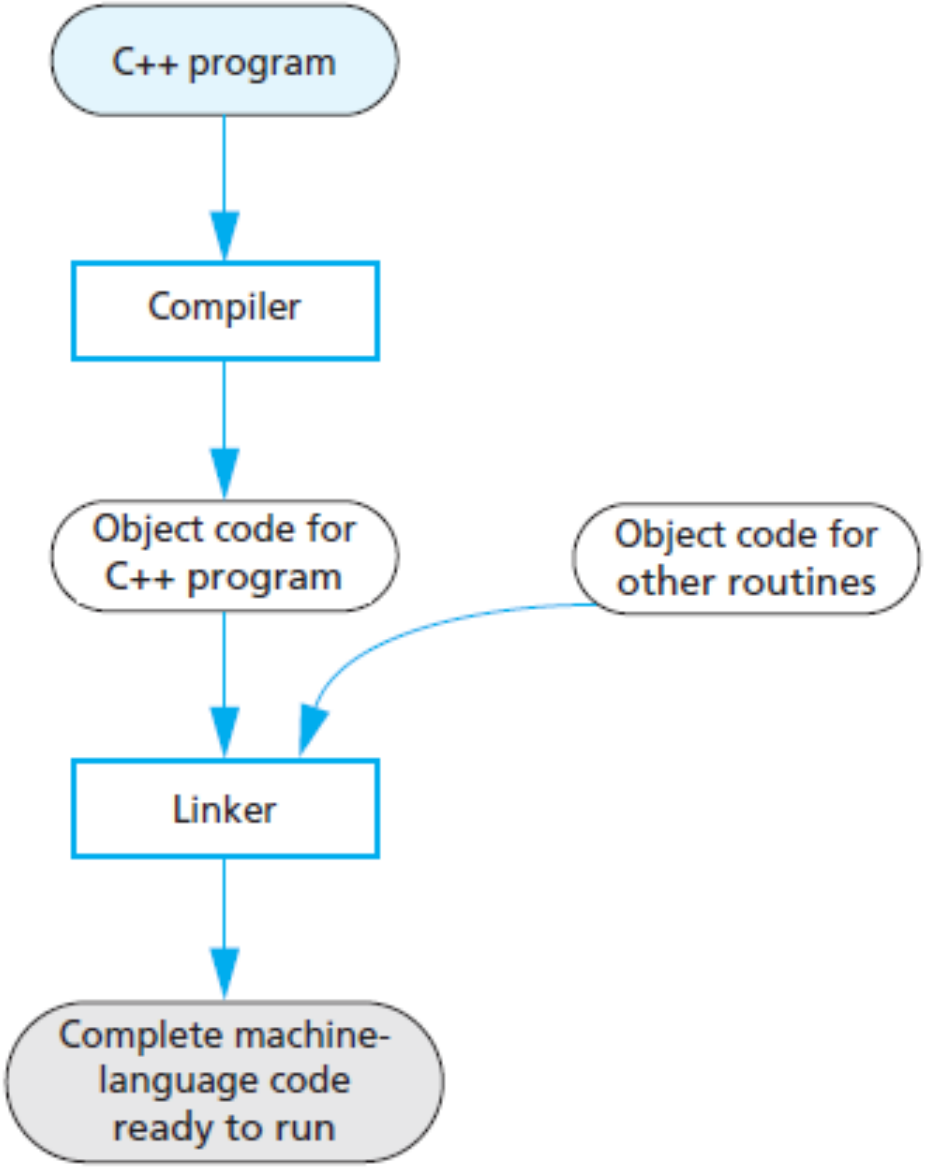
Compilers

- Translate high-level language to machine language
- Input (**Source code**)
  - The original program in a high level language
- Output (**Object code/ Machine Code**)
  - The translated version in machine language
- Example
  - C++ compiler, JAVA Compiler etc.

Compilation steps

- **Preprocessor**—adds to or modifies the contents of the source file before the compiler begins processing the code
- **Compilers**. A *compiler* translates the source code to target code
- **Linker**—combines the compiler-generated machine code with precompiled library code or compiled code from other sources to make a complete executable program

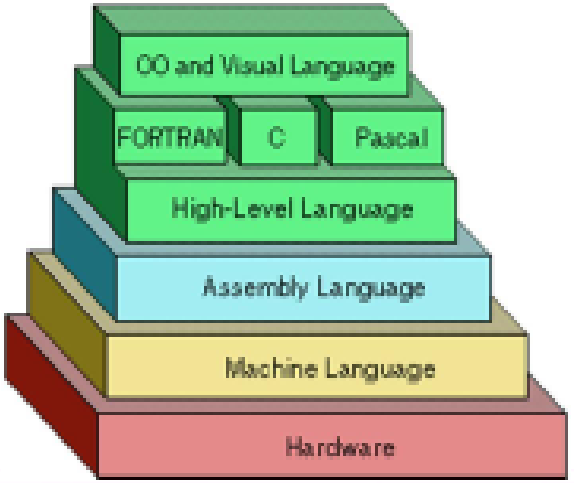



C++ program
Compiler
Object code for C++ program
Object code for other routines
Linker
Complete machine-language code ready to run

Programming language generations

This classification is used to indicate increasing power of programming styles

1. *First-generation programming languages*
2. Second-generation programming languages
3. Third-generation programming languages
4. Fourth-generation programming languages
5. Fifth-generation programming languages

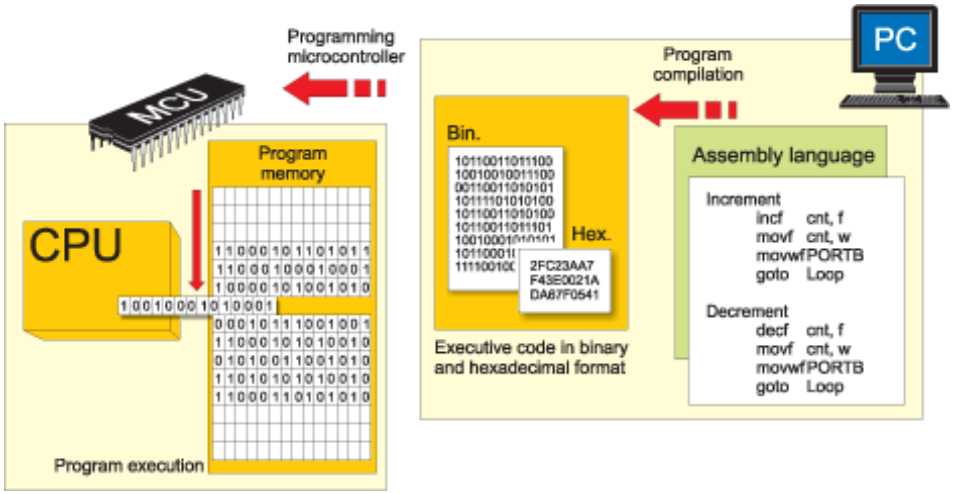


First-generation programming language (1GL)

- Is a machine-level programming language
- Translator isn't used to compile
- The instructions in 1GL are made of binary numbers, represented by 1s and 0s
- Advantage
  - The code can run very fast and very efficiently because the instructions are executed directly by the CPU
- Disadvantage
  - When an error occurs, the code is not as easy to fix

Second-generation programming language(2GL)

- Assembly language.
- Properties
  - The code can be read and written by a programmer
  - The language is specific to a particular processor family and environment
- Used in kernels and device drivers




```
MOV AX, 0x3C
MOV BX, 0000000000001010B
ADD AX, BX
MOV BX, 14
SUB AX, BX

MOV AH, 04CH
INT 21H
```

MOV eax, 3
MOV ebx, 4
ADD eax, ebx, ecx

Third-generation programming languages (3GL)

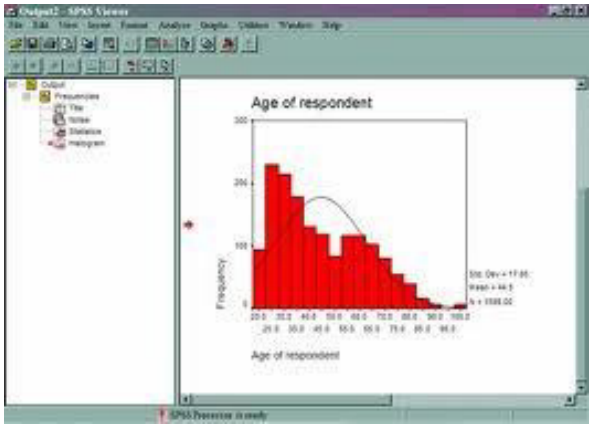
- Languages are more programmer-friendly
- Example
  - C, C++, C#, Java, BASIC and Pascal
- Support structured programming.
- Must be translated into machine language by a compiler or interpreter
- Advantages
  - Easier to read, write, and maintain

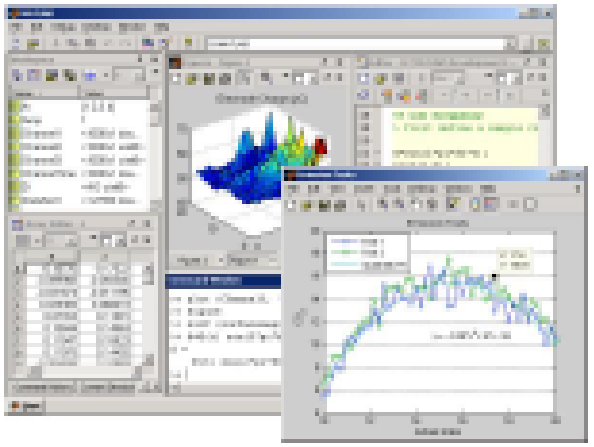
```java
// class declaration
public class ProgrammingExample {

    // method declaration
    public void sayHello() {

        // method output
        System.out.println("Hello World!");
    }
}
```

```cpp
#include <iostream>

using namespace std;
int main()
{
int i,num;
cout<<"Enter a number\t";
cin>>num;
i=0;
while(i<=num)
```

Fourth-generation programming languages(4GL)

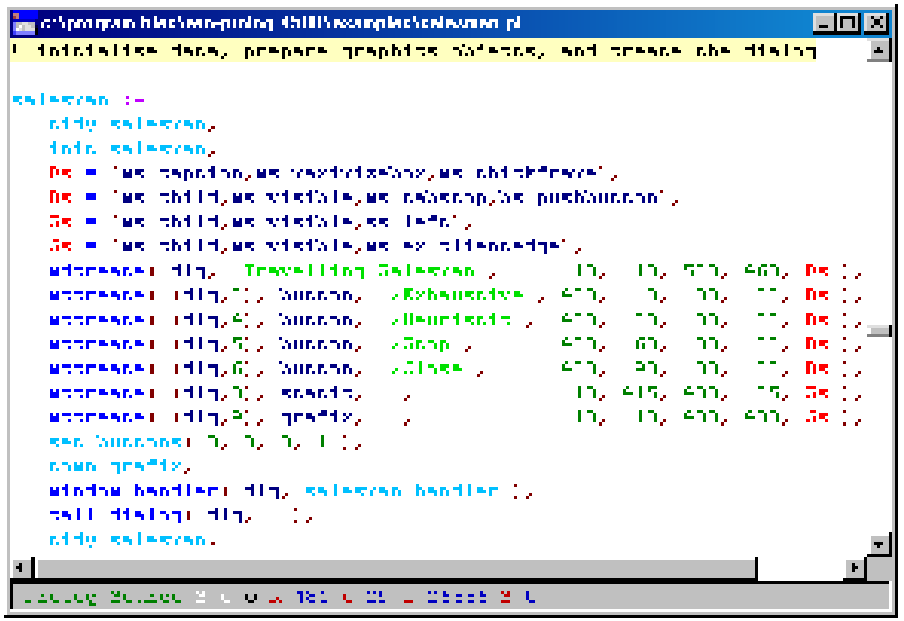
- Designed to reduce programming effort
- Consist of
  - Set of libraries
  - CRUD generators
  - Report generators
  - DBMS
  - Visual design tool and integration API
- Different types of 4GLs
  - Table-driven (codeless) programming
    - PowerBuilder
  - Data management
    - SAS, SPSS
  - Report-generator programming languages
    - Oracle Developer Suite





Fifth-generation programming language(5GL)

- Based on **solving problems** using constraints given to the program, rather than using an algorithm written by a programmer
- Use mainly in Artificial Intelligence research
- Example
  - Prolog, OPS5, and Mercury



Example of prog.

- Assembly Language Program
  - Device drivers, Virus
- C / C++ Program
  - Device drivers , DLLs
- JAVA / Visual C++ Program
  - Desktop applications, Web Applications
- Prolog
  - AI based applications, Games, Translators

Exercise

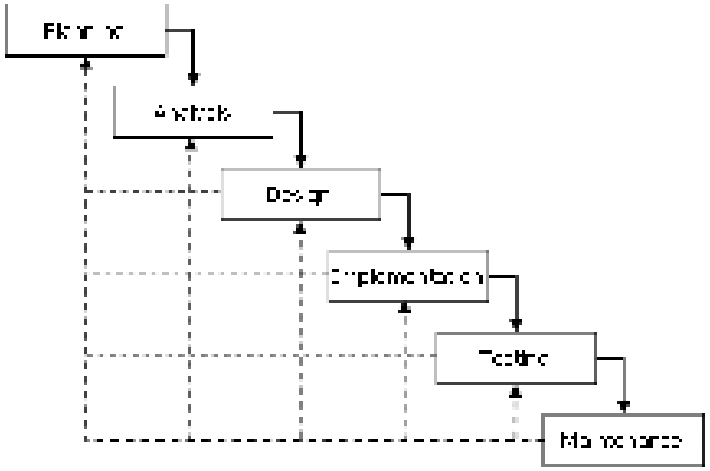
- To solve the following problems, identify the input, output and the process

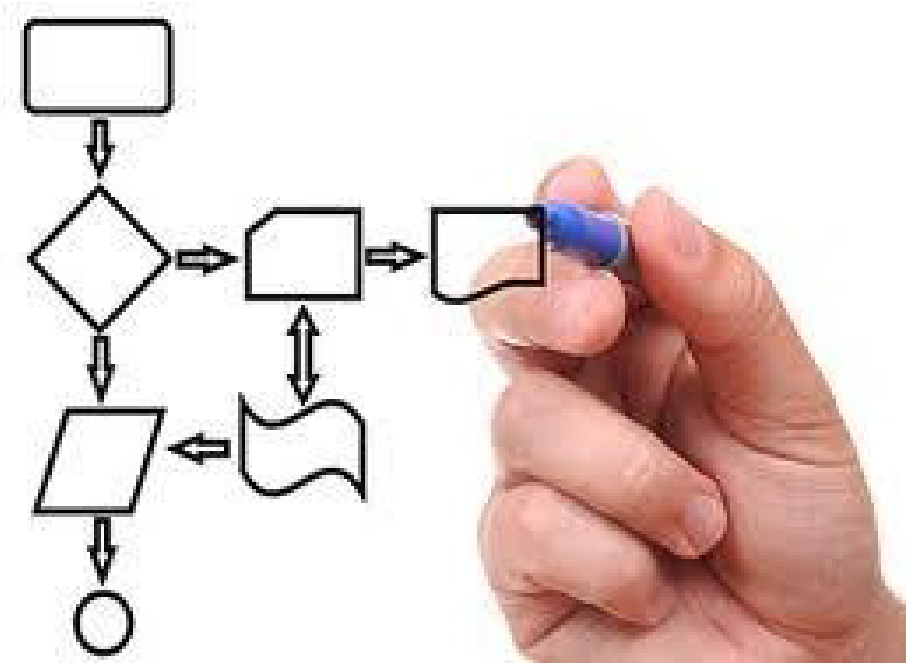
  Find the area of a room Search a place of a city

  Calculate grade for the given mark Get some amount from ATM machine

Stages of Computer Programming

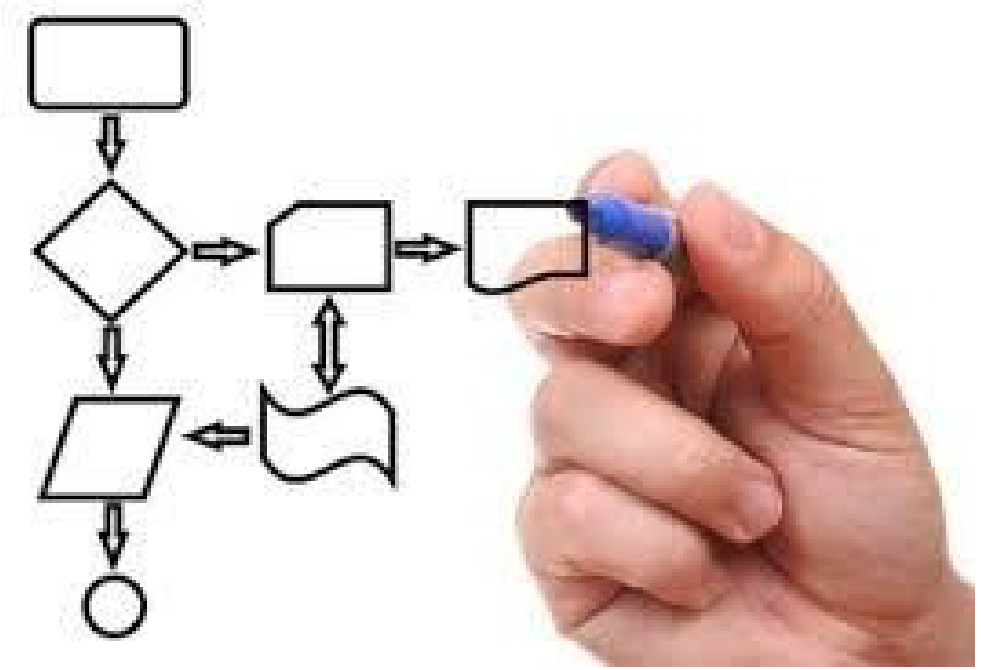
1. Planning
2. *Analysis*
3. *Design*
4. *Implementation*
5. *Testing*
6. *Maintenance and update*



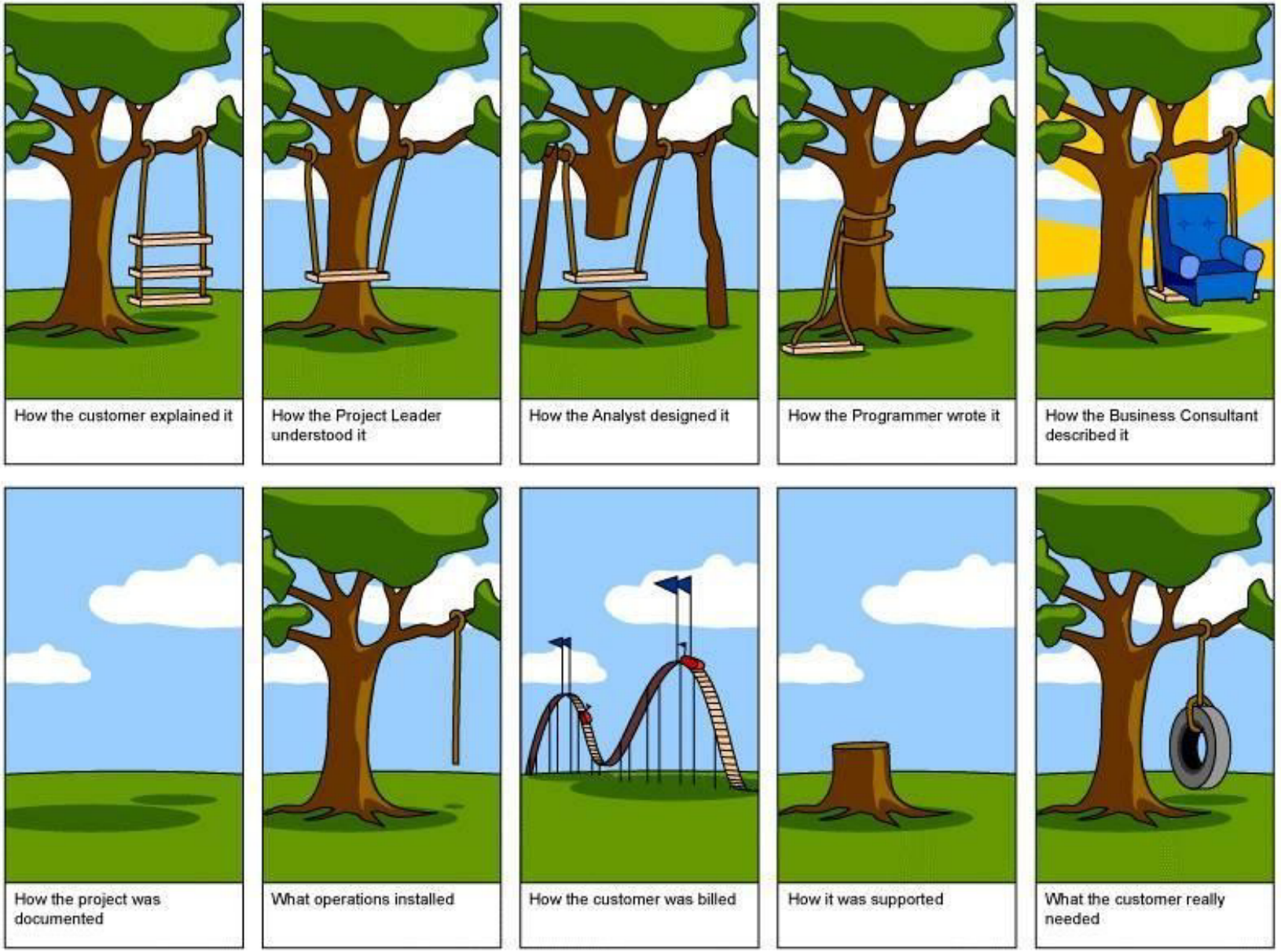


Software Design

- Software design is a process of problem-solving and planning for a software solution
- Types
  - Top down
  - Bottom up
  - Module design
- Use to describe
  - Algorithm
  - Flowchart
  - Pseudo code

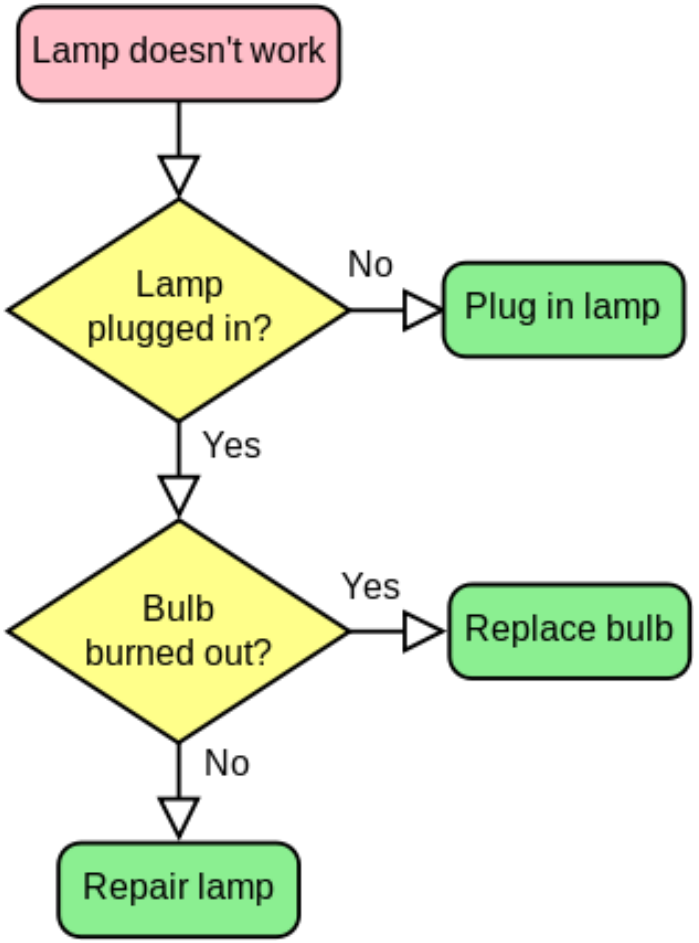


Design process



Design cont….

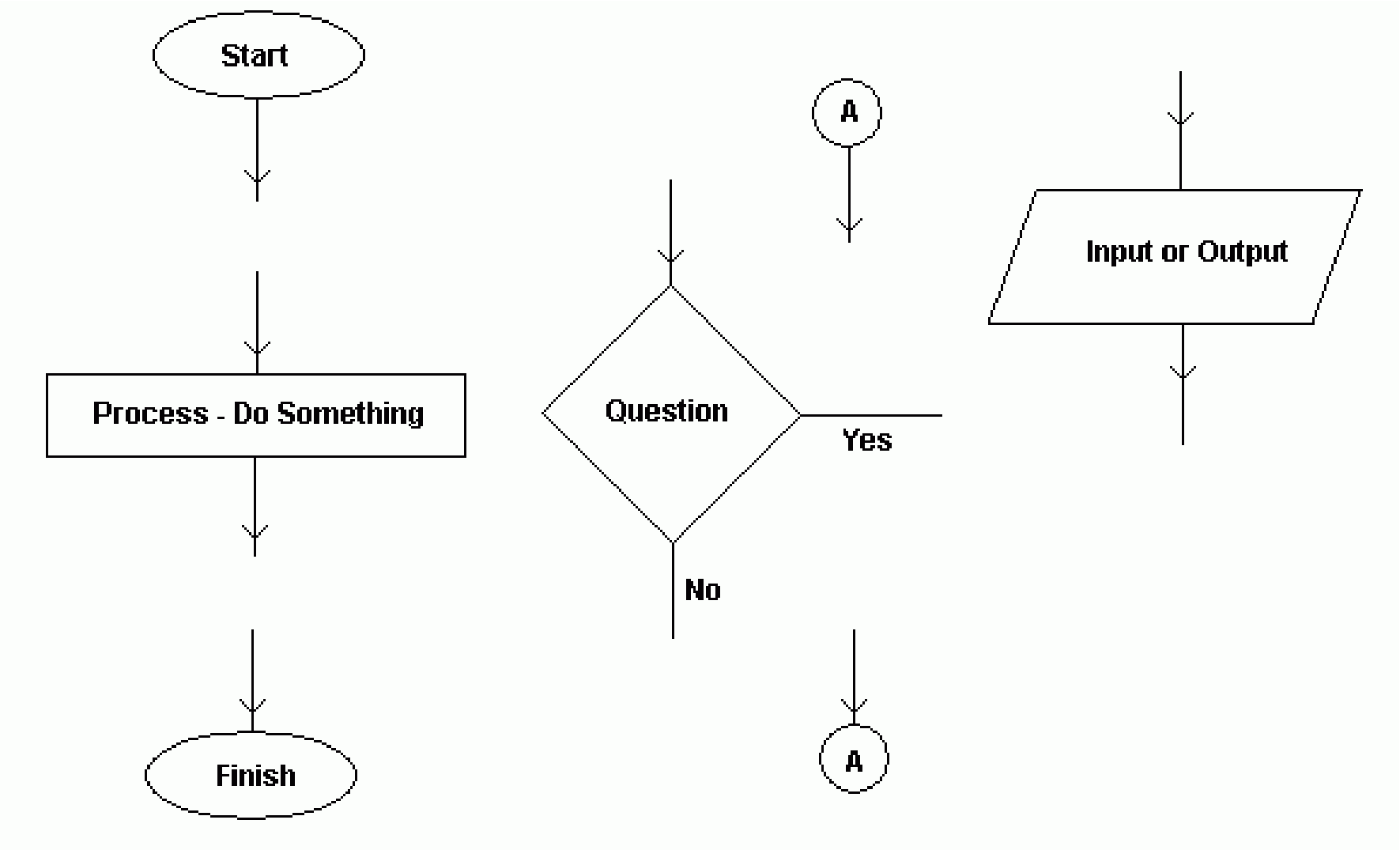
- Flowchart:
  - is a type of diagram that represents an algorithm or process
  - Gives diagrammatic representation solution to a given problem
  - Use in analyzing, designing, documenting or managing a process or program



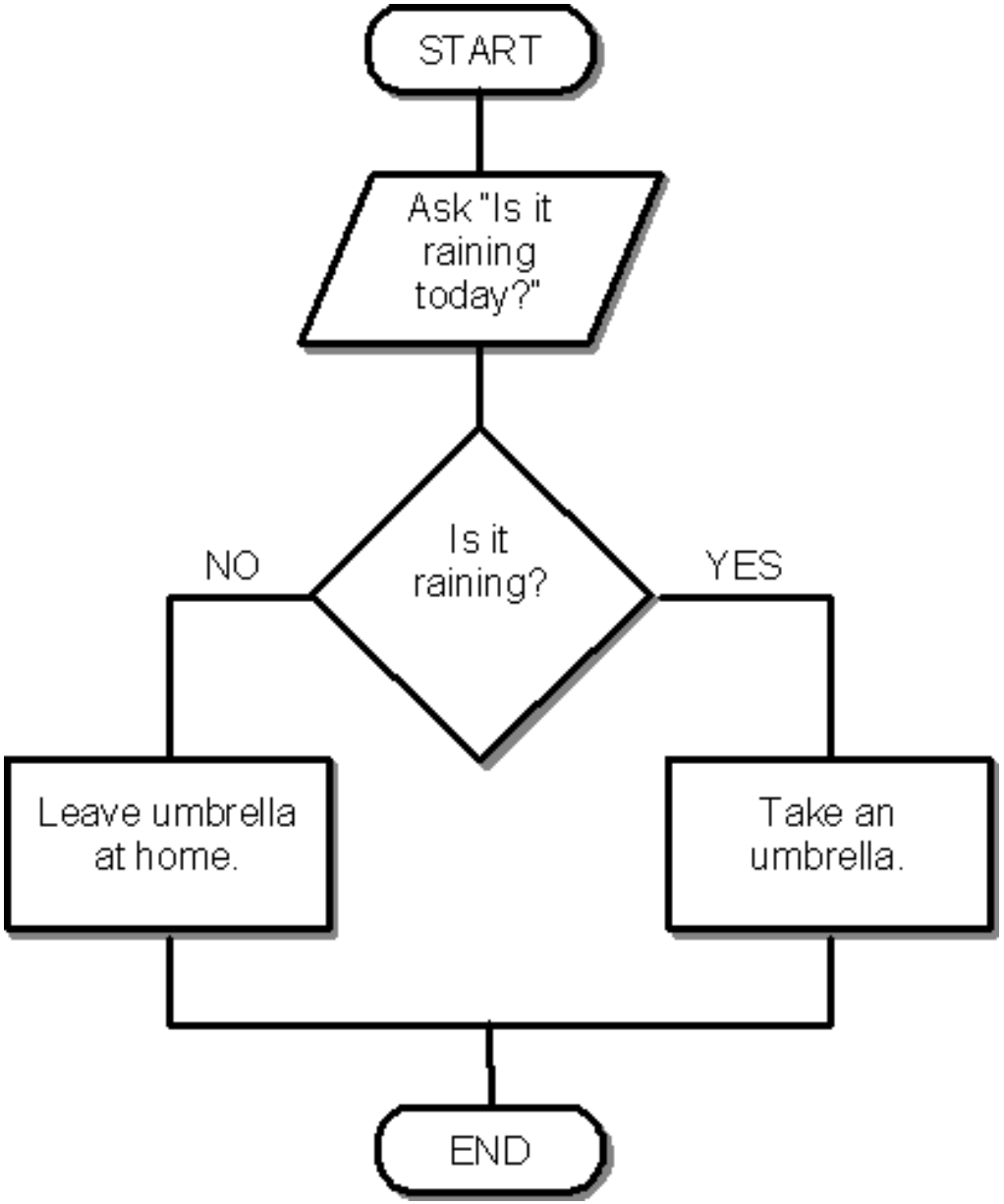
Flowchart- building blocks

Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g, PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).

Flowchart-building blocks contd...



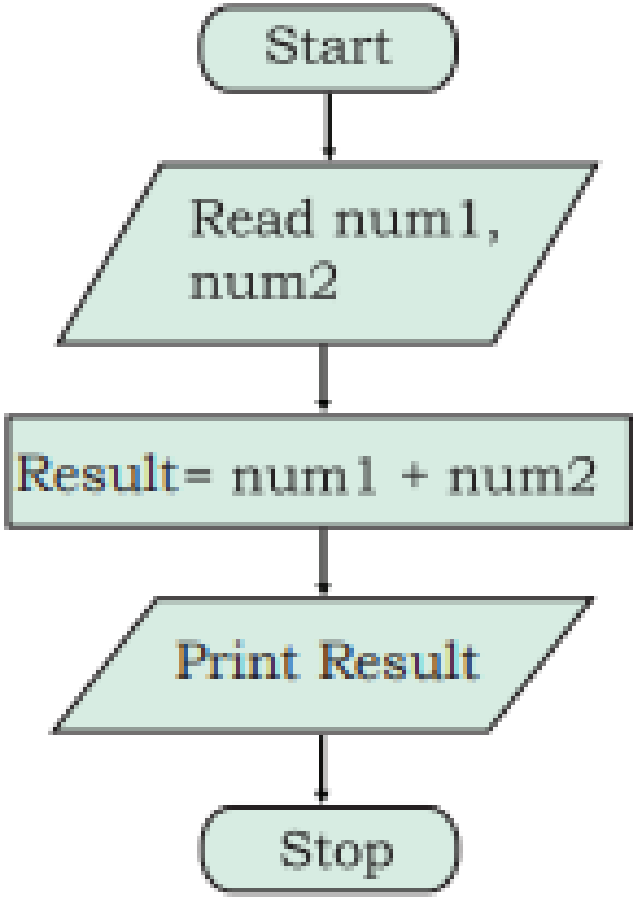
Example


START
Ask "Is it raining today?"
Is it raining?
NO
YES
Leave umbrella at home.
Take an umbrella.
END

Example

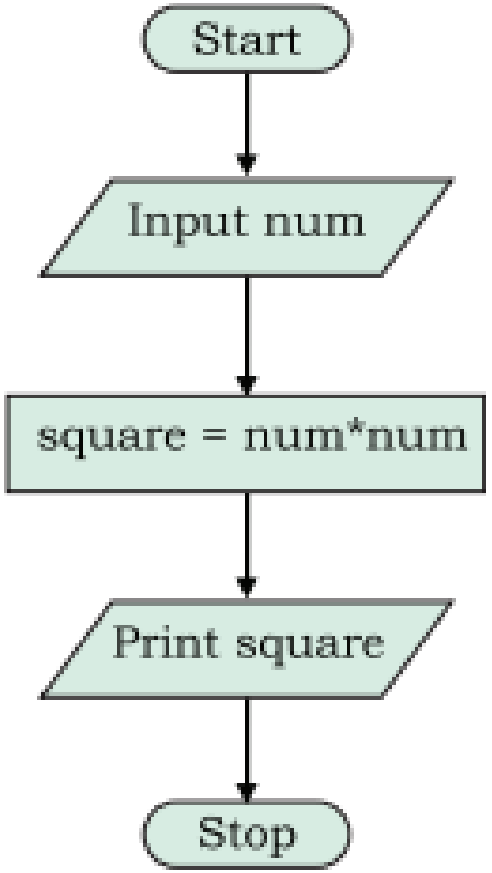
- Draw a flow chart to display total of the the two numbers

Name	Symbol	Use in flowchart
Oval		Denotes the beginning or end of a program.
Flow line		Denotes the direction of logic flow in a program.
Parallelogram		Denotes either an input operation (e.g., INPUT) or an output operation (e.g, PRINT).
Rectangle		Denotes a process to be carried out (e.g., an addition).
Diamond		Denotes a decision (or branch) to be made. The program should continue along one of two routes (e.g., IF/THEN/ELSE).


Start
Read num1, num2
Result= num1 + num2
Print Result
Stop

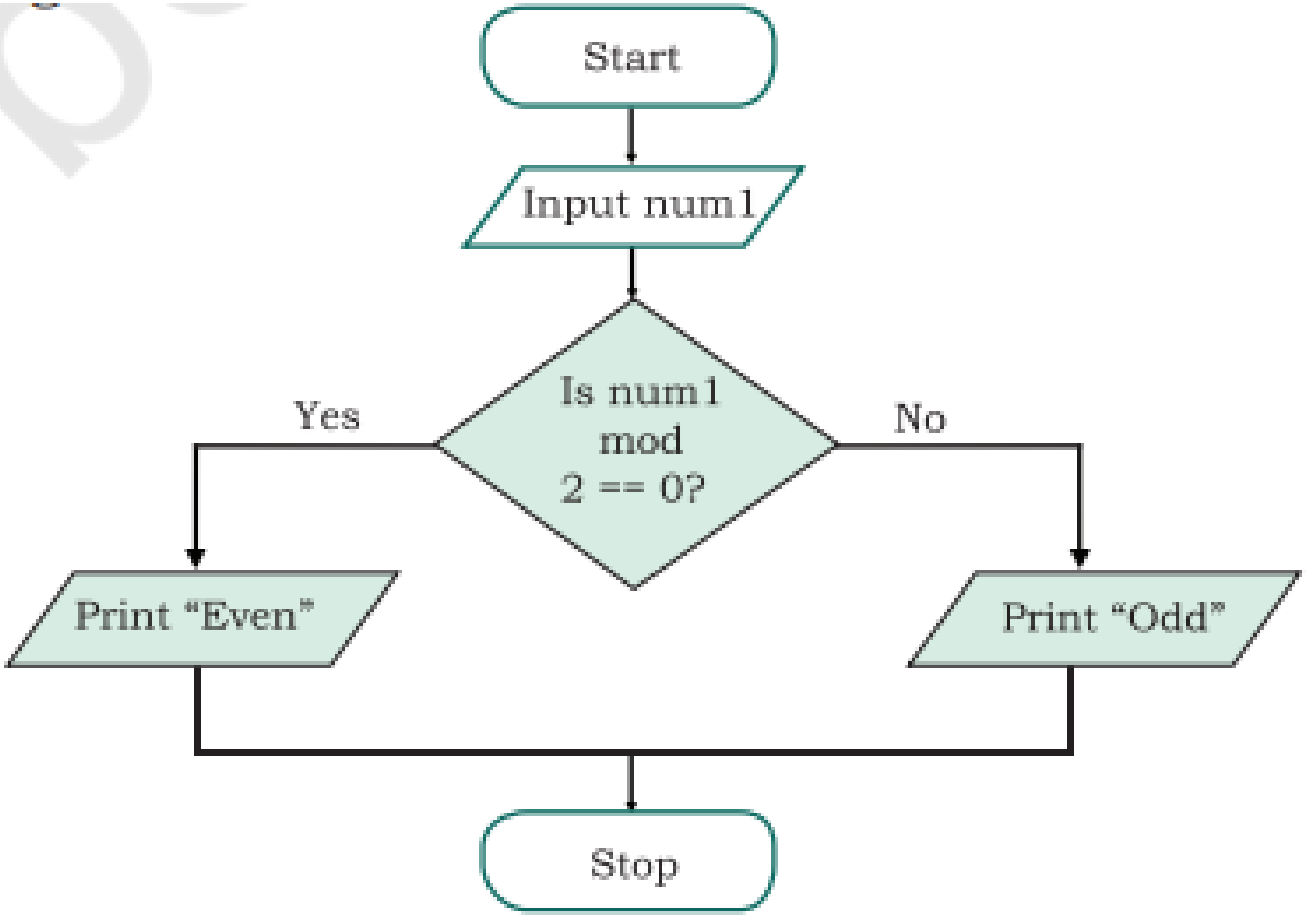
Example

Draw a flow chart to find the square of a number.



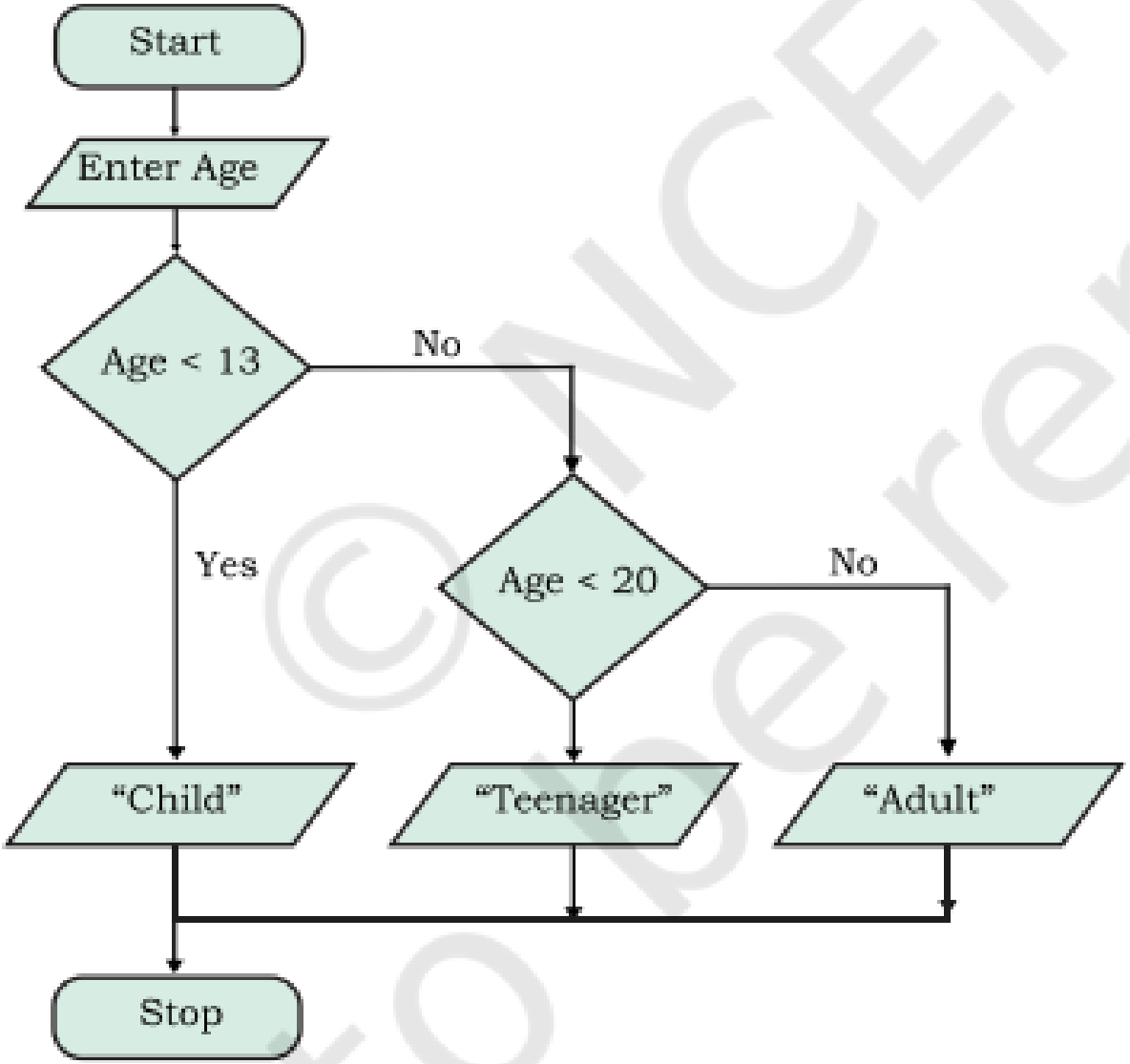
Example

Draw a flow chart to check whether a number is odd or even



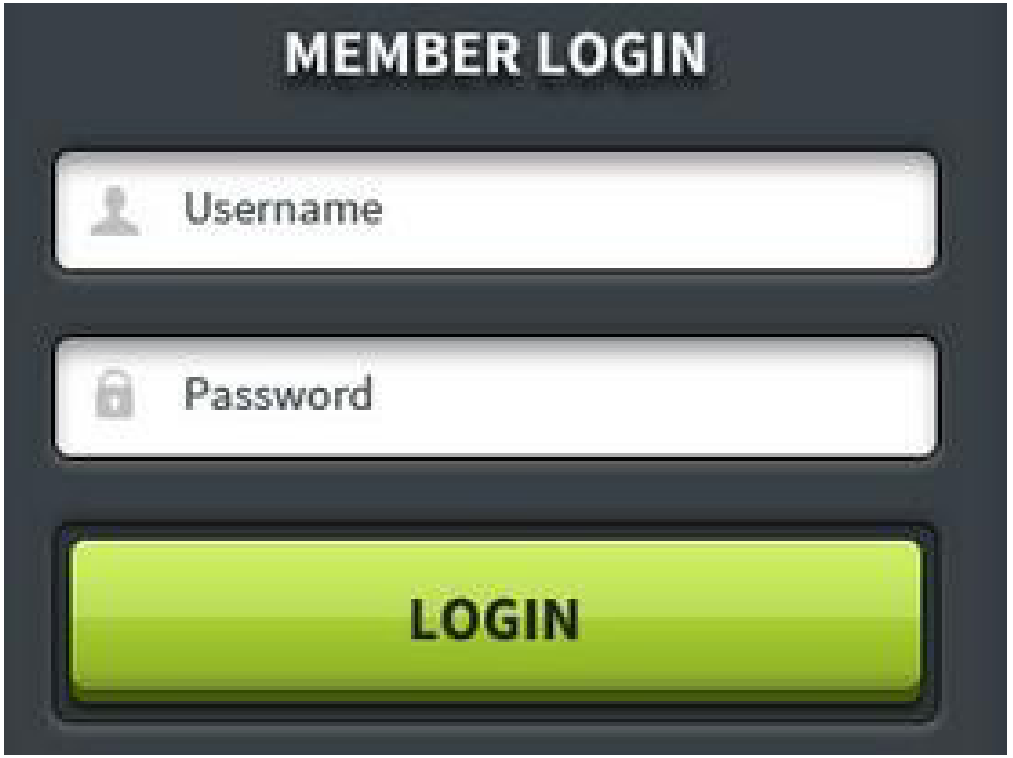
Example

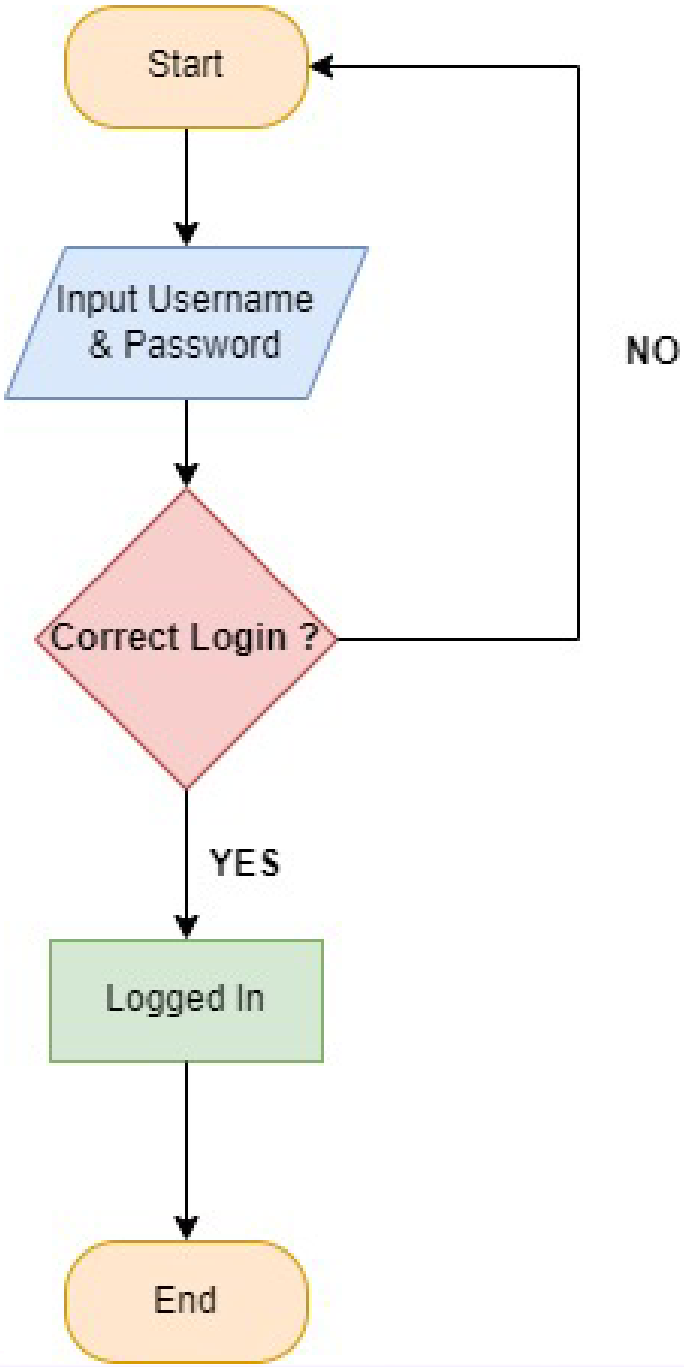
draw a flowchart where multiple conditions are checked to categorise a person as either child (<13), teenager (>=13 but <20) or adult (>=20)



Example

- Draw a flow chart to identify correct login for the following interface




Start
Input Username
& Password
Correct Login ?
NO
YES
Logged In
End

Programming with IDE

- IDE : **integrated design environment**
- consists of
  - source code editor
  - compiler and/or an interpreter
  - build automation tools
  - Debugger
  - Construction of a GUI
  - Class browser
  - Object inspector
  - Etc.

Programming IDEs

- Eclipse

  Eclipse is a multi-language software development environment

  http://www.eclipse.org/
- Code:blocks

  http://www.codeblocks.org/
- Netbeans

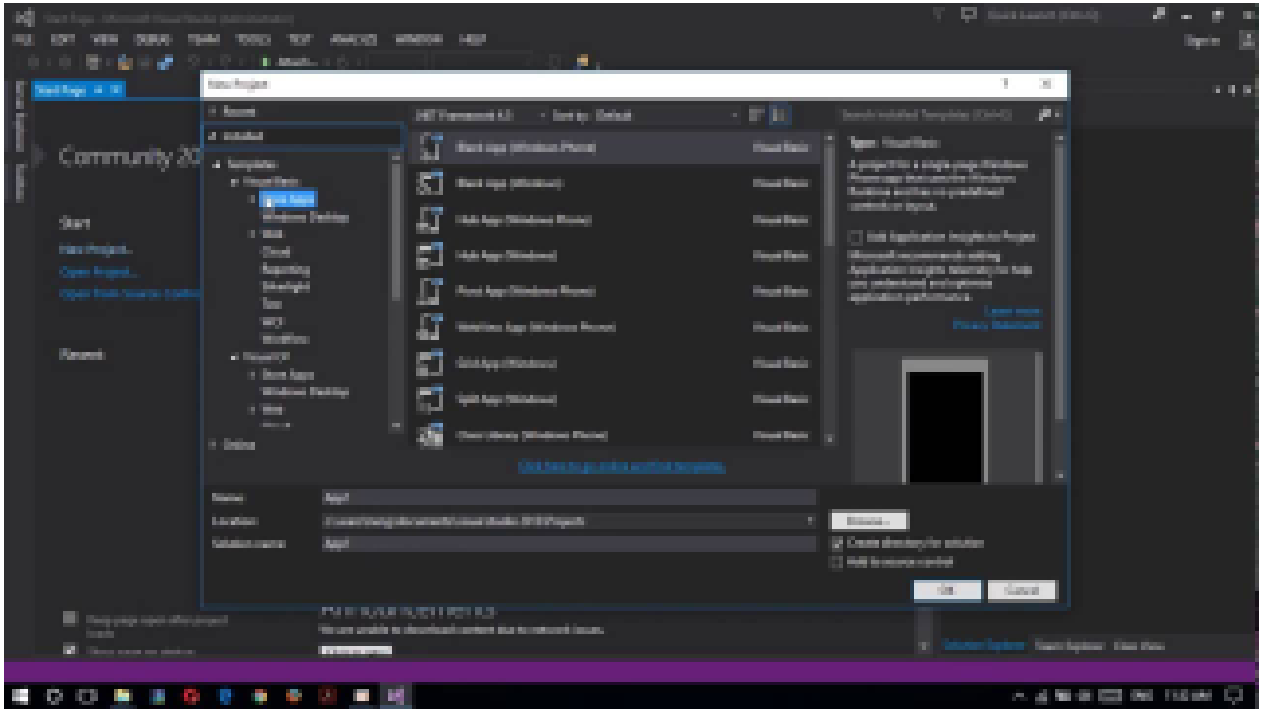
  http://netbeans.org/
- Microsoft Visual Studio

  http://www.microsoft.com/visualstudio/en-us

Exercise

- Download:
  https://visualstudio.microsoft.com/downloads/

Visual Studio 2022



Summary

- What is a machine?
- Computer program
- Programming languages
- Design
- Characteristics of a good computer program
- Tools & Tips for computer programmi