

Elementary Programming

TIBAKANYA JOSEPH

Computer programming

- A computer works by executing a set of instructions known as a **program**.
- The term **programming** refers to the process of developing computer instructions (programs) to solve a particular task.
- The process involves use **special characters, signs** and **symbols** in a programming language. Each programming language has special spelling and grammar called **syntax**.
- **A programmer** is a person who develops computer programs

Importance of programming

- Programming meets the increasing demand for computer programs.
- It provide instructions to a computer to do a specific task
- Programming provides a better understanding of how computers work.
- It helps develop thinking skills.
- It develops logical way of doing things.
Computer programming is a lucrative job.
- It is rewarding to see your ideas come out to life as a program.

Limitations to programming

- It is hard to memorize syntax of programming languages
- Limited knowledge of a programming language
- Some programming languages are hard to learn and use
- The difficulty to choose an appropriate programming language to use.
- Some programming language are not compatible with some operating systems

FEATURES OF A GOOD COMPUTER PROGRAM

- It should be able to run on multiple operating systems (portability)
- It should meet the requirements of the user
- It should be easy to learn and use
- It should be able to perform more than one task
- It is accompanied with clear and complete documentation

Programming Languages

- A **Programming language** is the vocabulary and set of grammatical rules used to write computer instructions.
- Each programming language has special spelling and grammar called **syntax** as well as contains keywords (words that it understands) and a special sequence for organizing program instructions.

COMMON EXAMPLES OF PROGRAMMING LANGUAGES

- **Java**
- **PHP**
- **Python**
- **C**
- **C++**
- **Visual basic**
- **JavaScript**
- **Ruby**
- **SQL**
- **C#**

Levels of programming languages

Many programming languages have been developed over the years. These languages are classified into two major levels namely:

Low-level languages

High-level languages

These levels are further subdivided into five generations. The first and second generations consist of low-level languages while the third to the fifth generation consist of high-level languages

Levels of programming languages

- Low-level languages are called as low because they can be easily understood by the computer directly or they require little effort to translate into computer understandable form. They are **machine code** and **assembly languages**
- High level languages are very close to the human language (English-like) and they can be read and understood even by people who are not experts in programming. High level languages are (3GL), (4GL), (5GL), Object Oriented Programming(OOP), Scripting Languages

ADVANTAGES OF LOW PROGRAMMING LANGUAGES

- Very fast to execute since they are already in a form that a computer can understand
- Requires less system resources in form of memory and disk space
- There is no need for language translators for machine languages
- Low languages are useful especially when writing programs that require a lot of accuracy

DISADVANTAGES OF LOW LEVEL LANGUAGES

- Programmers need manuals in order to interpret the codes
- It is easy to make mistakes in a sequence of writing the codes in 0s and 1s.
- It is difficult to identify a mistake in codes of 0s and 1s.
- Its tiresome to write codes while reading manuals.
- Low level languages are machine dependent, one written for apple machines cant work for IBM machines.

ADAVANTAGES OF HIGH LEVEL LANGUAGES

- High-level language programs are easy to debug
- They are machine independent. Provide programs that can be used on more than one computer.
- They are user friendly and easy to learn because they are near to natural language.
- They are flexible hence they enhance the creativity of the programmer, increasing productivity
- Allows the programmer to focus on understanding the user's needs and design the required software.
- They permit faster development of large programs.

DISADVANTAGES OF HIGH LEVEL LANGUAGES

- High level languages are slow to execute compared to low languages
- High level languages need translators before execution.
- They require more system resources compared to low level languages.

COMMON TERMS USED IN PROGRAMMING

- **Computer program** is a set of logical instructions that a computer goes through to perform a task.
They are written by programmers.
- **Program Syntax** is the spelling and grammar of a programming language.
- **Keywords** refer to words that make meaning to a programming language. **Common reserved words include: do, for, if, while, auto, case, etc**
- **Semantics** describes the processes a computer follows when executing a program in that specific language.
- **A function** is a section of a program that performs a task to return a value.

Source code is a Program instruction written as text file that must be translated into an object code before execution.

Program execution is when a computer performs an instruction of a computer program.

Object code is a program code in machine language that is ready for execution by the computer.

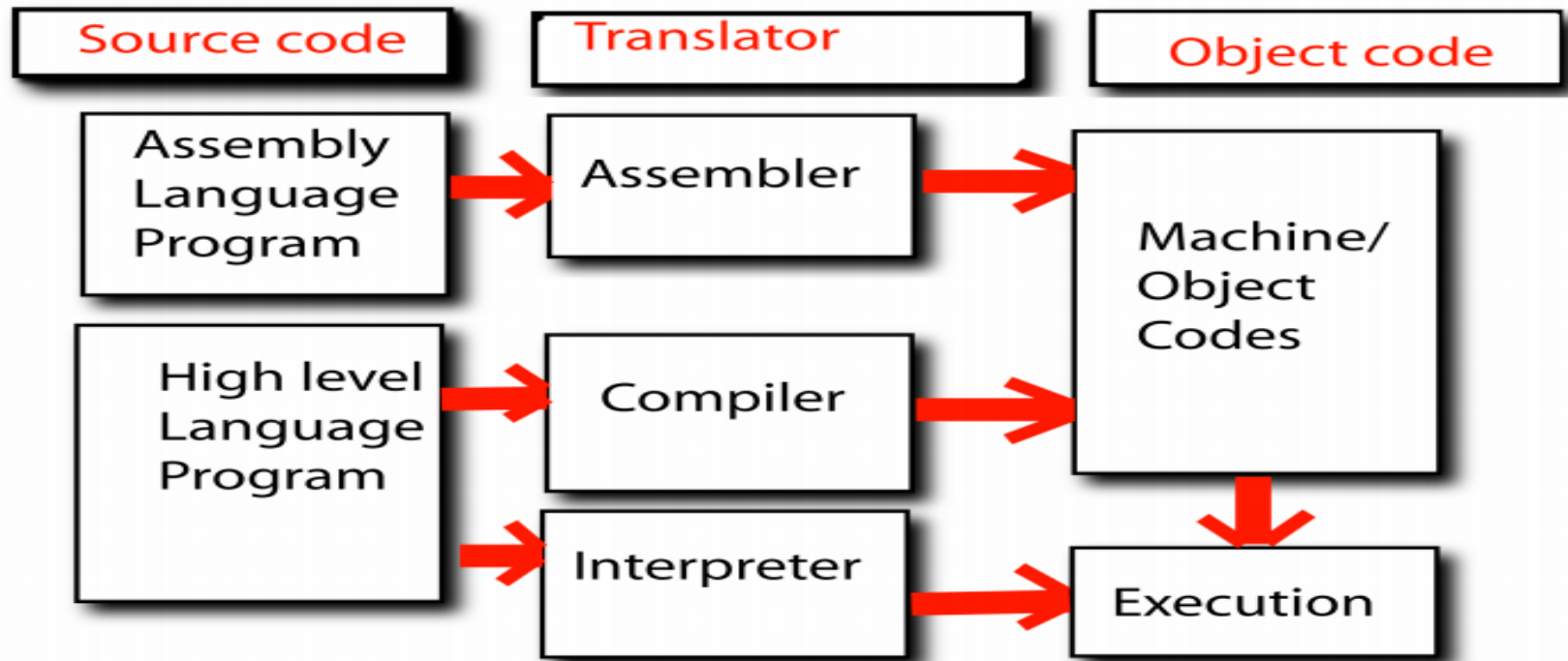
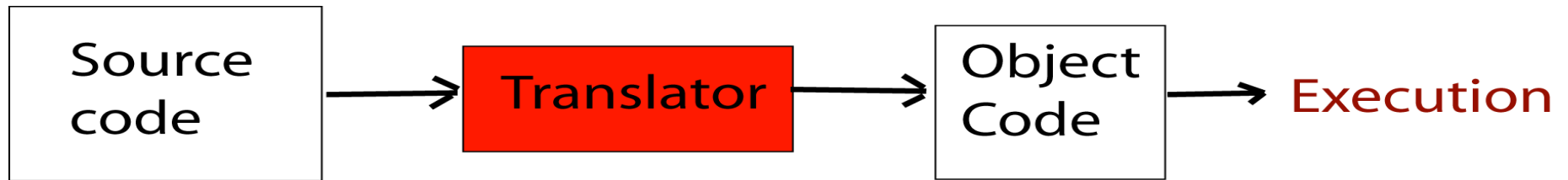
Object-oriented programming (OOP) is programming model organized around objects and data. The programmer packages the data and

the program procedures into a single unit called **an object**.

A variable is a memory for data in a computer program.

Comment is an unexcitable text included in a program code to provide additional useful information about lines of code `/* */`

Programs must be translated into Object codes before execution using **assemblers, compilers, interpreters**



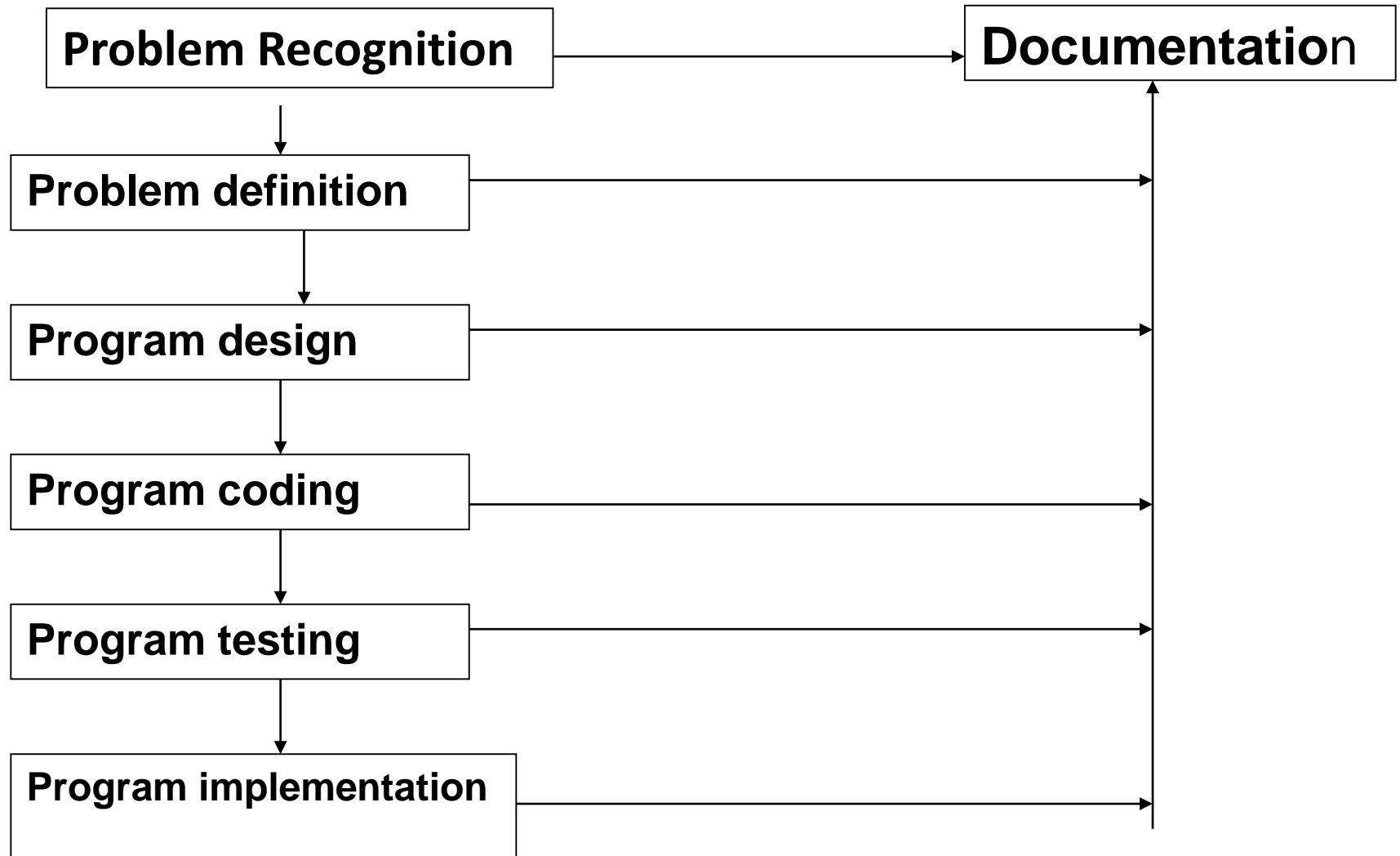
Exercise

- Define the term programming?
- Why is programming important?
- Explain five limiting factors to programming
- What is the difference between a computer program and a programming language
- State any four advantages of high level languages over low level languages
- Write short notes on the following; comment, variable, object code, source code, function, keyword, executable file.
- What is meant by OOP?

PROGRAM DEVELOPMENT LIFE CYCLE (PDLC)

These are stages undertaken in a process of developing computer programs. They include;

- 1. problem recognition**
- 2. Problem definition**
- 3. Program design**
- 4. Program coding**
- 5. Program testing**
- 6. Program installation and maintenance**
- 7. Documentation**



1.Problem Recognition. This is understanding and interpretation of a problem. Programming is a problem solving exercise, think of some who saw the tiresome job of creating documents using a typewriter and developed a word processor. For instance consider a math problem of finding area of a rectangle ($A = L * W$)

2.Problem definition. At this stage a programmer defines the likely input, processing activities and the expected output. For instance in our problem of finding area of a rectangle; inputs (length, width), process ($A = \text{length} * \text{width}$) and output is A (area of a rectangle).

This stage ends with **writing a requirements report.**

This a document that allows a programmer to come up with a program design

3. **Program design.** This is a stage for actual development of a program's **algorithm**.

An algorithm is definite number of logical steps that a program follows in order to solve a problem.

4. **Program coding.** This is actual process of converting designs into code using a programming language. The end result is a **source code**.

5. **Program testing & debugging.** This is a stages of detecting errors (**Bugs**) and correcting them (**Debugging**).

6. Implementation and maintenance. This is actual delivery and installation of the program.

7. Program documentation. This is the writing of formal support material explaining how the program was developed, how it is used and how it can be modified. Documentation can be internal or external

Internal documentation is written in non-executable lines of source code called comments.

External documentation is reference material such as user manuals.

Documentation is for three target groups i.e

- **User oriented** documentation for user of program
- **Operator oriented** for technical user to install & config
- **Programmer oriented** for reuse of source code.

PROGRAM TESTING

The major types of errors

-Syntax Errors. These are errors as a result of improper use of language.

-Logical Errors. These are errors not detectable by translator. The program runs but gives wrong output

Testing is process of running a code with an aim of detecting an error.

Methods of testing

- 1. Dry run (desk checking)** . This is going through the program while still on paper.
- 2. Using debugging utilities.** This is the entering the program into an editor to detect syntax errors.
- 3. Using test data.** This is carrying out a trial run of new program with sample data.

SYNTAX ERRORS IN A PROGRAM CODE.

- Program grammar error
- Spelling error *for instance - writing a keyword with wrong spelling*
- Punctuation errors or poor punctuation for example
- *missing semicolon to terminate execution of line of code*
- Missing Parenthesis e.g. (}) to indicate the start and end of main function.
- Printing the value of variable without declaring it
- Using a function that is not in the included header
- Using wrong case for keywords
- Poor use of space as a character

EXERCISE

- What is meant by program development life cycle?
- Explain the stages of the program development life cycle.
- Tabulate the end products of each stage in the program development life cycle
- What is the difference between program requirements and requirements report?
- Distinguish between program coding and program design
- What are program bugs? How are they detected?
- Differentiate between syntactic and logical errors
- To whom is program documentation done?
- Distinguish between internal and external documentation

RECALL ABOUT OPERATORS

DEVELOPMENT OF ALGORITHMS

Algorithms refers to a set of finite number of logical steps that a program follows in order to solve a problem. These algorithms can be designed through the use of **Flowcharts or Pseudocodes**.

Pseudocodes

Pseudocode is a set of statements written in human readable language but expressing the processing logic of a program. It's important to note that Pseudocodes are not executable.

GUIDELINES FOR DESIGNING A GOOD PSEUDOCODES

- Statements must be short, clear and readable
- Statements shouldn't have more than one meaning. I.e should be unambiguous
- Lines should be clearly outlined & indented.
- It should show clear start and stop
- The input, process and output should be clearly stated using keywords

PSEUDOCODE COMMONLY USED KEYWORDS

Pseudo-code uses several keywords to indicate common input, output, and processing operations:

INPUT: READ, READLINE, OBTAIN, GET, ENTER, INPUT

PROCESSING: Initialize, SET, INIT Add one, INCREMENT, BUMP, COMPUTE, CALCULATE, DETERMINE

OUTPUT: PRINT, DISPLAY, SHOW, WRITE,

SAMPLE PSEUDOCODES

EXAMPLE 1

Write a Pseudocode that can be used to prompt the user to enter two numbers, calculate the sum and display the output on the screen

START

PRINT "Enter two Nos"

INPUT X,Y

SUM = $X + Y$

PRINT SUM

STOP

SAMPLE PSEUDOCODES

EXAMPLE 2

Write a Pseudocode that can be used to prompt the user to enter length and width of a rectangle, calculate the area and display the output on the screen

START

PRINT "Enter L & W"

READ L, W

AREA = L * W

PRINT AREA

STOP

SAMPLE PSEUDOCODES

EXAMPLE 3

Write a Pseudocode that can be used to classify people according to their age. If a person is more than 20 year; output “adult” else output “young person”

START

PRINT “Enter the age”

INPUT AGE

IF AGE >20 THEN

 PRINT “Adult”

ELSE



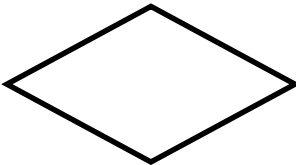

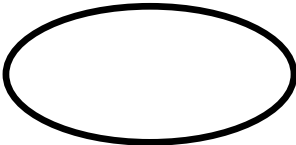
 PRINT “Young person”

STOP

PROGRAM FLOW CHARTS

- A flow chart is a diagrammatic representation of program's algorithm using standard symbols and short statements to describe various activities.
- These symbols are called **ANSI symbols** (called after the American National Standards Institute that developed them).
- A flowchart shows how a program works before one begins the actual coding of the program.

FLOW CHART SYMSBOLS

| Symbol | Name | Use |
|--|---------------------------------|---|
|  | Rectangle (PROCESS) | Denotes processes e.g. addition |
|  | Arrow (FLOW LINE) | Denotes the direction of logic flow in a program. |
|  | Diamond (DECISION) | Denotes a decision to be made. The program takes one of the routes (e.g. if/then/else). |
|  | Parallelogram (INPUT/OUTPUT) | Denotes either an input operation (e.g. input) or an output operation (e.g. print) |
|  | Oval (START/STOP) | Denotes the start or stop of the program |

FLOW CHART FOR CALCULATING AREA OF A CIRCLE

PSEUDOCODE

START

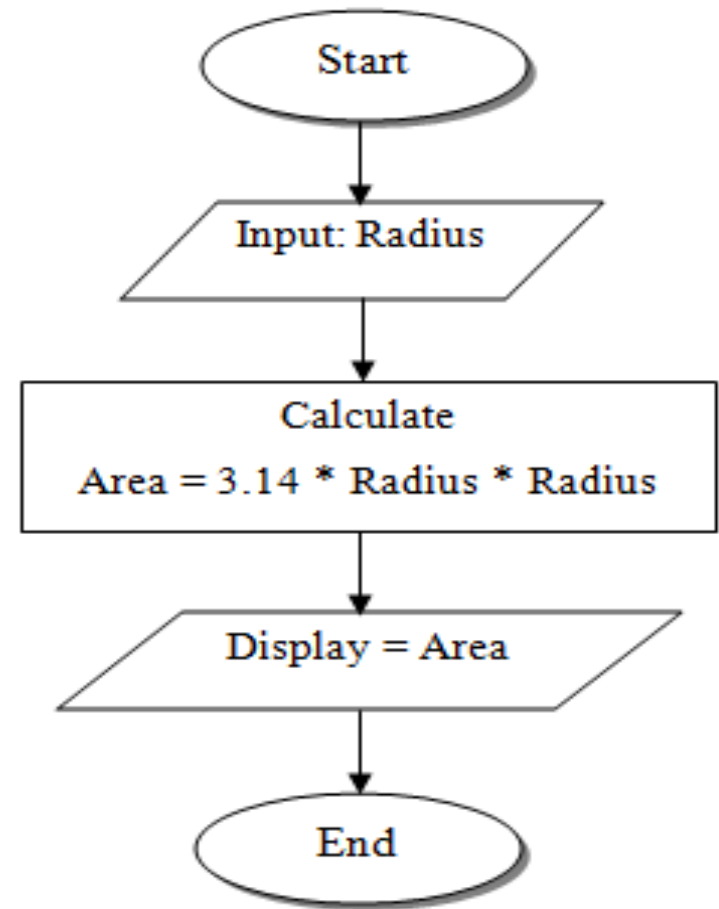
PRINT "Enter R"

READ R

$\text{AREA} = 3.14 * R * R$

PRINT AREA

STOP



FLOW CHART FOR A PROGRAM PROMPTING USER TO ENTER TWO NOS, FIND SUM AND AVG

START

PRINT "Enter X, Y"

READ X, Y

$SUM = X + Y$

$AVG = SUM/2$

PRINT SUM, AVG

STOP

FLOW CHART FOR A PROGRAM PROMPTING USER TO ENTER LENGTH & WIDTH OF RECTANGLE, FIND AREA AND PERIMETER

START

PRINT "Enter L, W"

READ L, W

$AREA = L * W$

$PERIMETER = 2(L+W)$

PRINT AREA, PERIMETER

STOP

EXERCISE

1. Design a flow chart for a program that can be used to classify people according to age limit. If a person is more than 20 year, output “Adult” otherwise output “Young person”.
2. Design an algorithm to prompt a user to enter two Nos X and Y, divide X by Y. however, if the value of Y is 0, the program should display an error “error: Division by zero”.
3. **UNEB 2018** A school uses a simple computerized system to manage students results. The school administers three types of exams namely beginning of term, mid term and end of term. The system administrator has a set the system to get total of the three exams and find average which is used to promote the students to the next class. If average is below 50 is dismissed. Write a suitable algorithm that will manage the students results

4. Design an algorithm for a program for user login into the Facebook account by providing user e-mail ID and password. The program should check whether the e-mail address and password are correct and show the account profile as well as display a message “Login error”
5. Design a program algorithm to prompt a user to enter a number. If the number is even, the program shows a message “Number is even” else display “Number is odd”

6. Afuna Adula is a business man in Kikuubo. He has a manual system to keep track of status of his stock. He determines whether he makes loss or profit on every item. If the selling price is greater than cost price, it's regarded as a profit and he calculates the profit value. Else if otherwise it's a loss he also computes the loss value. Design an algorithm for his system to tell whether he made a loss or profit as well as display its value.

7. A tax agency uses a manual system to determine tax income from a set of workers basing on the following schedule.

| Range (shs) | Tax rate |
|-------------------|-------------------|
| 0 – 250,000 | 0% |
| 250,001 – 500,000 | 15% + Shs 5,000 |
| 500,001 – 750,000 | 25% + Shs. 15,000 |
| 750,001 - above | 30% + Shs. 20,000 |

You are required to design a flow chart that will diagrammatically represent this information.

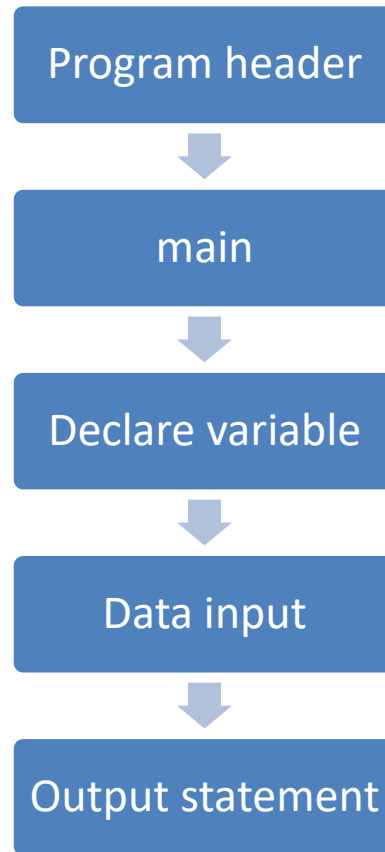
PROGRAM CODING IN C

Coding in the process of writing the program instruction in a programming language

WHY USE C?

It is Simple, Powerful, Portable, Machine independent, High efficiency, Flexible.

LOGICAL FLOW OF PROGRAM CODE



To output a string

```
#include <stdio.h>
int main()
{
    Printf("Hello world\n");
    return 0;
}
```

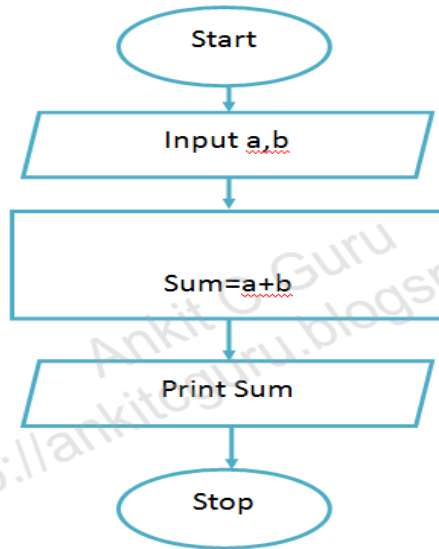
Output of above program:

"Hello World"

look at the various parts of the above program:

1. **#include <stdio.h>** is a preprocessor command, which tells a C compiler to include `stdio.h` file before going to actual compilation.
2. **int main()** is the main function where the program execution begins.
3. The next line `/*...*/` will be ignored by the compiler and it has been put to add additional comments in the program. So such lines are called comments in the program.
4. **printf(...)** is another function available in C which causes the message "Hello, World!" to be displayed on the screen.
5. **return 0;** terminates the `main()` function and returns the value 0

To add two numbers



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

```
int main()
```

```
{
```

```
int a, b, c;
```

```
printf("Enter two numbers to add\n");
```

```
scanf("%d%d",&a,&b);
```

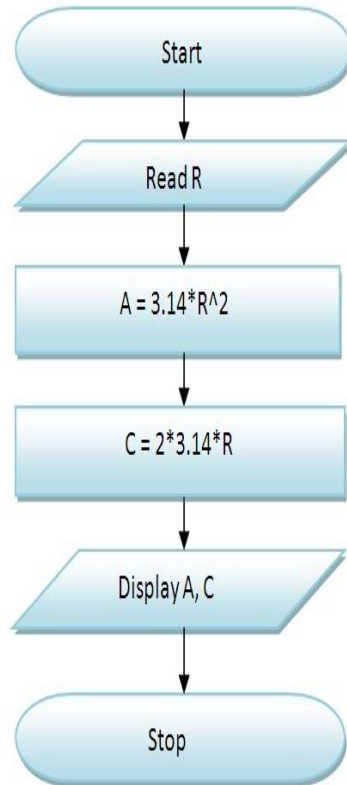
```
c = a + b;
```

```
printf("Sum of entered numbers = %d\n",c);
```

```
return 0;
```

```
}
```

To determine the area of a circle



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

```
int main()
```

```
{
```

```
    int A, R;
```

```
    printf("Enter the radius R");
```

```
    scanf("%d",&R);
```

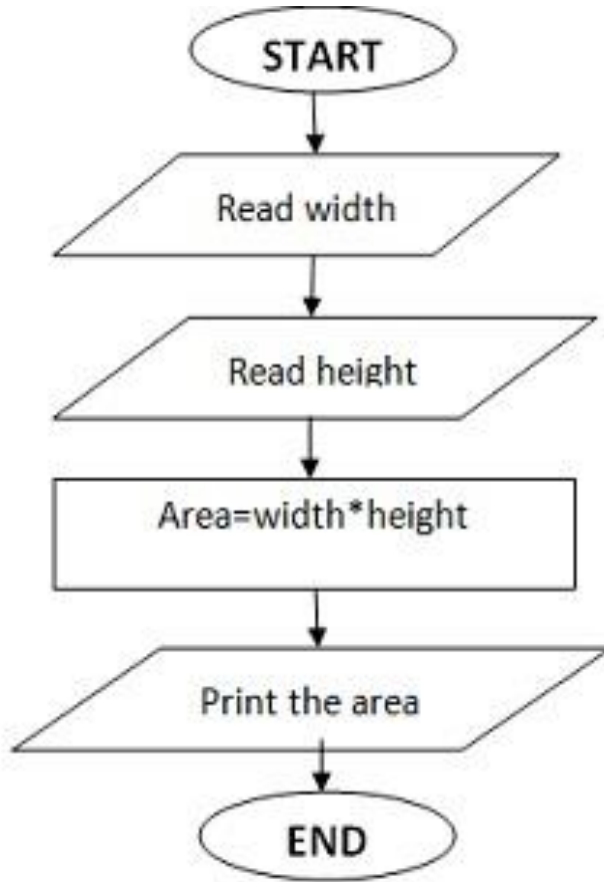
```
    A = 3.14 * R * R;
```

```
    printf("The area of the circle = %d\n",A);
```

```
    return 0;
```

```
}
```

To determine the area of a rectangle



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

```
int main()
```

```
{
```

```
    int A, w, H;
```

```
    printf("Enter the width W");
```

```
    printf("Enter the Height H");
```

```
    scanf("%d%d",&W&H);
```

```
    A = W*H;
```

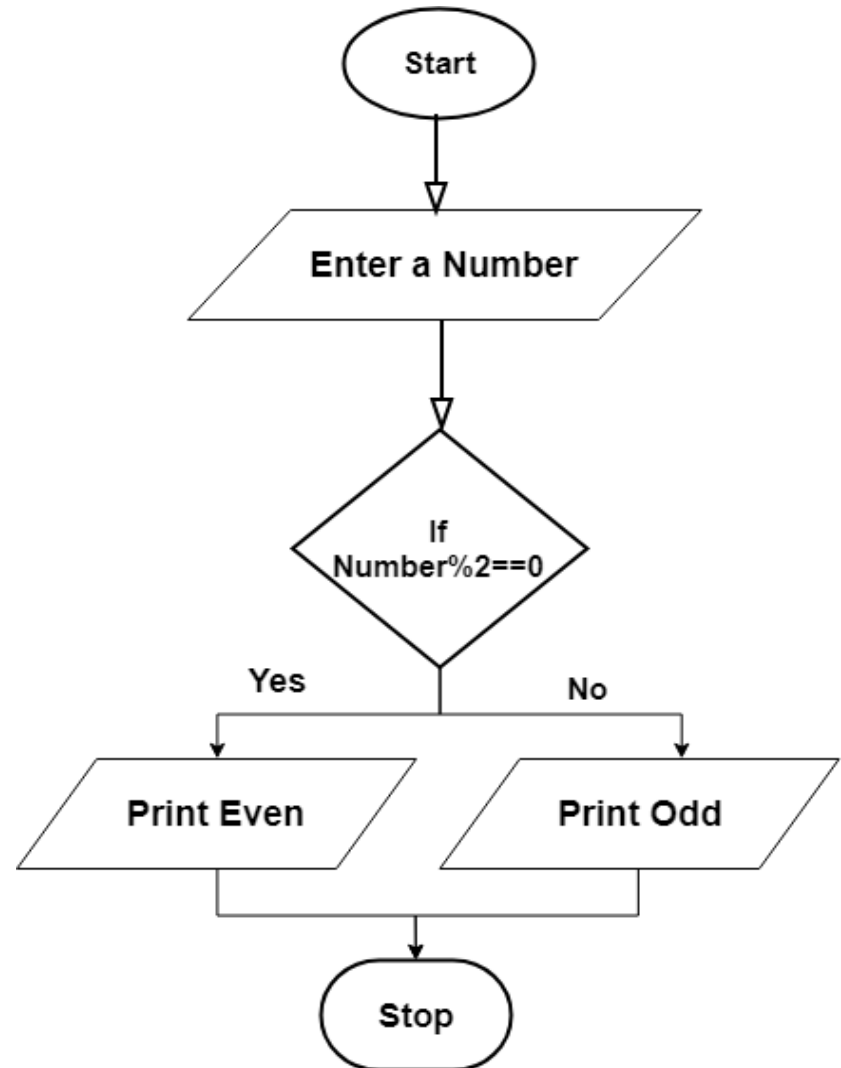
```
    printf("The area of the Rectangle = %d\n",A);
```

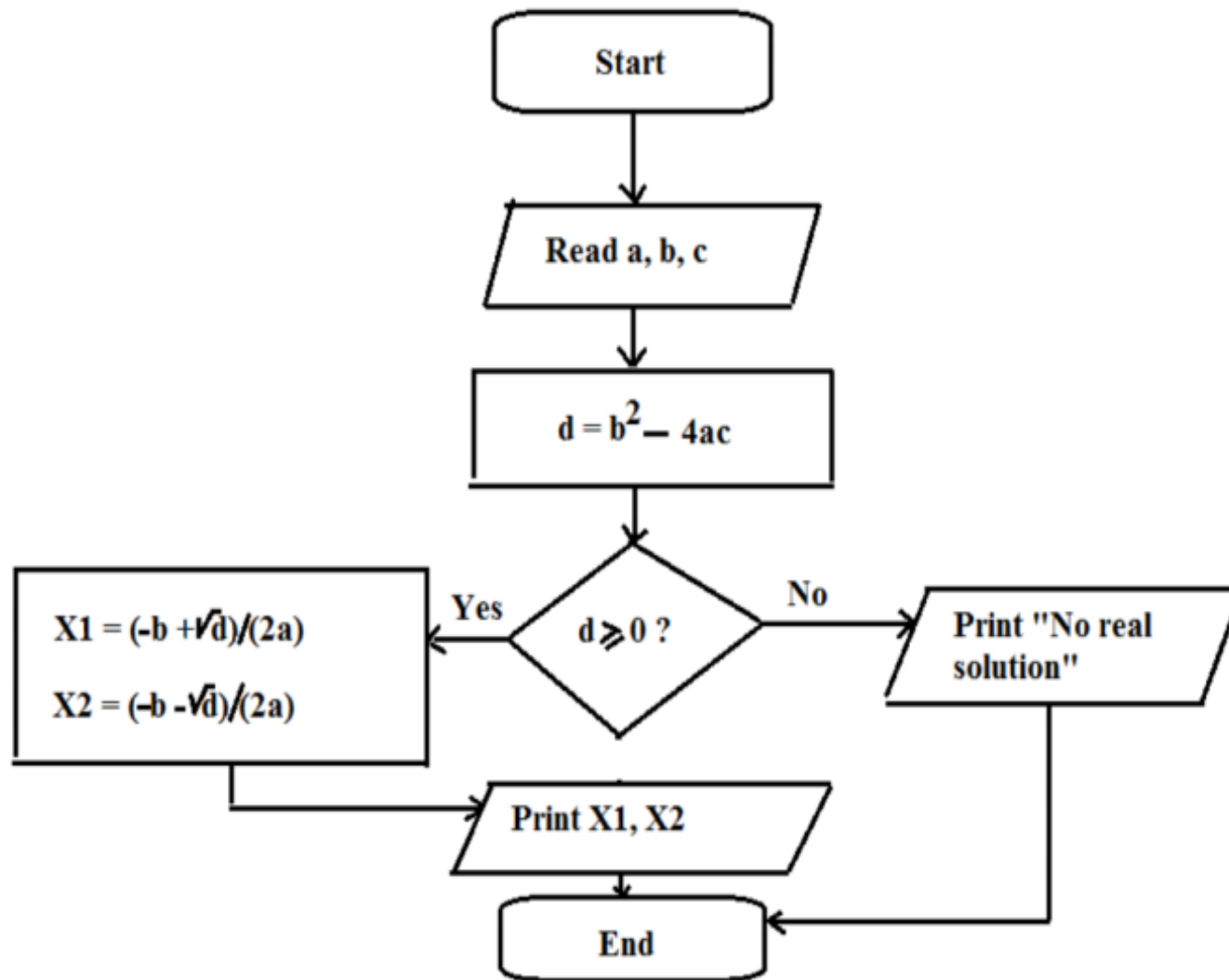
```
    return 0;
```

```
}
```

To test whether a number is odd or even

```
#include <stdio.h>
int main()
{
    int n;
    printf("Enter an integer\n");
    scanf("%d", &n);
    if (n%2 == 0)
        printf("Even\n");
    else
        printf("Odd\n");
    return 0;
}
```





Review questions

Write pseudo codes, flow charts and codes for the following programs

- **A program to return a string “ a computer is a good device”**
- **A program to find sum, product, quotient of two NOs**
- **A program to find areas of circle, cuboid, rectangle, square, rhombus**
- **A program to return first ten counting NOs**
- **A program to return ten prime numbers**
- **A program to return a determine if a NO is odd or even**
- **A program to find roots of Nos (square root, cube root, n^{th} root)**
- **A program to find roots of a quadratic equations**

NB. Write programs for all algorithms you developed in this chapter