

# Elementary Programming Principles

## Lesson#3 : Developing Algorithm

Kalema Golooba Ayub  
agkalema@gmail.com

077 259 2826

070 169 2724

[www.kalsworld.com](http://www.kalsworld.com)

[www.sustainablecs.org](http://www.sustainablecs.org)

1

# Algorithm

---

- ❑ Refers to a limited number of logical steps that a program follows in order to solve a problem
- ❑ Pseudo code – Refers to a set of statements written in a readable language (English – like) but expressing the processing logic of program.

# Guidelines for designing a good pseudocode

---

1. The statements must be short, clear and readable
2. Pseudocode lines should be clearly outlined and indented clearly.
3. It should show clearly the start and stop of executable statements and control structures
4. Statements must not have more than one meaning
5. The input, output and processing statements should be clearly stated, using keywords such as PRINT, READ, INPUT

# Key Words

---

- ❑ Input: READ, INPUT, OBTAIN, GET
  - ❑ Output: PRINT, DISPLAY, SHOW, WRITE
  - ❑ Compute: COMPUTE, CALCULATE, DETERMINE, SUM, AVERAGE
  - ❑ Initialize: SET, INIT
  - ❑ Add one: INCREMENT
  - ❑ Reduce by1: DECREMENT
- 



## Example 1a

### Pseudocode

Develop an algorithm for a program that can output the phrase "I am Cool"

```
START  
    PRINT "I am Cool"  
STOP
```

## Example 1b Pseudocode

Develop an algorithm for a program that can prompt the user to enter his/her and then output the phrase

"Welcome **name**"

START

PRINT "Please Enter Your Name"

READ Name

PRINT " Welcome Name"

STOP

## Example 1c Pseudocode

Develop an algorithm for a program that can prompt the user to enter his/her name and then output the following phrases

"Welcome **name**"

"You are cool"

on two different lines

START

PRINT "Please Enter Your Name"

READ Name

PRINT " Welcome Name"

PRINT "You are cool"

STOP

# Express Algorithm with Flow Charts

---



# Flow Chart

- A flowchart is a diagrammatic representation of a program's algorithm
- Using both statements and special symbols that have specific meaning
- The symbols are combined with short text clues which are a form of shorthand understood by programmers

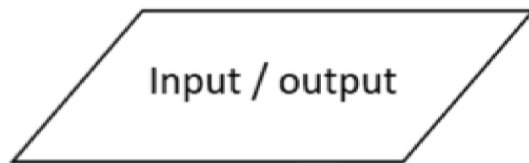
1.



***Ellipse:***

Denotes the beginning and end of the program algorithm.

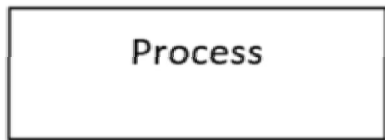
2.



***Parallelogram:***

Used to denote an input or output operation. For example, READ A, B, PRINT SUM.

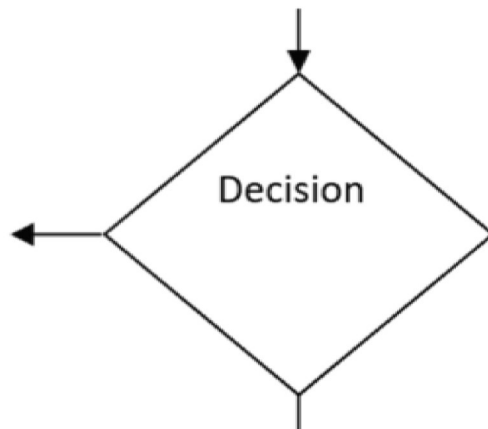
3.



***Rectangle:***

Indicates that a processing or data transformation is taking place. For example SUM = A + B.

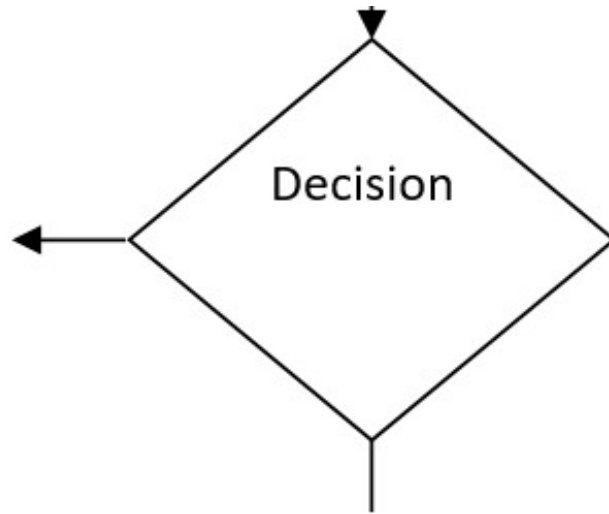
4.



***Rhombus:***

Used to specify a condition. A condition must evaluate to a *Boolean value* (True or false) for the program to execute the next instructions.

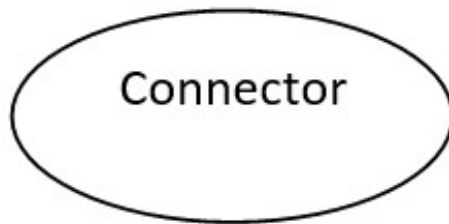
4.



**Rhombus:**

Used to specify a condition. A condition must return a *Boolean value* (True or false) for the program to proceed to the next instructions.

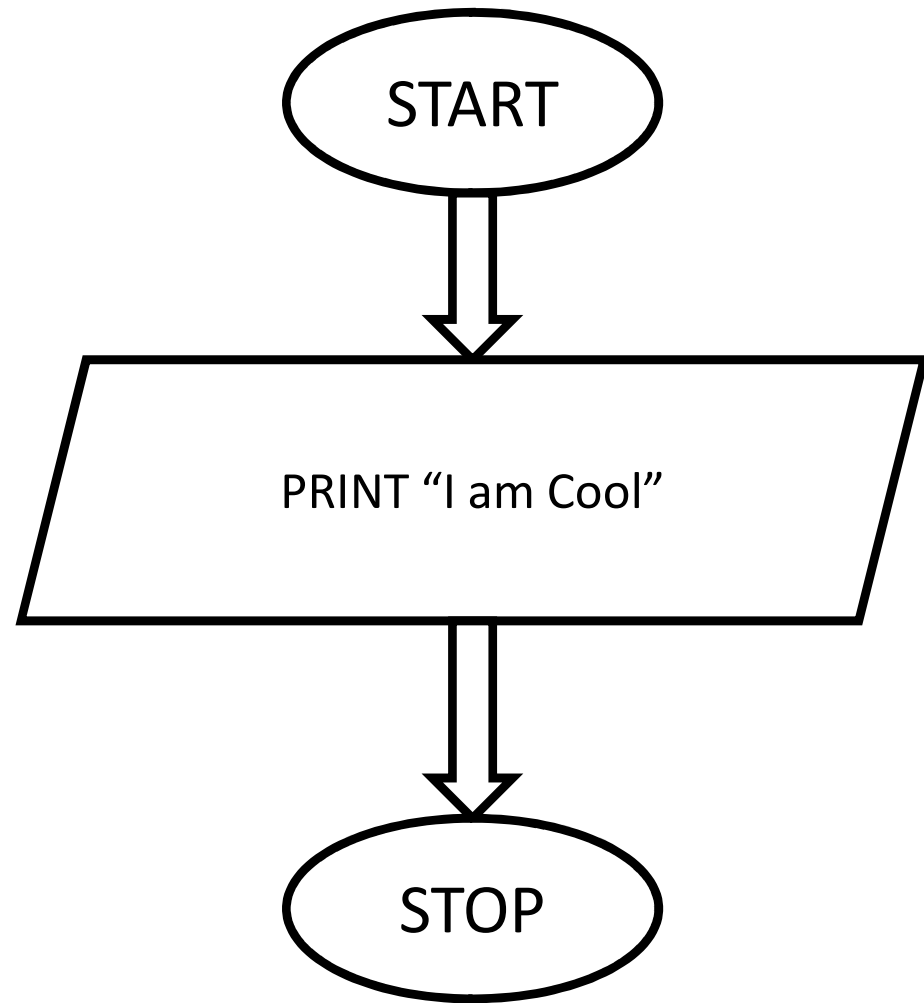
5.

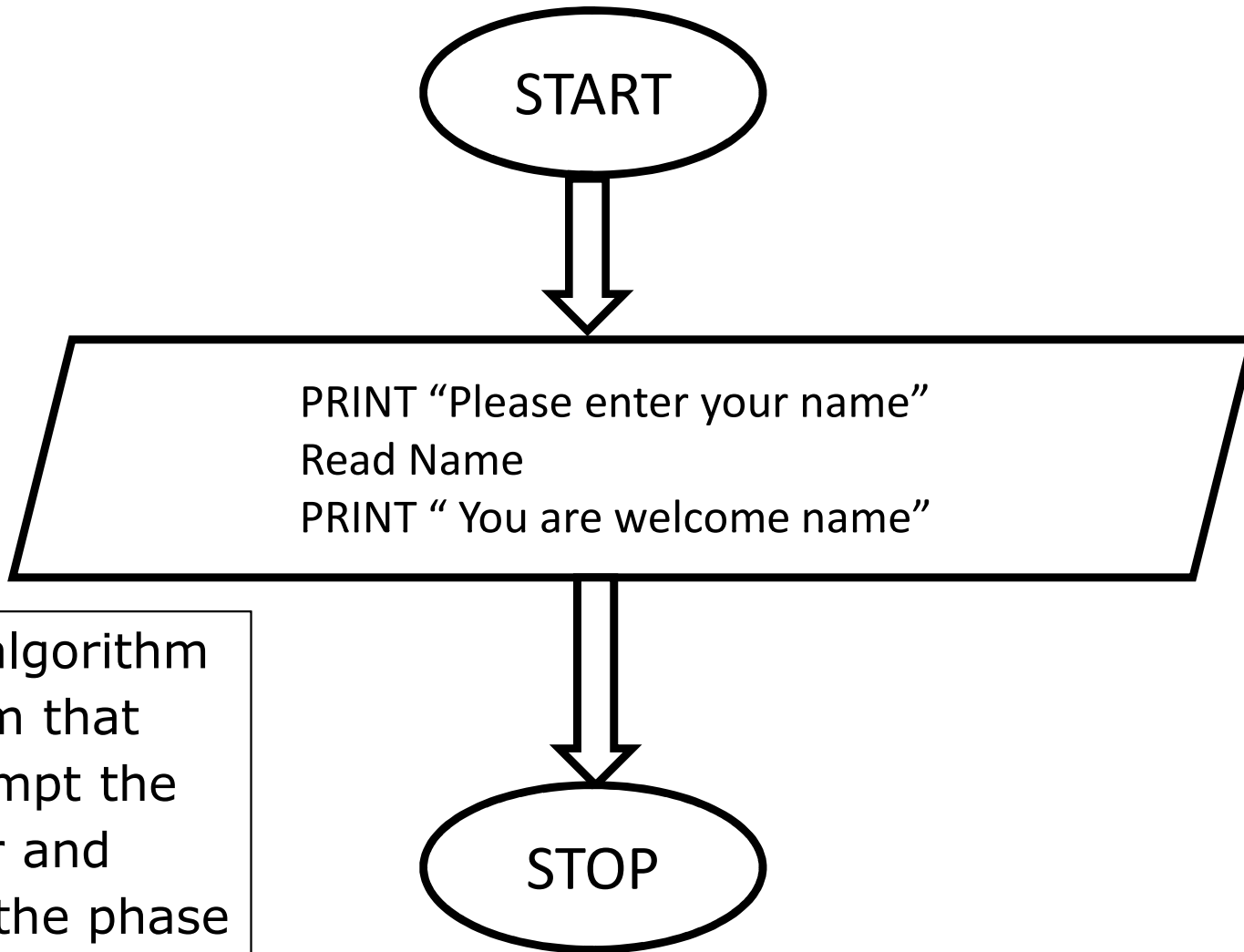


**Connector:**

Used as a connecting point or interface for flow from different directions.

Design a flow chart for a program that can output the phrase "I am Cool"





Develop an algorithm for a program that can prompt the user to enter his/her name and then output the phrase "Welcome **name**"

# Illustrations

Lets Get Dirty with Code



```
#include <stdio.h>  
int main()  
{  
    printf("You Are Cool");  
}
```

```
#include <stdio.h>  
int main()  
{  
    printf("I Love My School");  
    getch();  
    getchar();  
    return 0;  
}
```



```
#include <stdio.h>
```

```
Int main()
```

```
{
```

```
    printf("Enter Two Numbers");
```

```
}
```

```
#include <stdio.h>  
int main ()  
{  
    int x;  
    printf("Enter Two Numbers");  
    scanf("%d",&x);  
    getch();  
    return 0;  
}
```

## Exercise: PERIMETER & AREA

Write a structured algorithm that would prompt the user to enter the length and width of a rectangle, calculate the area and perimeter then display the result

**START**

**PRINT "Enter Length and Width"**

**READ L, W**

**AREA = L \* W**

**PERIMETER = 2(L + W)**

**PRINT AREA**

**Print PERIMETER**

**STOP**

## Method 2: Pseudocode

Begin

Set  $\pi$  to  $\frac{22}{7}$

Prompt the user for the radius (R).

Store the radius in a variable

(R)

Set area (A) to  $\pi \times R \times R$

Store the area in a variable (A)

Print A.

Stop

## Method 1: Flowchart

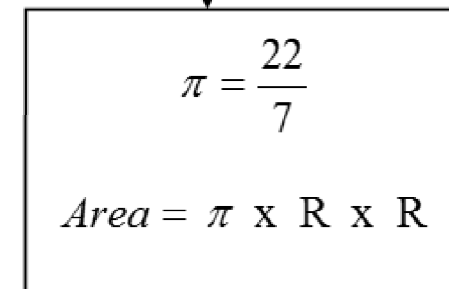
1.



2.



3.



4.



5.



# Comparison between a pseudocode and a flowchart

- Taking our example of calculating the area of a circle mentioned in the earlier subtopic the algorithm by both methods are shown below.

## Example Rolex 2

Write a pseudocode for a program that can be used to classify people according Tribe and Profession. If a person is a Musoga, output “Rolex maker”, if the person is a Muganda, output “Muyaaye” else output “Falla”.

```
START  
PRINT “Enter Your Tribe”  
INPUT Tribe  
IF Tribe =Musoga THEN  
    PRINT “Rolex Maker”  
IF Tribe =Muganda THEN  
    PRINT “Muyaaye”  
ELSE  
    PRINT “Falla”  
STOP
```

### Example 3

Write a pseudocode for a program that can be used to classify people according to the age limit. If a person is more than 35 years; output "Mature" else output "Youth".

```
START  
    PRINT "Enter the age"  
    INPUT AGE  
    IF AGE > 35 THEN  
        PRINT "Mature"  
    ELSE  
        PRINT "Youth"  
STOP
```

## Example 1 Pseudocode

Write a pseudocode that can be used to prompt the user to enter two numbers, calculate the sum and average of the two numbers and then display the output on the screen.

**START**

**PRINT “Enter two numbers”**

**INPUT X, Y**

**SUM = X + Y**

**AVERAGE = SUM/2**

**PRINT SUM**

**PRINT AVERAGE**

**STOP**



# Example

---

- Write a pseudo code that can be used to prompt the user to enter two numbers, calculate the sum and average of the two numbers and then display the output on the screen

# IF ... THEN

---

- ❑ IF ... THEN selection is used if only one option is available.
- ❑ In this case, all other options are ignored

```
IF mark > = 80 then  
    Print "Give reward"  
ENDIF
```

# IF ...THEN ... ELSE

---

Binary choice on a given Boolean condition is indicated by the use of four keywords: IF, THEN, ELSE, and ENDIF. The general form is:

IF *condition* THEN

    sequence 1

ELSE

    sequence 2

ENDIF

# IF ...THEN ... ELSE

---

IF *HoursWorked* > *NormalMax* THEN

    Display overtime message

ELSE

    Display regular time message

ENDIF

# Exercise

---

- ❑ In a football match, if a player does a mistake which is considered serious by the rules of the game, he/she is given a red card. Otherwise, he/she is given a yellow card.
- ❑ Use IF ...THEN...ELSE selection to write an algorithms (pseudocode) to illustrate the case above

# IF ...THEN ...ELSE

---

IF fault = serious THEN

    PRINT "Give Red Card"

ELSE

    Print "Give yellow card"

Print yellow card

Print red card

ENDIF

jkizito@ucc.co.ug



- Write a structured algorithm that would prompt the user to enter the length and width of a rectangle, calculate the area and perimeter then display the result.



# PROGRAM CONTROL STRUCTURES

... are blocks of statements that determine how statements are to be executed.

In structured programming languages, there are three control structures namely;

1. *Sequence*,
2. *Selection* and
3. *Iteration* (looping)

# Sequence

- In this the computer reads instructions from a program file starting from the first top line and proceeding downwards one-by-one to the end. This is called *sequenced program execution*
- It enables the computer to perform tasks that are arranged consecutively one after another in the code.

**Limitation:** Most programs that solve real world problems need to enable the computer either to repeat tasks or to make decisions when certain conditions are true or false or to repeat tasks hence the need for selection and iteration

# IF ... THEN

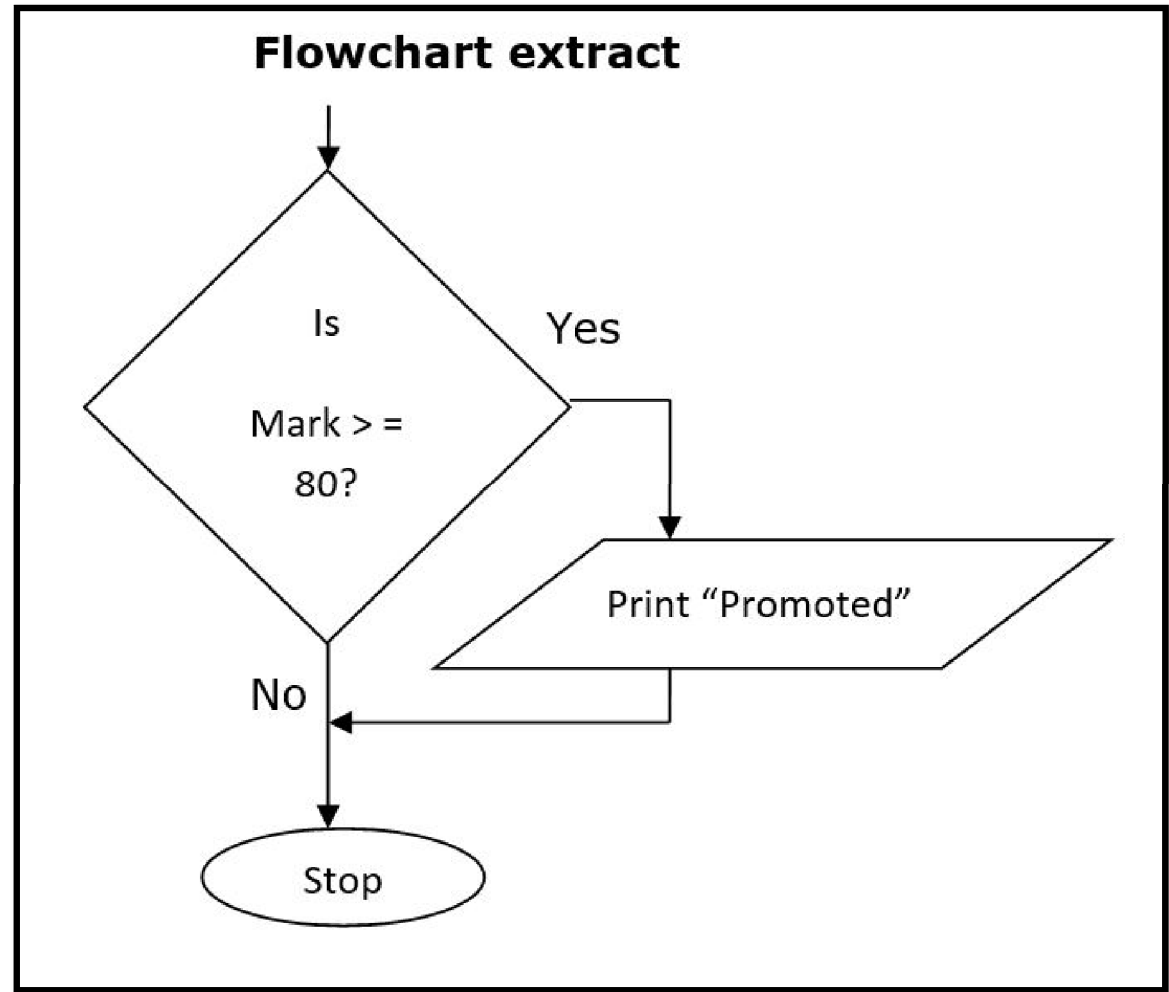
- IF ... THEN selection is used if only one option is available. In this case, all other options are ignored.
- For example, in a school environment, the administration may decide to promote only those students who attain a mean mark of 80% and above. Therefore, if a student attains 80% and above, he or she is Promoted while the rest are ignored.

### IF ... THEN:

IF ... THEN selection is used if only one option is available. In this case, all other options are ignored

### Pseudocode

```
Is Mark >= 80?  
IF mark > = 80 THEN  
    Print "Promoted"  
ENDIF
```



## Question for the Group

Design a program that will prompt a user to enter her age. If the person's age is below 11 years, it should output "Young", if she is from 11 to 19, it should output "Teenager", above 19, it should output "Adult)

## Question for the Group - Modified

Design a program that will help a user to categorise people according to age. If someone is below 11 years, it should output “Young”, If the person is from 11 to 19 years, it should output “Teenage”, above 19 years, it should output “Mature Person”

```
START
PRINT "Please enter your age"
READ Age
IF Age < 11 THEN
    PRINT "Young"
ELSE
    IF Age < = 19
IF Age > = 11
IF Age > = 11 AND < = 19
IF 11 < = ` Age < = 19
```

**START**

**PRINT "Please enter your age"**

**READ Age**

**IF Age < 11 THEN**

**PRINT "Young"**

**ELSE**

**IF Age > = 11 AND < = 19 THEN**

**PRINT "Teenager"**

**ELSE**

**PRINT "Adult"**

**ENDIF**

**ENDIF**

**STOP**



# Structured Programming

- A method for designing and coding programs in a systematic, organized manner.
- It combines the principles of top-down design, modularity and the use of the three accepted control structures of **sequence**, **repetition** and **selection**.

- Design a flowchart for a program that can be used to classify people according to the age limit. If a person is more than 20 years, output “Adult” otherwise output “Young person”.

- Draw a flowchart for a program used to prompt the user to enter two numbers. The program should find the sum and average then display the output on the screen.
- Write a pseudocode that can be used to prompt the user to enter two numbers, calculate the sum and average of the two numbers and then display the output on the screen.

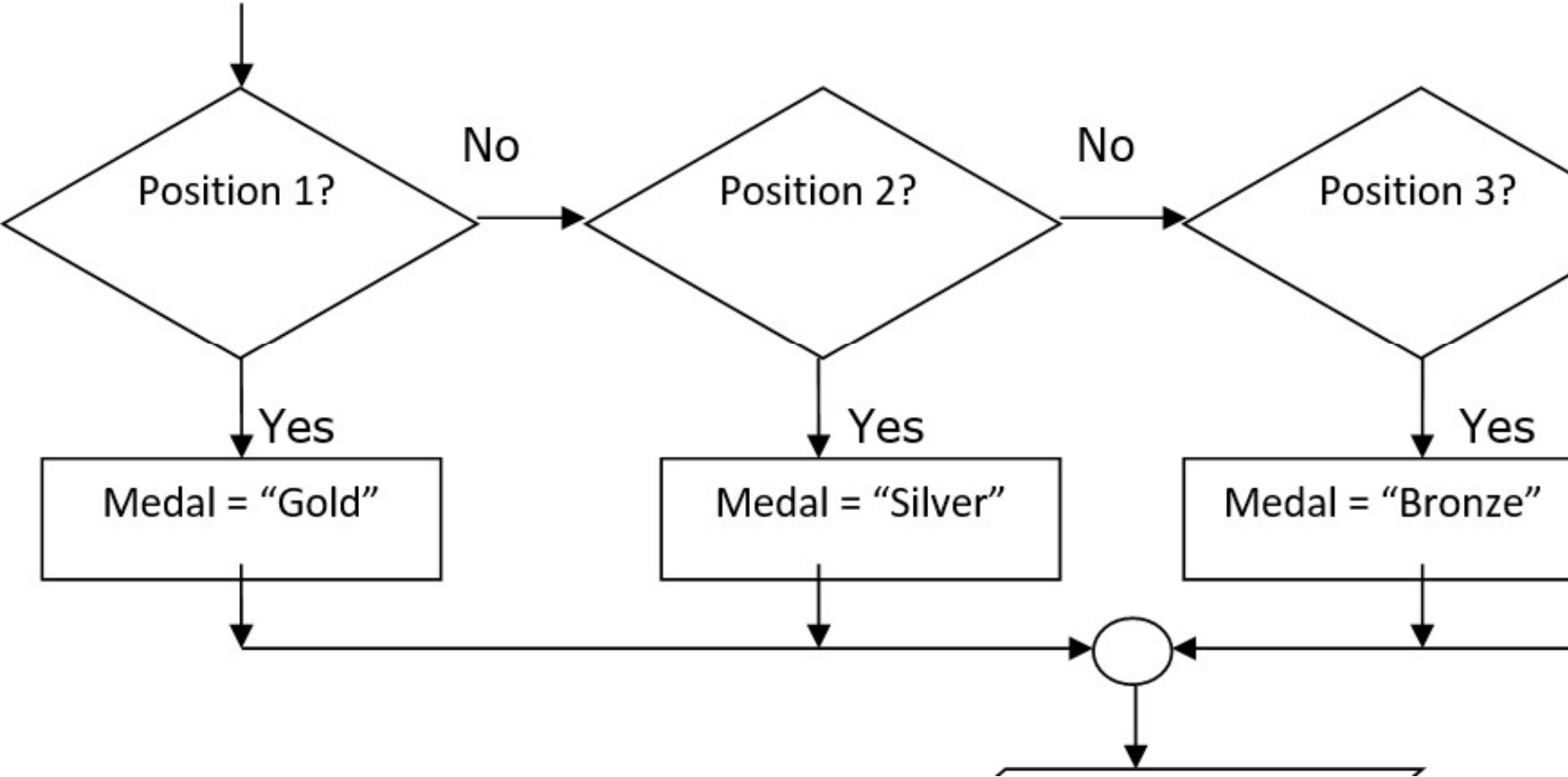
- Dev C++

# NESTED IF

- Nested IF selection is used where two or more options have to be considered to make a selection. For example, in an Olympics track event, medals are awarded only to the first three athletes as follows:
  - Position 1: Gold medal
  - Position 2: Silver medal
  - Position 3: Bronze medal

```
IF position = 1 THEN
    medal = "Gold"
ELSE
    IF position = 2 THEN
        medal = "silver"
    ELSE
        IF position = 3 THEN
            medal = "bronze"
        ELSE
            medal = "nil"
        ENDIF
    ENDIF
ENDIF
```

# Flowchart extract of a nested IF:



# WHILE

The WHILE construct is used to specify a loop with a test at the top. The beginning and ending of the loop are indicated by two keywords; WHILE and ENDWHILE. The general form is:

```
WHILE condition  
    sequence  
ENDWHILE
```



# WHILE

- The loop is entered only if the condition is **true**
- The "sequence" is performed for each iteration
- At the conclusion of each iteration, the condition is evaluated and the loop continues as long as the condition is true

# WHILE : Example

WHILE BankBalance >30,000

    Withdraw Cash

    Update Bankbalance

ENDWHILE



# CASE



# CASE

A CASE construct indicates a multiway branch based on conditions that are mutually exclusive. Four keywords, CASE, OF, OTHERS, and ENDCASE, and conditions are used to indicate the various alternatives. The general form is:

CASE expression OF

    condition 1 : sequence 1  
    condition 2 : sequence 2

    ...

    condition n : sequence n

OTHERS:

default sequence

ENDCASE

The OTHERS clause with its default sequence is optional.

Conditions are normally numbers or characters

# CASE

CASE **grade** OF

A : points = 6

B : points = 5

C : points = 4

D : points = 3

E : points = 2

O : points = 1

F : points = 0

ENDCASE

CASE

CASE Title OF

Mr : Print "Mister"

Mrs : Print "Missus"

Miss : Print "Miss"

Ms : Print "Mizz"

Dr : Print "Doctor"

ENDCASE

# REPEAT-UNTIL

- This loop is similar to the WHILE loop except that the test is performed at the bottom of the loop instead of at the top.
- Two keywords, REPEAT and UNTIL are used. The general form is:

REPEAT

    sequence

UNTIL condition

# REPEAT-UNTIL

- The "sequence" in this type of loop is always performed at least once, because the test is performed after the sequence is executed. At the conclusion of each iteration, the condition is evaluated, and the loop repeats if the condition is false.
- The loop terminates when the condition becomes true.



- The gross salary of employees in KARU ENTERPRISE is based on basic salary and additional benefit as follows
  - Employees who have worked for the company for more than 10 years receive an additional pay of 10% to their basic salary
  - monthly Salary bonus on monthly sales of books as follows:

<b>Monthly Sales</b>	<b>Bonus Rate (%)</b>
<b>Above 500,000</b>	15
<b>Between 250,000 and 500,000</b>	10
<b>Below 250,000</b>	5

Draw a flow chart for a program that would be used to calculate the gross salary then output each employee's basic salary, gross salary and all benefits

# Control Structures

- **Sequence** –in sequential order.
  - The simplest of control structures – start at the beginning and continue in sequential order.
- **Selection** – selectively execute statements
  - Called a **branch**, it requires a condition to determine when to execute statements.

# Control Structures

- ***Repetition*** – repeat statements more than once
  - Called a **loop**, it needs a stop condition, i.e, the program will continue to loop until some condition is met.

# Event-Driven Programming

- In an **event-driven** program, the flow of control is based on the user's clicking on menus and buttons, etc. These user actions are called **events**.
- Event-driven programming still uses the basic principles of structured programming – program modules, control structures, good programming style, and program testing.



For Code

W`[www.tutorialspoint.com/](http://www.tutorialspoint.com/)

[file:///C:/My%20Web%20Sites/C-Programming/www.tutorialspoint.com/cprogramming/c\\_variables.html](file:///C:/My%20Web%20Sites/C-Programming/www.tutorialspoint.com/cprogramming/c_variables.html)

httrack



# Selection : IF...THEN...ELSE

- IF ... THEN ... ELSE selection is suitable when there are two available options.

***IF < condition > THEN***

***statements;***

***ELSE***

***Statements;***

***ENDIF***

- For example, in a football match, if a player does a mistake which is considered serious by the rules of the game, he/she is given a red card. Otherwise, he/she is given a yellow card.

## Pseudocode segment

IF fault = serious THEN

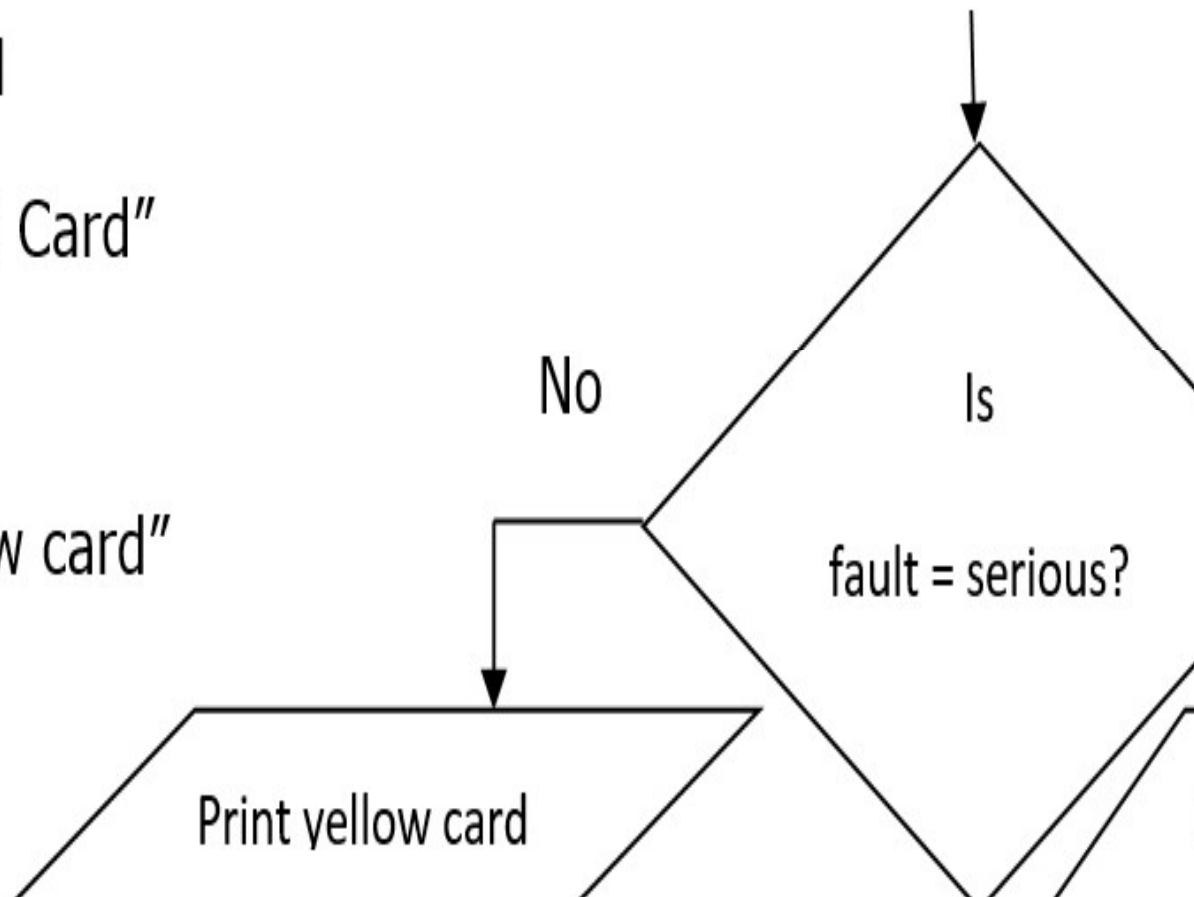
    PRINT "Give Red Card"

ELSE

    Print "Give yellow card"

ENDIF

## Flowchart extract







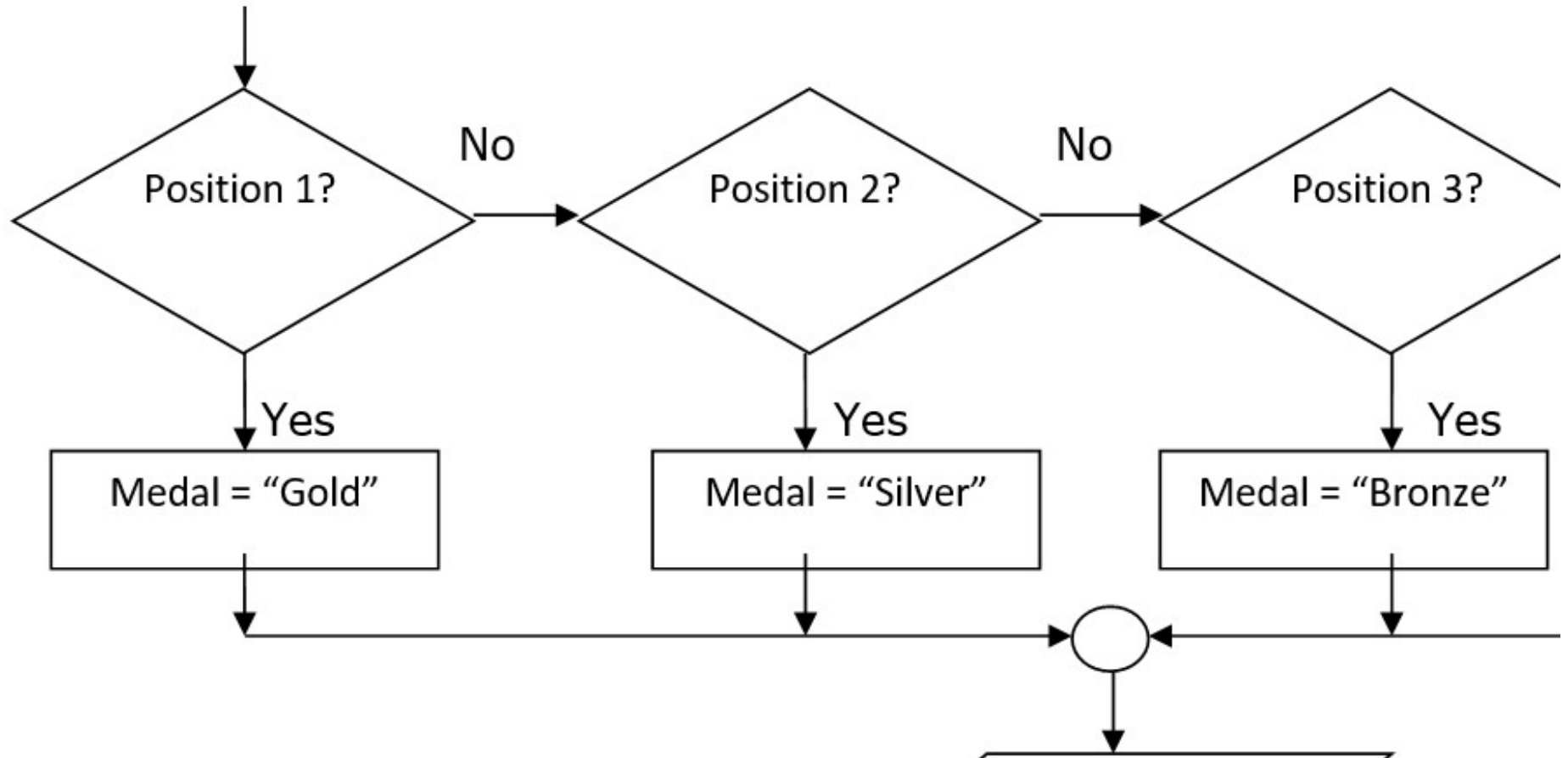
# Nested IF Selection

- Nested IF selection is used where two or more options have to be considered to make a selection. For example, in an Olympics track event, medals are awarded only to the first three athletes as follows:
  - Position 1: Gold medal
  - Position 2: Silver medal
  - Position 3: Bronze medal
- The pseudocode segment and flowchart extract below shows the structure of the Nested IF selection.

## Pseudocode Segment

```
IF position = 1 THEN
    medal = "Gold"
ELSE
    IF position = 2 THEN
        medal = "silver"
    ELSE
        IF position = 3 THEN
            medal = "bronze"
        ELSE
            medal = "nil"
        ENDIF
    ENDIF
ENDIF
```

## Flowchart extract of a nested IF:



- 
- Start
  - Enter "I Love Superman"
  - Read "I love suprt"
  - Print (" Love superman")
  - Stop

- 
- Start
  - Print "Enter Your Name"
  - Read Name
  - Print "Welcome Name"
  - Stop



## (a) **Reasons for using a flowchart other than a pseudo code for solving a problem**

---

- ❑ A flowchart is easy to interpret and understand
- ❑ A flowchart provides a better/easier understanding of the problem processing logic  
Flowcharts provide more detail yet readable structure of analyzing a problem.
- ❑ Are more capable of showing the overflow of instructions or data from one process to another.

- 
- ❑ A flowchart provides an easier way of error identification and rectification. They offer/give more efficient program maintenance as they give the programmer which part of the program logic to put emphasis on and can be edited to suite new changes.
  - ❑ With flowcharts information needs or problems are analyzed in a more effective way that reduces costs and time wastage
-



- 
- ❑ One can easily conceptualize the whole program at just a glance from a flowchart.
  - ❑ Makes results look attractive and organized

# ITAU -

---

- ❑ ITAU has organised a 2 day CDW for teachers, and want to develop a computerised program to help them bill the participants. Participants may be residential or non residential, and have a choice to attend one or both days.
- ❑ If the participant attends 2 days and is non residential, he pays 50,000/-. If the participant attends only one day and is residential, he pays Shs.