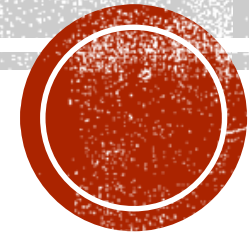




# COMPUTER PROGRAMMING CONCEPTS



CS&IT 1101

Instructor: Shakar H. Salih

E-mail: [shakar.salih@uhd.edu.iq](mailto:shakar.salih@uhd.edu.iq)



# LECTURE 2: PROBLEM SOLVING WITH COMPUTER

# OUTLINES

- Characteristic of a Good Program
- Planning the Computer Program
- Different methods for solving problem
  1. Algorithm
  2. Flowchart
  3. Pseudocode

# PROGRAM

- Sequence of instruction written in a programming language.
- Generally every program takes an input, manipulates it and provides an output.



# CHARACTERISTIC OF A GOOD PROGRAM

The different aspects of evaluating program are:

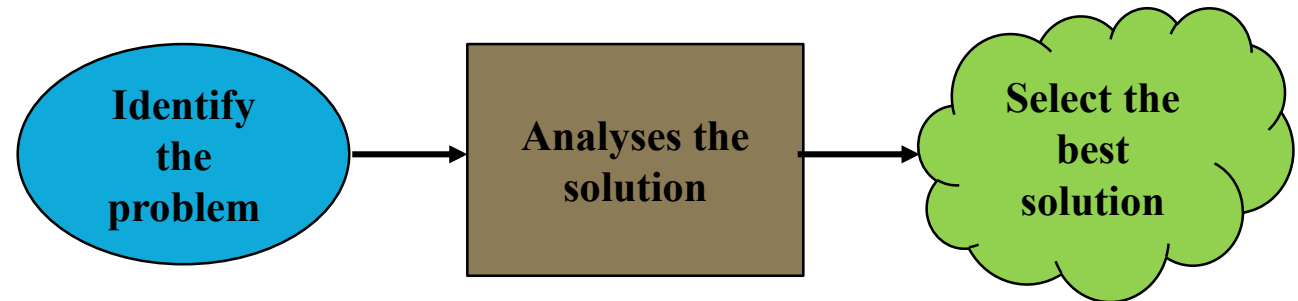
1. **Efficiency:** It is of three types: programmer effort, execution time and memory space .
2. **Flexibility:** A program that can serve many purposes is called a *flexible* program.
3. **Reliability:** Program having such ability are known as reliable.
4. **Portability:** A program is called *portable* if it can be transferred from one system to another with ease.
5. **Robustness:** It provides meaningful results for all inputs (correct or incorrect).
6. **User Friendly:** A program that can be easily understood even by a novice is called a *user friendly* .
7. **Self-documenting code:** The source code which uses suitable names for the identifiers.

**Just Writing Code Is Not Sufficient To Solve A Problem.**

**Program Must Be Planned Before Coding In Any Computer Language Available.**

# PLANNING THE COMPUTER PROGRAM

- Planning makes program:
  1. Efficient
  2. Effective
- Problem solving is one of the most significant advantages of a computer. Before writing programs, it is a good practice to understand
  - ✓ the complete problem
  - ✓ Analyze the various solution
  - ✓ Arrive at the best solution.



# STAGE WHILE SOLVING A PROBLEM USING COMPUTER

- There are many steps to be done before and after writing code:
  1. Problem Analysis
  2. Algorithm Development
  3. Flowcharting
  4. Program Coding
  5. Compilation and Execution
  6. Debugging and Testing
  7. Documentation



# PROBLEM SOLVING STEPS

1. Identify the problem
2. Understand the problem
3. Identify alternative ways to solve the problem
4. Select the best way to solve the problem from the list of alternative solution
5. List instructions that enable you to solve the problem using the selected solution
6. Evaluate the solution

# DIFFERENT METHODS FOR SOLVING PROBLEM

1. Algorithm
2. Flowchart
3. pseudocode

# ALGORITHM (1/7)

- A sequential solution of any problem
- Effective procedure for solving a problem in finite number of steps.
- A procedure for solving a problem in term of
  1. The **action** to execute
  2. The **order** in which these action execute

Is called **Algorithm**

# RISE-AND-SHINE ALGORITHM (2/7)

Getting out of bed and going to work:

1. Get out of bed;
2. Take off pajamas;
3. Take a shower;
4. Get dressed;
5. Eat breakfast;
6. Carpool to work.

# ALGORITHM (3/7)

## **Example 1:** Algorithm for sum of two numbers

Step 1: Start

Step 2: Enter two numbers in a and b

Step 3: Add a and b and store in c

Step 4: Display c

Step 5: End

# ALGORITHM (4/7)

**Example 2:** Algorithm to check whether the input number is even or odd

Step 1: Start

Step 2: Enter a number in a

Step 3: Check if it is even or odd

Step 4: If even print “even” ( $a \bmod 2 = 0$  )

Step 5: Else print “odd”

Step 6: End

# ALGORITHM (5/7)

**Example 3:** An algorithm to find largest of three different numbers:

Step 1: input A, B, C

Step 2: if (A > B) then

    if (A > C) Then

        print “Biggest number is”, A

    Else

        print “Biggest number is”, C

Else

    if (B > C) then

        print “Biggest number is”, B

    Else

        print “Biggest number is”, C

Step 3: End

# ALGORITHM (6/7)

## ■ **Example 4:** Exchanging values of two variables

Step 1: Start

Step 2: Input A, B

Step 3: Temp  $\leftarrow$  A

A  $\leftarrow$  B

B  $\leftarrow$  Temp

Step 4: Write “ Exchanged values are” , A, B

Step 5: End



# ALGORITHM (7/7)

## ■ **Example 5:** Algorithm to get average 3 tests of students

Step 1: Get three scores

Step 2: Add them together

Step 3: Divide by three to get the average

Step 4: Display the average

Step 5: Repeat step 1 to 4 for next student

Step 6: stop

# PROPERTIES OF ALGORITHM

- Finiteness
- Properly defined
- Input
- Output
- Effectiveness
- Independent to any other programming language

# THREE FEATURES OF ALGORITHM

- *Sequence*

- Each step in the algorithm is executed in specified order. If not, the algorithm will fail.

- *Decision*

- We have to make a decision to do something.

- If the outcome of the decision is true, one thing is done; otherwise, other.

- **If condition then process1**

**OR**

- **If condition  
then process1  
Else process2**

- *Repetition*

- – *For example*

**Repeat**







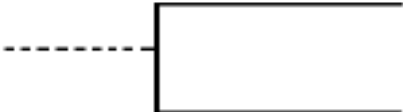
# FLOWCHART

- Diagrammatic representation of algorithm
- An important programming tool
- Solving a problem using figures
- Different figures having different functions

# MERITS OF FLOWCHART

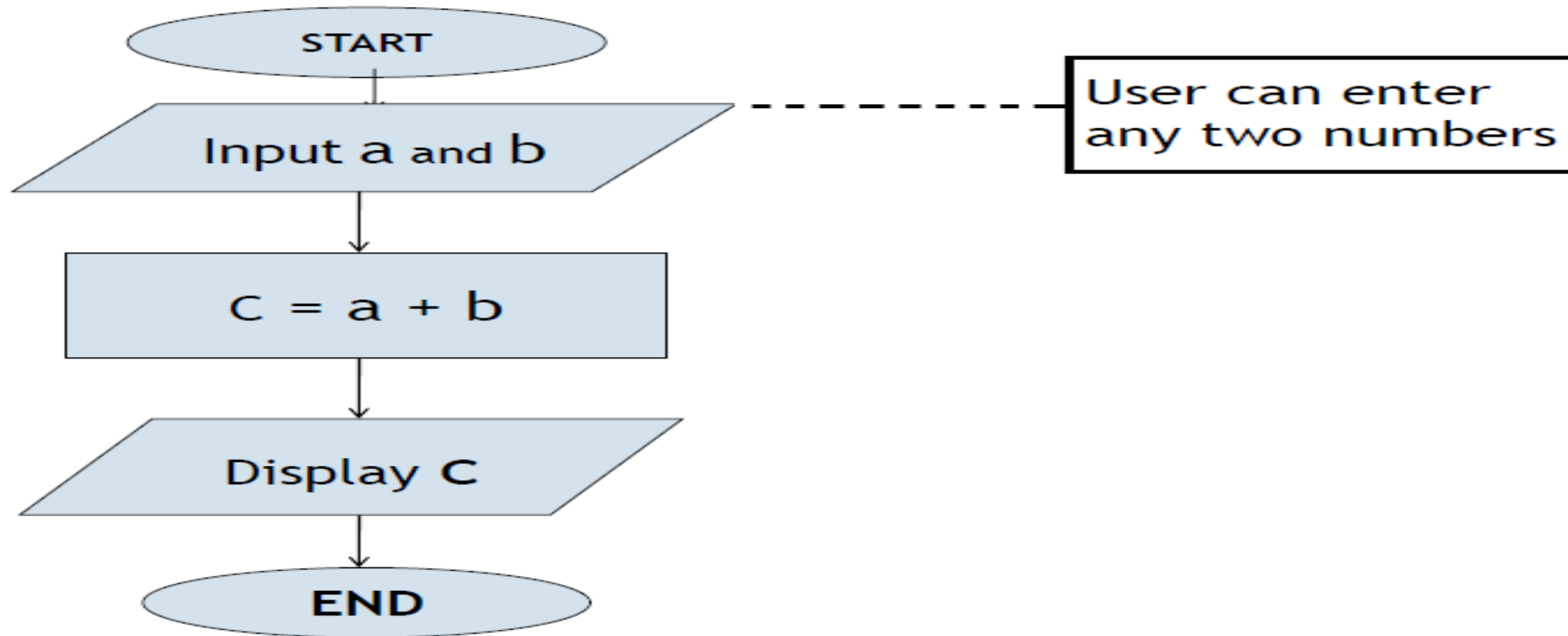
- Easy to explain program logic
- Makes coding effective and faster
- Effective communication
- Different symbols used
- Serve as documentation
- Easy to detect, locate and remove bugs in a program

# FLOWCHART SYMBOLS

Symbol	Function	Description
	Start/End	Start and end point
	Input/output	Input and output operations
	Processing operation	Editing and calculation of data
	Decision	Check logical condition
	Connector	Indicates logical flow from one page to another
	Direction of logic	Direction of flow of logic
	Comment	Indicates any comments for explanatory notes

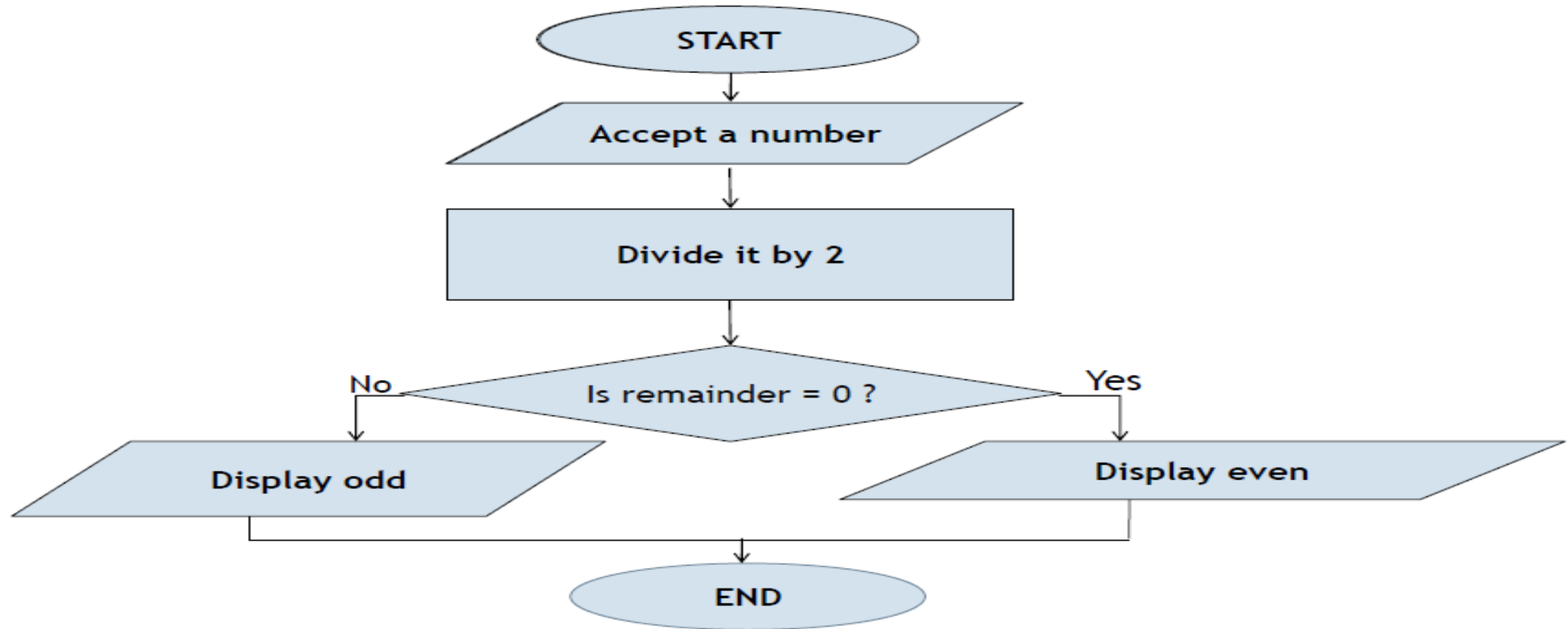
# FLOWCHART (1/2)

- **Example 1:** Flowchart sum of two numbers



# FLOWCHART (2/2)

- **Example 2:** Flowchart to check whether the input number is even or odd



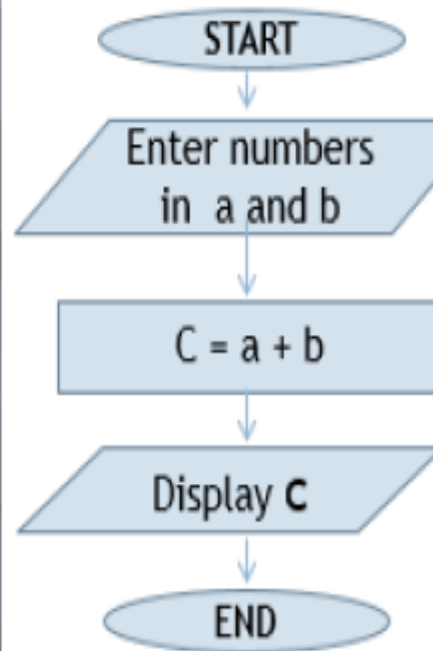


# USING ALGORITHM AND FLOWCHART IN CODING

## Algorithm

Step 1: Start  
Step 2: Enter numbers in a and b  
Step 3: Add a and b and store in c  
Step 4: Display c  
Step 5: End

## Flowchart



```
public static void main(String[] args) {
```

```
    int x, y, z;
```

```
    System.out.println("Enter two integers to calculate their sum");
```

```
    Scanner in = new Scanner(System.in);
```

```
    x = in.nextInt();
```

```
    y = in.nextInt();
```

```
    z = x + y;
```

```
    System.out.println("Sum of the integers = " + z);
```

```
} }
```

run:

Enter two integers to calculate their sum

1

1

Sum of the integers = 2

BUILD SUCCESSFUL (total time: 14 seconds)

# QUIZ

Write algorithm and flowchart to log in to your Facebook account

# PSEUDOCODE (1/3)

- *Pseudo code* is an artificial and informal language that helps programmers develop algorithms. Pseudo code is very similar to everyday English.

# PSEUDOCODE (2/3)

## ■ **Example 1:** pseudocode sum of two numbers

declare number1, number2, sum as variables

when the flag is clicked initialize all variables to 0

output: “enter the first number”

set number1 = user answer

ask user: “enter the second number:”

set number2 = user answer

set sum = number1 + number2

output: “the sum is:”

output: sum

# PSEUDOCODE (3/3)

- **Example 2:** pseudocode to check whether the input number is even or odd

H.W

# THANK YOU.....

**DO YOU HAVE ANY QUESTIONS ?**

