Computer studies Form three work

Table of contents

- Data representation in a computer

- Data processing

- Elementary programming principles

- System development
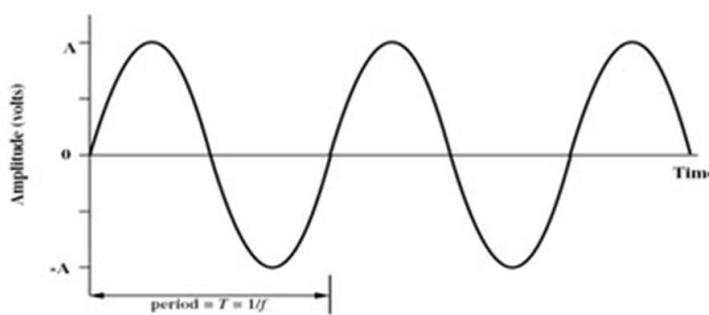
Chapter 1
Data representation in a computer

Introduction

Digital signal
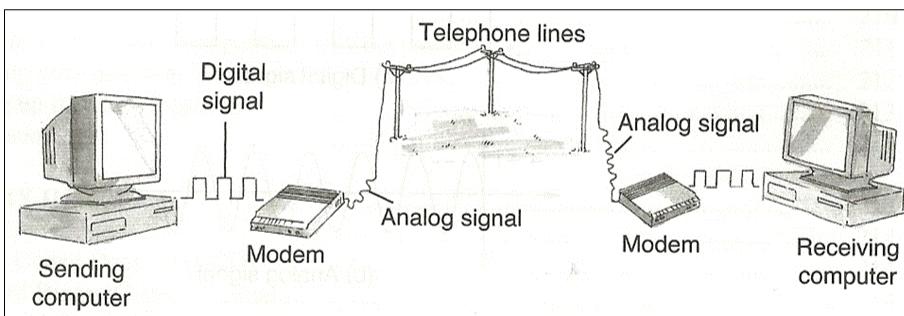
Analog signal

- In digital computers, the user inputs is converted and transmitted as electrical pulses that can be represented by two digits '1' and '0' before processing. These two digits are referred to as **Binary** digits.

Sending data via a telephone line

- When a digital signal is to be sent over analog telephone lines e.g. e-mail, it has to be converted to analog signal. This is done by connecting a device called a modem to the digital computer. This process of converting a digital signal to an analog signal is known as modulation. On the receiving end, the incoming analog signal is converted back to digital form in a process known as demodulation.
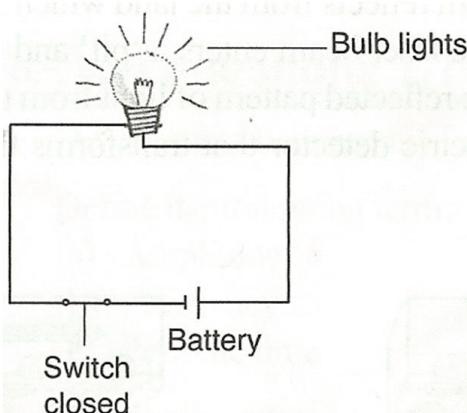
# Concepts of data representation in digital computers
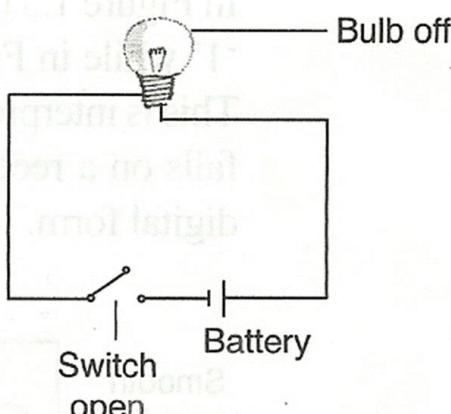
- Data and instructions cannot be entered and processed directly into computers using human language. Any type of data entered must first be converted into **machine-readable form. (Binary form)**. Computer together with it's peripheral devices handle data in it's electronic circuits, magnetic media and in optical devices.

Data representation in electronic circuits

Switch closed, bulb lights
— Bulb lights
Battery
Switch closed
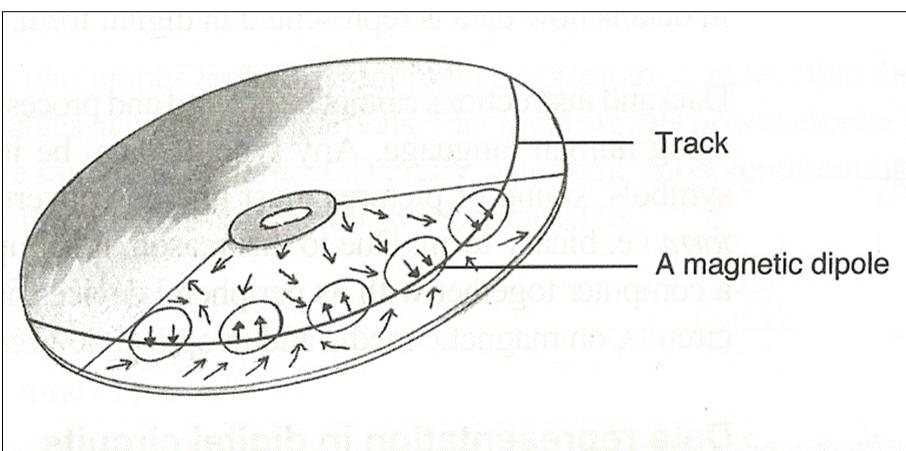
Switch open, bulb goes off
— Bulb off
Battery
Switch open

- Electronic components such as microprocessor are made up of millions of electronic circuits. Availability of a high voltage (ON) in these circuits is interpreted as '1' while a low voltage (OFF) is interpreted as '0'.

Data representation on magnetic media

Track

A magnetic dipole

- The presence of a magnetic field in one direction on magnetic media is interpreted as '1' while the absence of magnetic field is interpreted as '0'.
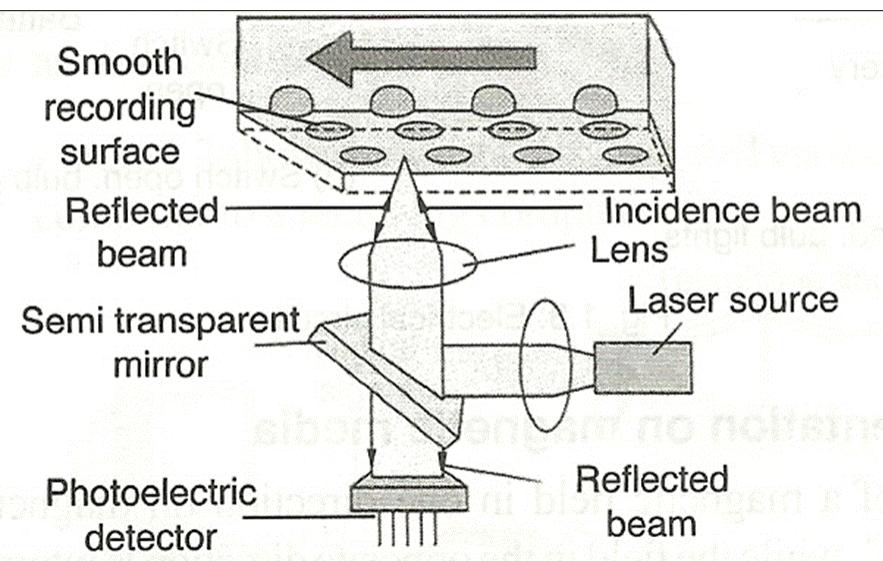
Data representation on optical media

In optical devices, the presence of light is interpreted as '1' while its absence is interpreted as '0'.
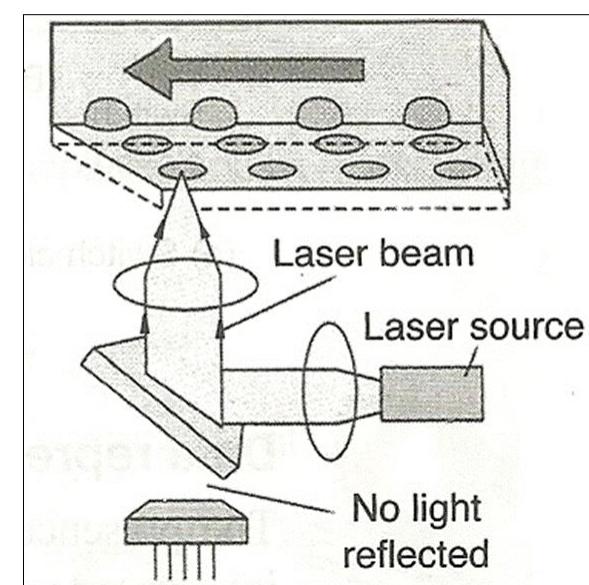
- **BITS** – this is a binary digit, which can either be 0 or 1. it's the basic unit of data or information in digital computers.
- **BYTE** – a group of 8 bits. It is the basic unit of measuring memory size in computers.
- **NIBBLE** – it is a half a byte or a group of 4 bits.
- **WORD** – its consists of one or more bytes and is the largest amount of memory that the computer can handle.
- **WORD LENGTH** – is the number of bits in each word of computer.

Reading data on a CD-ROM

Laser beam is reflected

Laser beam is not reflected

Types of data representation

- Number System

- Base conversion

- Negative Numbers

- Binary Addition
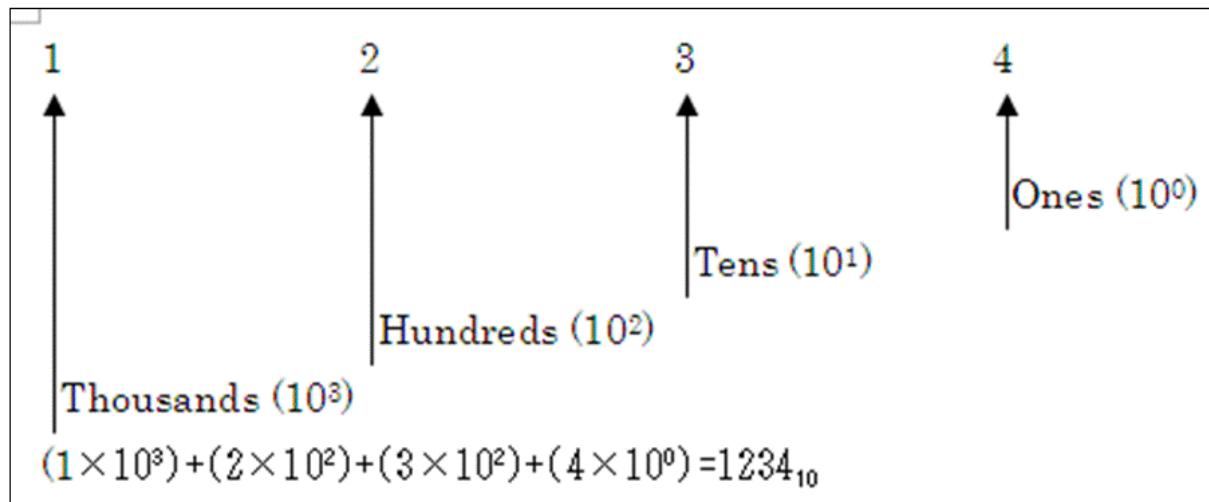
Types of number systems

- Decimal Number System

- Binary Number System

- Octal Notion

- Hexadecimal Number System
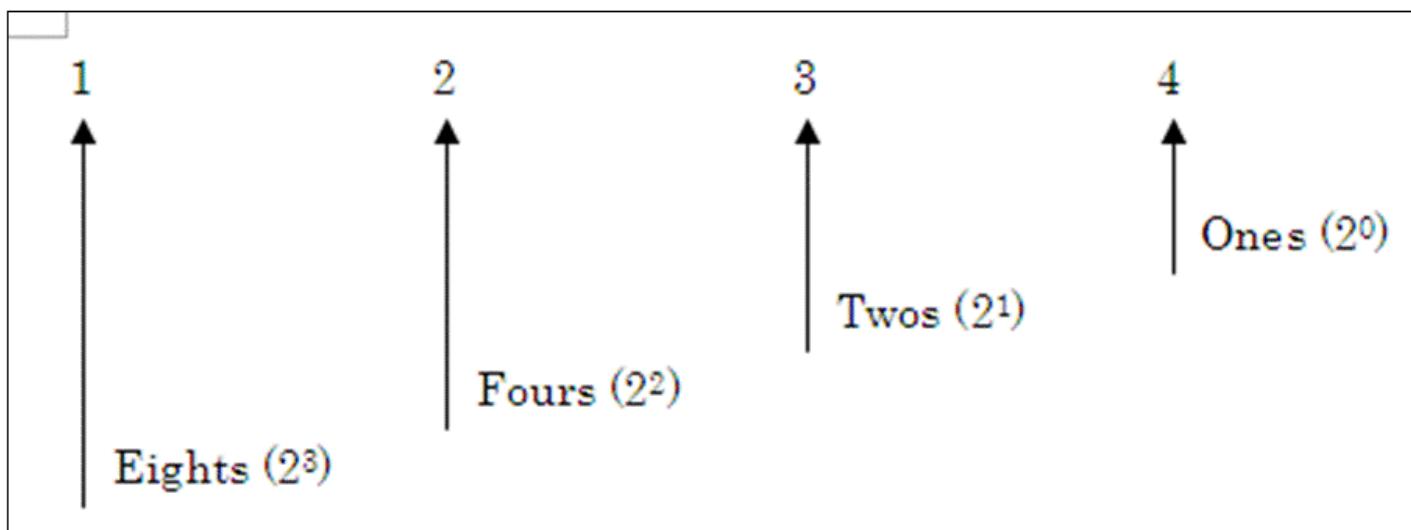
Decimal Number System

- This number system has ten digits ranging from 0-9. A decimal number should be written with a subscript 10. e.g. 132_{10}. Each digit of a given number in the base has a place value, which is influenced by base ten. E.g. number 1234_{10}

```
1               2               3               4
↑               ↑               ↑               ↑
                                                |Ones (10⁰)
                                |Tens (10¹)
                |Hundreds (10²)
|Thousands (10³)
(1×10³)+(2×10²)+(3×10²)+(4×10⁰)=1234₁₀
```
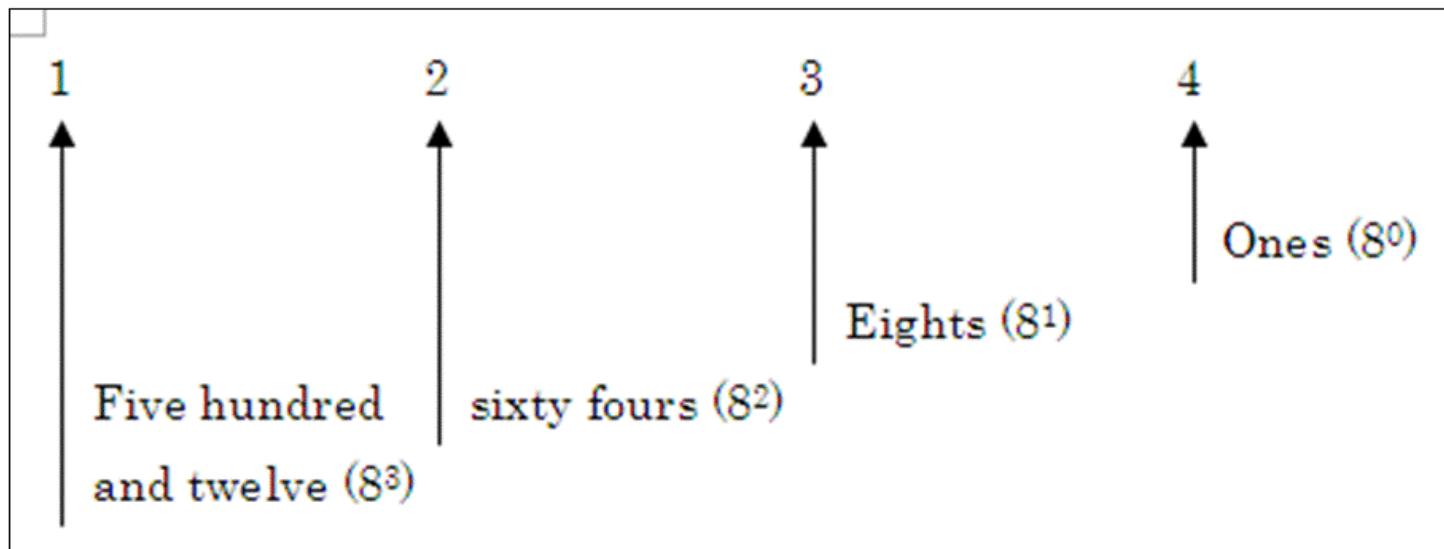
Binary Number System

- This number system uses two digits 1 and 0 to represent numbers. The place value of the digits of a base two number is influenced by two (the base/radix). e.g. the place values of the 1010_2 are illustrated below
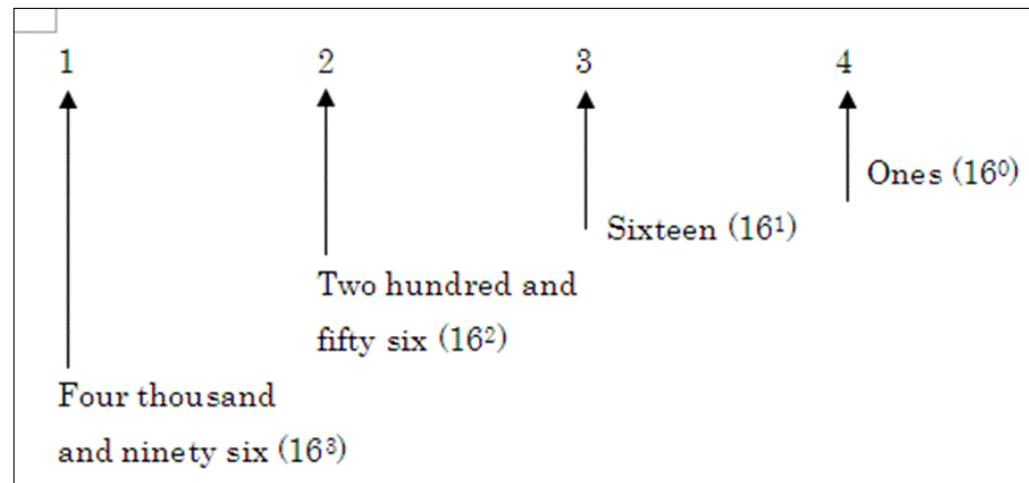
Octal Notation

- The octal number system consists of eight digits running from 0-7. each digit of a number in base eight has its place value determined by eight. E.g. 2163_8 can be expressed as.

1	2	3	4
Five hundred and twelve (8^3)	sixty fours (8^2)	Eights (8^1)	Ones (8^0)

Hexadecimal Number System

- This number system consists of sixteen digits ranging from 0-9 and letters A-F where A is equivalent to 10, B to 11 up to F which is equivalent to 15, in base ten systems. Each digit of a number in base sixteen has its place value expressed in terms of sixteen. For example, the value 12A3 can be expressed as;

1
2
3
4

Ones (16^0)

Sixteen (16^1)

Two hundred and fifty six (16^2)

Four thousand and ninety six (16^3)

Base conversion

- Decimal to Binary
- Binary to Decimal Number
- Decimal to Octal
- From Octal to Decimal
- Converting Octal to Binary
- From Decimal to Hexadecimal
- Hexadecimal to Decimal
- Hexadecimal to Binary

Decimal to Binary 1

- To convert a number from base 10 to base 2, the number in base 10 is repeatedly divided by 2 unit the divided is zero on each division the remainder is noted. The remainders are copied bottom up wise to give the binary equivalent of the decimal number.

Example 1

121_{10} to Binary

2	121	Rem
2	60	1
2	30	0
2	15	0
2	7	1
2	3	1
2	1	1
	0	1

Copy

$121_{10} \Rightarrow 1111002_2$

Example 2

26_{10} to Binary

2	26	Rem
2	13	0
2	6	1
2	3	0
2	1	1
	0	0

Copy

$26_{10} \Rightarrow 11010_2$

Decimal to Binary 2

Example 1

45.25_{10} to Binary

2	45	Rem
2	22	1
2	11	0
2	5	1
2	2	1
1	1	0	Copy $45 \Rightarrow 101101_2$
	0	1

$42.25 \Rightarrow 101101.01_2$

Fraction part. (0.25)

$0.25 \times 2 = 0.05$ Copy the whole units
$0.50 \times 2 = 1.00$ downwards

$0.25 \Rightarrow 0.01$

Example 2

0.357_{10} to Binary

0 is the whole number it remains a 0

Fraction part 0.375

$0.375 \times 2 = 0.750$

$0.750 \times 2 = 1.500$

$0.500 \times 2 = 1.000$ (Fraction become zero)

$0.375 \Rightarrow 0.011_2$

- For fractional numbers the number is divided into two parts. The whole number part and the fraction part. The whole number is repeatedly divided by 2 while the fraction is repetitively multiplied by 2 until the fraction becomes zero or starts recurring. (Repeating it self)

Binary to Decimal Number 1

Example

11110_2 to decimal

1	1	1	1	0
2^4	2^3	2^2	2^1	2^0
$(16{\times}1)$	$(8{\times}1)$	$(4{\times}1)$	$(2{\times}1)$	$(1{\times}0)$

$16+8+4+0+2=31_{10}$

$11110_2 \Rightarrow 62_{10}$

Example

1001_2 to decimal

1	0	0	1
2^3	2^2	2^1	2^0
$(8{\times}1)$	$(4{\times}1)$	$(2{\times}0)$	$(1{\times}1)$

$8+0+0+1=9_{10}$

$1001 \Rightarrow 9_{10}$

- **Steps**
  1. First write the place values starting from the right hand side.
  2. Write each digit under its place value.
  3. Multiply each digit by its corresponding place value.
  4. Sum up the products.

Binary to Decimal Number 2

Example

11010.01_2 to decimal

1	1	0	1	0	.	0	1
2^4	2^3	2^2	2^2	2^0	.	2^{-1}	2^{-2}
(16×1)	(8×1)	(4×0)	(2×1)	(1×0)	.	$(1/2 \times 0)$	$(1/4 \times 1)$

$16 + 8 + 0 + 2 + 0.0 + 1/4 = 26.25_{10}$

$11010.01 \Rightarrow 26.25_{10}$

Example

10011.001_2 to decimal

1	0	0	1	1	.	0	0	1
2^4	2^3	2^2	2^2	2^0	.	2^{-1}	2^{-2}	2^{-3}
(16×1)	(8×0)	(4×0)	(2×1)	(1×0)	.	$(1/2 \times 0)$	$(1/4 \times 1)$	$(1/8)$

$16 + 0 + 0 + 2 + 1.0.125$

$19 + 0.125_{10}$

$10011.001_2 \Rightarrow 19.125_{10}$

- For fraction number, the whole number is converted to decimal as above. The fraction part digits are divided by multiples of two starting from the decimal point.

Decimal to octal 1

Example 1

691_{10} to octal

8	691	Rem
8	86	3
8	10	6
8	1	2
8	0	1

Copy

$691_{10} \Rightarrow 1263_8$

Example 2

261_{10} to octal

8	261	Rem
8	32	5
8	4	0
8	0	4

Copy

$261_{10} \Rightarrow 405_8$

- To convert a number from base 10 to base 8, the number in base 10 is repeatedly divided by 8, until the dividend is zero each division the remainder is noted.

Decimal to octal 2

■ To convert fractional decimal to octal the procedure is same as for fractional decimal to binary, but the multiplication is by 8 instead of 2.

Example

261_{10} to octal

8	98	Rem
8	12	2
8	1	4	Copy
	0	1	

Fraction Part

$0.125 \times 8 = 1.00$ (Fraction is zero)

$98.125_{10} \Rightarrow 142.1_8$

Form Octal to Decimal

Example 1

1263_8 to decimal

1	2	6	3
8^3	8^2	8^1	8^0
(512×1)	(64×2)	(8×6)	(1×3)

$512 + 128 + 48 + 3 = 691_{10} \Rightarrow 1263_8 \Rightarrow 261_{10}$

Example 2

143.1_8 to decimal

1	4	2	.	1
8^2	8^1	8^0	.	8^{-1}
(64×1)	(8×2)	(1×2)	.	$(1/8 \times 1)$

$64 + 32 + 2 + 1/8$

$142.1_8 \Rightarrow 98.125_{10}$

- The procedure is same as from binary to decimal, but the weights as assigned in terms of 8's.

- The fractional number are converted to decimal using the same procedure as that of converting fractional binary to decimal, but divisions is by multiplies of eight.

Converting Octal to Binary

Example 1

523_8 to binary

Octal	Binary
5	101
2	010
3	011	combine the three from left to right.

$521_8 \Rightarrow 100110000_2$

Example 2

460_8 to binary

Octal	Binary
4	100
6	110
0	000

$460_8 \Rightarrow 100110000_2$

- Each digit is represented by 3 binary digits. E.g. 7 is represented as 111_2

From Decimal to Hexadecimal 1

Example 1

122_{10}

16	122	Rem
16	7	10
	0	7

Copy

$7A_{16}$

Example 2

169_{10}

16	169	Rem
16	10	9
	0	10

Copy

$A9_{16}$

- The procedure is same as from decimal to binary though divisions are by 16

From Decimal to Hexadecimal 2

- For the fractional numbers, the fraction part is multiplied by 16.

Example

32.125_{10}

16	32	Rem
16	2	0
	0	2

20

$0.125 \times 16 = \underline{2}.00$ (Fraction Zero stop)

$32.125_{10} \Rightarrow 20.2_{16}$

Hexadecimal to Decimal

Example 1

$7A_{16}$ to decimal

7	A
16^1	16^0
(16×7)	(1×10)

$112 + 10 = 122_{10}$

Example 2

20.2_{16} to decimal

2	0	.	2
16^1	16^0	.	16^{-1}
(16×2)	(1×0)	.	$(1/16 \times 2)$

$32 + 0$. $2/16 \Rightarrow 32.125_{10}$

- The same procedure as from Binary to Decimal is used with the weights assigned in terms of 16's.

- For fractional hexadecimal values. The division is by 16 and the procedure is the same as the binary fraction to decimal.

Hexadecimal to Binary

The hexadecimal numbers are represented using 4 digits.

Hexadecimal	Binary
00	0000
01	0001
02	0010
03	0011
04	0100
05	0101
06	0110
07	0111
08	1000
09	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Examples of hexadecimal to binary

Example 1

321_{16} to Binary

3	0011
2	0010
1	0001

Combine the three sets of bits

$321_{16} \Rightarrow 001100100001_2$

Example 2

ADC_{16} to Binary

A	1010
D	1101
C	1100

$ADC_{16} \Rightarrow 101011011100_2$

Negative numbers

- Both positive and negative numbers can be represented in the computer memory during processing. various methods are used to represent negative numbers in the computer. These methods are:

    i. One's compliment
    ii. Two's compliment

    The negative numbers are used to carry out subtraction in the computer arithmetic operations.

One's complement (1c method)

- In this method the binary bits representing the negative number are negated that is the 1's in the number are changed to 0's and 0's to 1's.

E.g. -17_{10} can be represented in binary as a negative as follows.

$\Rightarrow 17_{10} \longrightarrow 10001_2$

$-17_{10} \Rightarrow$ 1   0   0   0 1_2

        0   1   1   1   0

Therefore $-17_{10} \Rightarrow 011110_2$ (1 compliment)

Two's complement (2c method)

■ In this method, the negative number is represented into binary, then complimented as in 1c method but a '1' is added to the least significant digit of the complement value.

Example -17_{10} can be represented as:

$\Rightarrow 17_{10} \Rightarrow 10001_2$

-17_{10} ⟶ 1 0 0 0 1

	0	1	1	1	0	(1'c)
+					1	
	0	1	1	1	1	(2'c)

Signed magnitude

- In decimal numbers, signed number has a prefix "+" for a positive number e.g. $+32_{10}$ and "-" for a negative number e.g. -32_{10}. in binary, a negative number may be represented by prefixing a digit 1 to the number while a positive number may be represented by prefixing a digit 0.

i.e. $17_{10} \Rightarrow 10001_2$ (unsigned)

$-17_{10} \Rightarrow$ To indicate it's a negative number we add an extra bit (1) 10001_2

$+17_{10} \Rightarrow$ To indicate it's positive we add an extra bit (0) 10001_2

Advantages of two's compliment

- There are no two ways of representing a zero as in one's compliment and signed magnitude.

- Effective addition and subtraction can be done even with numbers that are represented with a sign bit.

Binary arithmetics

- Binary addition

- Binary subtraction

- Binary multiplication

Binary addition

Example -17_{10} can be represented as:

$\Rightarrow 17_{10} \Rightarrow 10001_2$

-17_{10} →	1	0	0	0	1	
	0	1	1	1	0	(1'c)
+					1	
	0	1	1	1	1	(2'c)

Example 2

	1	1	1
+	0	1	1
①	0	1	0

Overflow

= ① 0 1 0_2

Overflow

- Binary addition rules

A+B	sum	+	carry
0+0	0		0
0+1	1		0
1+0	1		0
1+1	0		1
1+1+1	1		1

Binary subtraction

Example 1

1	1	1	0	1_2
-	1	0	1	0_2
1	0	0	1	1_2

Example 2

1	0	1	0	1_2
-		1	1	0_2
	1	1	1	1_2

- Binary subtraction rule

A-B	difference	borrow
0-0	0	0
0-1	1	1
1-0	1	1
1-1	0	0

Subtraction using ones complements

Example 2

31-17

Decimal	Binary
31 ————————→	11111_2

$17 \Rightarrow 10001_2$

Convert 17 to 1 compliments

10001_2 1'c $\Rightarrow$ 01110_2

Add the binary 01110_2 to binary 31

```
    11111
+   01110
  1 01101₂
```

① ———→ Over flow

Example 1

5-3

Decimal	Binary
5 ————————→	0101_2
3 ————————→	0011_2

Convert 3 to 1 complements

0011_2

1'c $\Rightarrow$ 1100_2 add the binary 1100_2 to binary 5

```
    0   1   0   1₂
+   1   1   0   0₂
    0   0   0   1₂
```

Subtraction using two's compliment

Example 1

27-5

Decimal	binary
27	$\longrightarrow$ 00011011_2
5	$\longrightarrow$ 00000101_2

The 1's complement of 5 $\Rightarrow$ 111110101_2

The 2's complement of 5 $\Rightarrow$ + _____1

11111011_2

Add 2's complement of 5 to binary 27

$\Rightarrow$

$\qquad$ 00011011

$\qquad$ + 11111011

$\qquad$ ① 00010110_2

$\qquad\qquad$ 00010110

Example 2

14-7

Decimal	binary
14	1110_2
7	0111_2

The 1's complement of 7 $\Rightarrow$ 1000_2

The 2's complement of 7 $\Rightarrow$ + ____1

1001_2

Add 2's complement of 7 to binary 14

$\Rightarrow$

$\qquad$ 1110

$\qquad$ + 1001

$\qquad$ ① 0111

$\qquad\qquad$ Overflow $\qquad$ 0111_2

Binary multiplication

Multiplication rule	Product
A * B	0
0 * 0	0
0 * 1	0
1 * 1	1

Examples of binary multiplication

Example 1

```
        1111₂
      * 110
        1111
        1111
        0000
    ① 011010
```

Overflow ⇒ 011010

Example 2

```
        110
      * 11
        110
    +  110
    ①  0010
```

Overflow = 0010₂

Symbolic representation using coding system

- Binary Coded Decimal (BCD)
- Extended BCD code / standard BCD code
- Extended Binary Coded Decimal Interchange Code (EBCDIC)
- American Standard Code for Information Interchange (ASCII)

Binary Coded Decimal (BCD)

- BCD is a 4 bit code used to represent numeric data only. For example, a number like 3 can be represented using binary coded decimal as 0011_2. BCD is mostly used in simple electronics devices like calculators etc.

Extended BCD code / standard BCD code

- This standard binary coded decimal is an enhanced format of binary coded decimal (BDC) it's a 7-bit representation scheme which can represent non numeric characters. This allows 64 characters to be represented. For example letter A is represented as 110001_2 using standard BCD code.

Extended Binary Coded Decimal Interchange Code (EBCDIC)

- EBCDIC is an 8-bit character coding scheme used primarily in IBM computers. A total of 256 (2^8) characters can be coded using this scheme. For example letter A is represented as 11000001_2 in EBCDIC code.

American Standard Code for Information Interchange (ASCII)

- ASCII is a 7-bit code, which means that only 128 characters can be represented. There is an 8th bit to this coding scheme which can provide for 256 characters. The 8th bit coding scheme is referred to as an 9-bit ASCII for example letter A is represented as

$$1000001_2 \text{ in ASCII code}$$

Review questions 1

1. Differentiate between analog and digital data.
2. Define the following terms:
   a. Amplitude
   b. Frequency
   c. Periodic time
3. Explain the role of a modem in communication.
4. State the reasons for use of binary in digital technology.
5. Using illustration, explain how data is represented:
   a. In electronic circuits
   b. On magnetic media
   c. On CD-ROM
6. Distinguish between:
   a. A byte and a nibble
   b. Word and word length

Answer of review questions 1-1

1. Analog data is a type of data that is continuous in nature while digital data is in discrete form.

2. a). Amplitude is the maximum.

   b). Frequency is the number of cycles per second.

   c). Periodic time is the time taken by a signal to complete a cycle.

3. A modem converts (modulates) data to be transmitted over the telephone line from digital signal to analog form. At the receiving end, the modem attached to the receiving computer converts (demodulates) the analog signal to original digital form.

4. Devices designed using binary logic are simple, more reliable and consume less energy.

Answer of review questions 1-2

5. a). In electronic circuits – availability of high voltage 'on' is interpreted as a '1' while low voltage 'off' is interpreted as a '0'.

   b). On magnetic media, arrangement of magnetic dipoles of iron oxide in one direction is interpreted as '1s" while those that face in the opposite direction are interpreted as '0s'.

   c). On the shiny surface of the CD are pits and lands. When a laser beam falls on the shinny surface, it is reflected and interpreted as a '1'. On the other hand, when the laser beam takes on the pit or bump, there is no reflection and this is interpreted as a '0' bit.

6. a). A byte is made up of seven or eight bits while a nibble is made up of four bits i.e. it is half of a byte.

   b). A word consists of two or more bytes while a word length is used as a measure of the number of bits in each word.

Review questions 2

- Convert the following base two numbers into denary (base 10) numbers.

  a. 0101_2

  b. 1111_2

  c. 10101101110_2

  d. 101111111_2

  e. 1011011_2

  f. 111000111_2

Answer of review questions 2

a. 5_{10}

b. 15_{10}

c. 1390_{10}

d. 191_{10}

e. 89_{10}

f. 455_{10}

Review questions 3

- Using the place value and log division methods convert each of the following base 10 numbers to their binary equivalents.

  a. 10_{10}

  b. 43_{10}

  c. 365_{10}

  d. 512_{10}

  e. 143

  f. 954_{10}

Answer of review questions 3

a. 1010_2

b. 101011_2

c. 101101101_2

d. 1000000000_2

e. 10001111_2

f. 1110111010_2

Review questions 4

1. Convert the following binary numbers to their decimal equivalents.

   a. 0.10011_2

   b. 0.0010_2

   c. 0.10101_2

   d. 11.0110_2

   e. 101.11110_2

   f. 100.110_2

2. Convert the following decimal numbers into their binary equivalents.

   a. 0.635_{10}

   b. 0.450_{10}

   c. 2.500_{10}

   d. 5.1625_{10}

   e. 7.1875_{10}

   f. 0.350_{10}

Answer of review questions 4

1. Convert the binary numbers to their decimal equivalents.

a. 0.59375_{10}

b. 0.125_{10}

c. 0.65625_{10}

d. 3.375_{10}

e. 5.9375_{10}

f. 4.75_{10}

2. Convert the decimal numbers into their binary equivalents.

a. 0.101_2

b. 0.01110_2

c. 10.1000_2

d. 101.001010011

e. 111.0011_2

f. 110101.0101110_2

Review questions 5

- Convert the following octal numbers to decimal numbers.

a. 77_8

b. 64_8

c. 102_8

d. 1200_8

e. 1000_8

f. 173_8

g. 123_8

h. 777_8

i. 345_8

j. 166_8

Answer of review questions 5

a. 63_{10}

b. 52_{10}

c. 66_{10}

d. 640_{10}

e. 512_{10}

f. 123_{10}

g. 83_{10}

h. 511_{10}

i. 229_{10}

j. 118_{10}

Review questions 6

- Convert the following octal numbers to their binary equivalents.

  a. 244_8

  b. 247_8

  c. 162_8

  d. 565_8

  e. 222_8

  f. 3370_8

  g. 1413_8

  h. 1331_8

  i. 6347_8

  j. 46653_8

Answer of review questions 6

a. 10100100_2

b. 10100111_2

c. 1110010_2

d. 101110101_2

e. 10010010_2

f. 11011111000_2

g. 11000011011_2

h. 1011011001_2

i. 110011100111_2

j. 100110110101011_2

Review questions 7

- Convert the following hexadecimal numbers into decimal numbers.
  a. 32_{16}
  b. CCD_{16}
  c. EFE_{16}
  d. 119_{16}
  e. 328_{16}
  f. ABD_{16}
  g. $10AFFD_{16}$
  h. $DDFF34_{16}$
  i. $11ABDF_{16}$
  j. $CDFF31_{16}$

Answer of review questions 7

a. 50_{10}

b. 3277_{10}

c. 3838_{10}

d. 281_{10}

e. 808_{10}

f. 954_{10}

g. 1093629_{10}

h. 14548788_{10}

i. 1158111_{10}

j. 13500209_{10}

Review questions 8

- Convert the following hexadecimal numbers into binary form
  a. 29416
  b. 24716
  c. EFD16
  d. 90716
  e. BAD16
  f. CBDF16
  g. 585C16
  h. ABCD16
  i. 55C3A16
  j. 33111CF16
  k. Convert 1111000110111 to its hexadecimal equivalent.

Answer of review questions 8

a. 1010010100_2

b. 1001000111_2

c. 11101111111101_2

d. 100100000111_2

e. 101110101101_2

f. 1100101111011111_2

g. 1011000010111100_2

h. 10101011111001101_2

i. 10101011100001111010_2

j. 1100110001000100 0111001111_2

k. $IE37_{16}$

Review questions 9

Work out the following binary additions

1. 1010_2
     $+\ 111_2$
     _____

2. 1111_2
     $+\ 111_2$
     _____

3. 1011_2
     $+\ 1110_2$
     _____

4. 11101_2
     $+\ 10110_2$
     _____

5. 1000111_2
     $+\ 10010_2$
     _____

6. 1101_2
     101_2
     $+\ 11_2$
     _____

7. 111110_2
     111_2
     $+\ 10_2$
     _____

8. 1000011_2
     10101_2
     $+\ 111_2$
     _____

9. 101110_2
     1111_2
     $+\ 11_2$
     _____

10. 1111111_2
     111111_2
     $+\ 1101_2$
     _____

11. 100101_2
     11011_2
     $+\ 1101_2$
     _____

12. 110010_2
     111011_2
     $+\ 1100_2$
     _____

Answer of review questions 9

1. 10001_2
2. 10110_2
3. 11001_2
4. 110011_2
5. 1011001_2
6. 10101_2
7. 1000111_2
8. 111111_2
9. 1000000_2
10. 11001011_2
11. 1001011_2
12. 1111001_2

Review questions 10

Work out the following binary subtractions

1. 11001_2 2. 101_2 3. 11011_2 4. 1100_2 5. 111011_2

   $- 1010_2$ $- 100_2$ $- 111_2$ $- 011_2$ $- 110_2$

   _____   _____   _____   _____   _____

6. $100010_2 - 11_2$ 7. $01101_2 - 1011_2$ 8. $11111111_2 - 10101101_2$

9. $11101101_2 - 100111_2$ 10. $100000_2 - 1111_2$

Answer of review questions 10

1. 1111_2
2. 001_2
3. 10100_2
4. 1001_2
5. 110101_2

1. 11111_2
2. 10_2
3. 1010010_2
4. 11000110_2
5. 10001_2

Review questions 11

1. Convert the decimal number -7 to 8-bit binary using ones and twos complement.

2. Using 8-bits length, find the twos complement of the following decimal numbers.
   A) -31_{10}
   B) -28_{10}
   C) -5_{10}

3. Using ones and twos complement perform the following arithmetic operations.
   A) $14_{10} - 7_{10}$
   B) $28_{10} - 12_{10}$
   C) $34_{10} - 33_{10}$
   D) $100_{10} - 50_{10}$
   E) $102_{10} - 64_{10}$

Answer the review questions 11

1. 11111001_2

2. A). 11100001_2

   B). 11100100_2

   C). 11111011_2

3. A). 111_2

   B). 10000_2

   C). 1_2

   D). 110010_2

   E). 0100112_2

Revision questions 1

1. Explain the two types of data signals.

2. Differentiate between
   A) Octal and decimal number systems.
   B) Binary and hexadecimal number systems.

3. Convert the following binary numbers to their decimal equivalents.
   1. 101110_2
   2. 101011_2
   3. 0110_2
   4. 10101_2

4. Convert the following decimal numbers to their decimal binary equivalents.
   A) 789_{10}
   B) 570_{10}
   C) 42_{10}
   D) 30_{10}

5. Find the sum of the following binary numbers.
   A) $1110_2 + 1111_2 =$
   B) $001_2 + 100_2 =$
   C) $1101_2 + 1011_2 + 100_2 =$
   D) $1010_2 + 101_2 =$
   E) $111_2 + 11_2 =$

Answer of revision questions 1

1. Analog and digital
2. A). Octal number system is a base 8 system while the decimal number system is a base 10 system.

   B). Binary number system is a base 2 system while hexadecimal number system is a base 16 system.
3. A). 46_{10}

   B). 43_{10}

   C). 6_{10}

   D). 22_{10}
4. A). 1100010101_2

   B). 1000111010_2

   C). 101010_2

   D). 11110_2

Revision questions 2

1. Work out the following binary subtraction using direct, ones and twos complement methods.
   A) $11001 - 1101$
   B) $1000 - 101$
   C) $100011 - 111$
   D) $10101110 - 1000110$
   E) $10001000110 - 101$

2. Find the ones and twos complement of the following decimal number in binary form.
   A) -75_{10}
   B) -80_{10}
   C) -100_{10}

3. Determine the value of X in the equations:
   A) $100110_2 - X_2 = 001010_2$
   B) $X_2 + 1101_2 = 70_{10}$

4. Work out the decimal equivalents of the following binary numbers.
   A) 0.10010_2
   B) 101.11_2
   C) 11.101_2
   D) 0.001_2

5. Work out the decimal equivalents of the following binary numbers.
   A) 0.35_{10}
   B) 2.50_{10}
   C) 0.20_{10}
   D) 7.125_{10}

Answer of revision questions 2

1. A). 10101_2
   B). 101_2
   C). 11100_2
   D). 1111_2
   E). 1010_2
2. A). 1100_2
   B). 011_2
   C). 11100_2
   D). 1101000_2
   E). 10001000001_2
3. A). 11100_2
   B). 111001_2

1. A). 0.562510
   B). 5.7510
   C). 3.62510
   D). 0.12510
2. A). 0.0101102
   B). 10.12
   C). 0.00112
   D). 111.0012
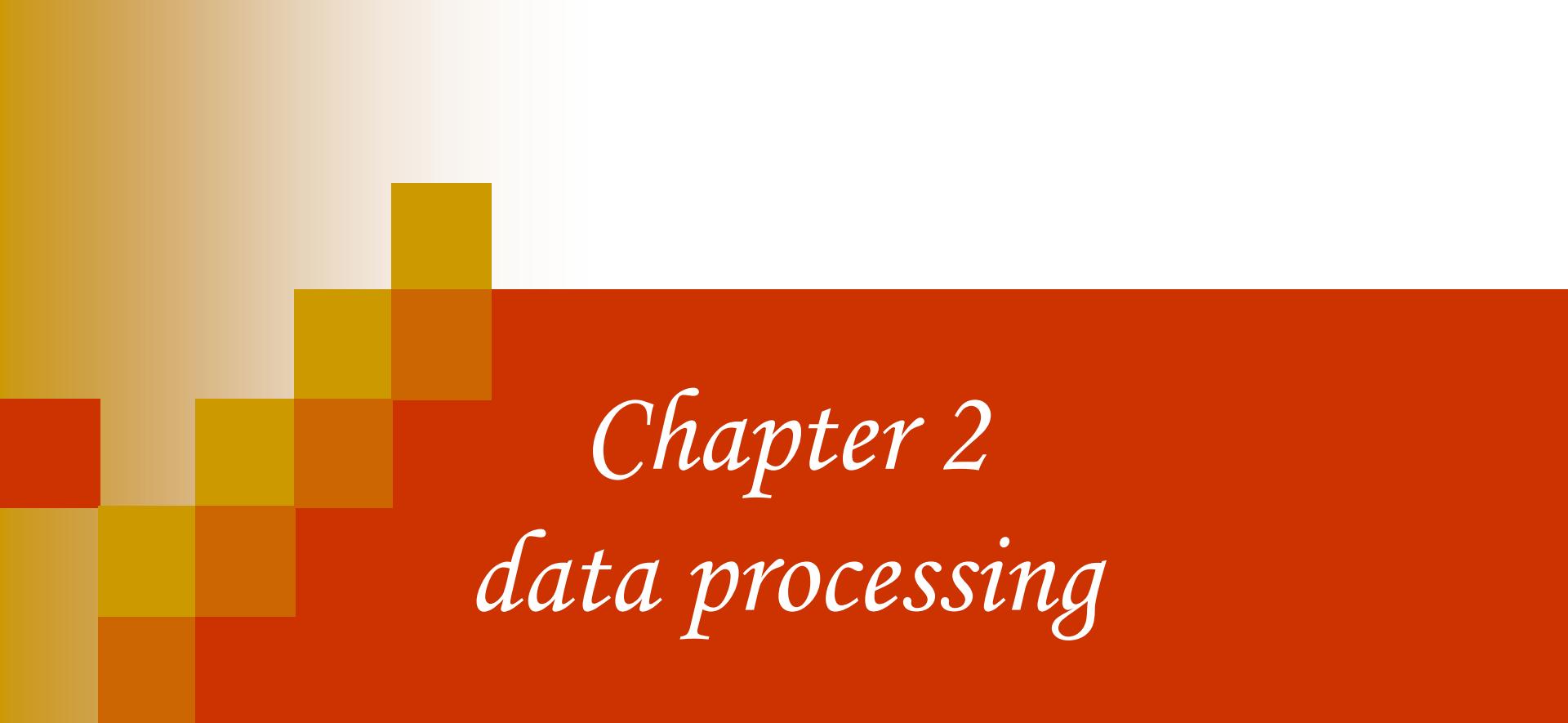
Revision questions 3

1. Write down the character represented by the following codes:
    A) ASCII
        i. 0110000
        ii. 0110111
        iii. 1000011
        iv. 1011001
    B) EBCDIC
        1. 11110000
        2. 11110111
        3. 11000001
        4. 01011101

Answer of revision questions 3

	ASCII	Character	EBCDIC	Character
A).	0110000	0	11110000	0
B).	0110111	7	11110111	7
C).	1000011	C	11000001	A
D).	1011001	Y	01011101	)

	Number	1's complement	2s complement
A)	-75	0110100	0110101
B)	-80	0101111	0110000
C)	-100	0011011	0011100
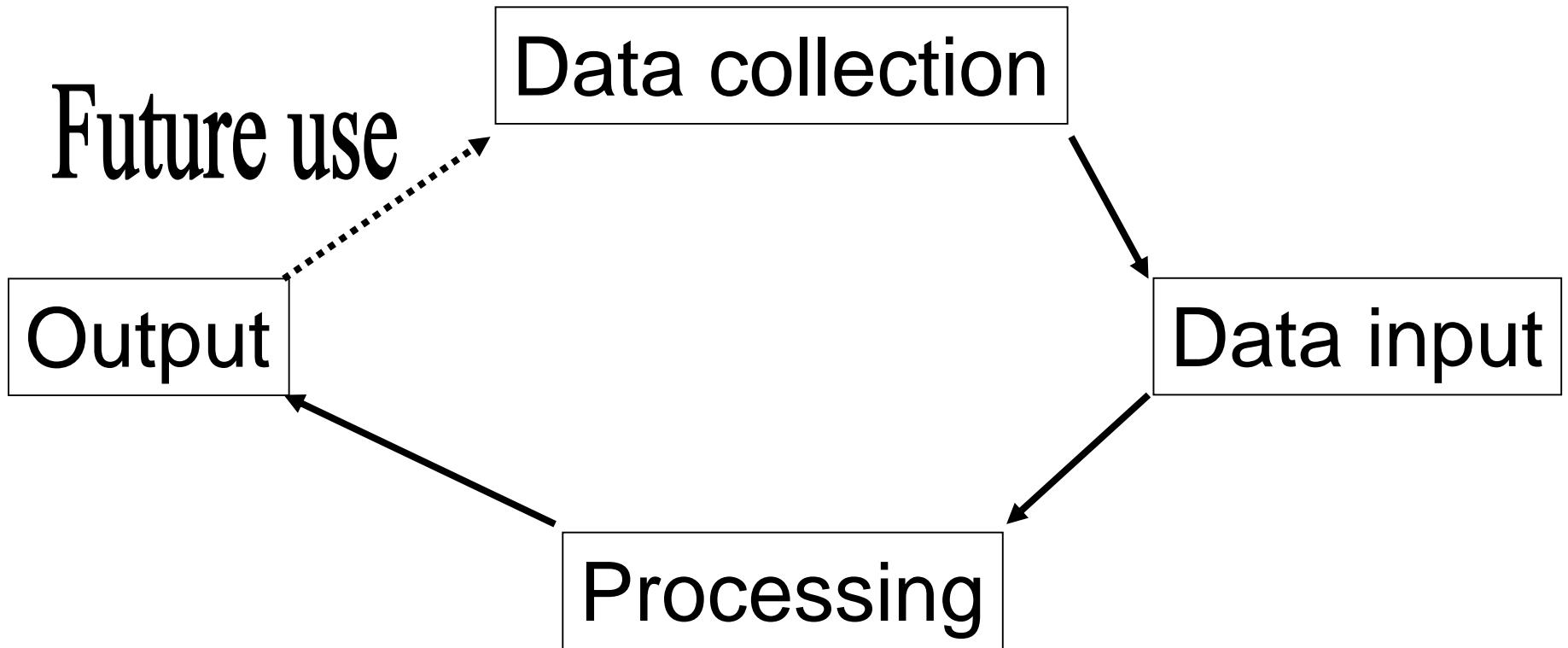
Chapter 2
data processing

Definition of Terms

- Data – this is a collection of raw facts (figures, letters, characters etc) that convey little meaning by themselves.

- Information – this refers to data that has undergone processing and is meaningful to the user when he/she need it, in the form he needs and at the time he/she need it.

- Data processing – refers to the process of transforming raw facts into meaningful output (information)

- Data processing cycle – refers to input process output stages that data goes through to be transformed into information.

Data processing cycle

Future use

Data collection

Output

Data input

Processing

Stages of data processing cycle

1. Data collection – this involves looking for or getting the data from its point of origin for processing purpose.
2. Data input – this is the process where the collected data is converted from human readable form to machine – readable form.
3. Processing – this is transformation of input data by the central processing unit to a more meaningful output.
4. Output – this is the final activity of data processing cycle which produces the desired output (information). The information is then distributed to the target groups.

Methods of data collection

- Interview

- Questionnaire

- Observation

- Record inspection

Stages of data collection

- Depending on the method of data collection used, the process of data collection may involve the following stages.

  - Data creation
  - Data transmission
  - Data preparation
  - Media conversion
  - Input validation
  - Sorting

Data creation

- This is the process of putting together facts in an organised format. This may be in form of manually prepared documents or captured from the source using data capture devices. E.g. scanners, digital camera etc.

Data transmission

- This refers to the transfer of data from the point of collection (source) of the point where processing is to be done. Transmission can be electronically through computer – to – computer, physically by the post office etc.

Data preparation

- This refers to conversion of data from source document to machine – readable form. Data collected using devices that capture data in digital form do not require transcription conversion.

Media conversion

- Data may need to be converted from one medium to another. E.g. from a compact disk to hard disk for faster input.

Input validation

- Data entered into the computer is subjected to validity and verification checks before processing.

  - Verification – this involves checking that what is on input document is exactly the same as what is entered into the computer.
  - Validation – this is identification and removal of errors by the computer through the control of the program.

Sorting

- The source documents are arranged in a particular order for easy and faster entry.

Description of errors in data processing

- The accuracy of the data entered in the computer determines the accuracy of the information given out.

Types of data processing errors

- Transcription errors – these errors occurs during data entry. These errors includes misreading errors and transposition errors.

- Computational errors – they occur when an arithmetic operation does not produce the expected results. They include overflow errors, truncation errors and rounding errors.

- Algorithm or logical errors – these errors occurs as a result of wrong algorithm design.

Misreading errors

- They are brought about by incorrect reading of source document by the user and hence entering wrong values. This may be caused by bad hand writing or confusion on source document. E.g. the number '5' can be typed as s or letter 0 typed as Zero.

Transposition errors

- These errors occurs as a result of incorrect arrangement of characters. i.e. putting characters in the wrong order. E.g. the user may enter 542 instead of 542. transcription errors can be eliminated by use of data capture devices.

Overflow errors

■ This errors occurs if the result from a calculation is too large to be stored in the allocated memory space. E.g. if the memory space can accommodate an 8-bit number and the results from arithmetic is 9-bit number.

Truncation errors

- This result from having real numbers that have a long fractional part which cannot fit in the allocated memory space. The computer cut off the extra characters from the fraction part. E.g. a number like 0.854692 can be truncated to four digits as 0.854.

Rounding errors

- This results from a raising or lowering a digit in a real number to the required rounded number. E.g. a number 3.59 can be rounded to 3.6.

Data integrity

- Data integrity refers to the accuracy and completeness of data entered in a computer or received from the information system.

Factors that determine data integrity

- Accuracy

- Timeliness

- Relevance

- Audibility

Accuracy

- This refers to whether information/data is accurate – true or untrue. As long as the correct instructions and data are entered, computers produce accurate results efficiently.

Timeliness

- This refers to whether the information is available when it is needed, and if its outdated when it is received or when it is used. If information is not available when needed or its outdated by the time its used then has little or no value in decision making.

Relevance

- The data entered must be pertinent to the proce3ssing needs at hand and must meet the requirements of the processing cycle.

Audibility

- Audit ability also refers to as verifiability. This is the ability of users to check the accuracy and completeness of information.

Threats to data integrity
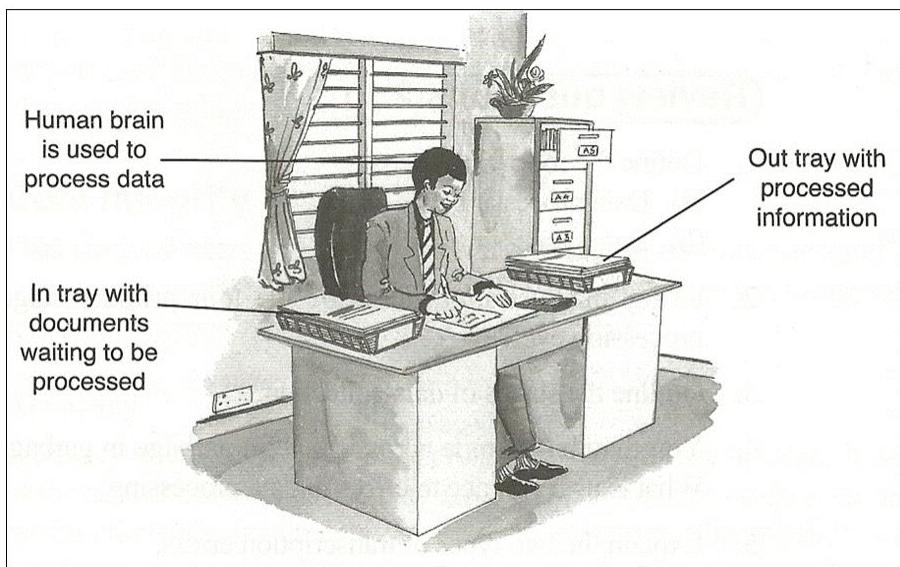
- Threats to data integrity can be minimised through the following;
  - ☐ Using error detection and correction software when transmitting data.
  - ☐ Design user interfaces that minimizes chances of invalid data entry.
  - ☐ Using devices that capture data directly from source e.g. scanners etc.
  - ☐ Control access to data by enforcing security measures.
  - ☐ Back up data preferably on external storage media.

Data processing methods

- Data can be processed using one of the following methods.

    ☐ Mechanically

    ☐ Manually

    ☐ Electronically

Manual data processing

Human brain is used to process data

In tray with documents waiting to be processed

Out tray with processed information

- In this method staff in an organisation uses the laid down procedures and rules to collect data, process and distribute information to relevant departments for use. The tasks are done manually with a pen and a paper. No machines are used to process data instead they use simple tools like tables and rulers.
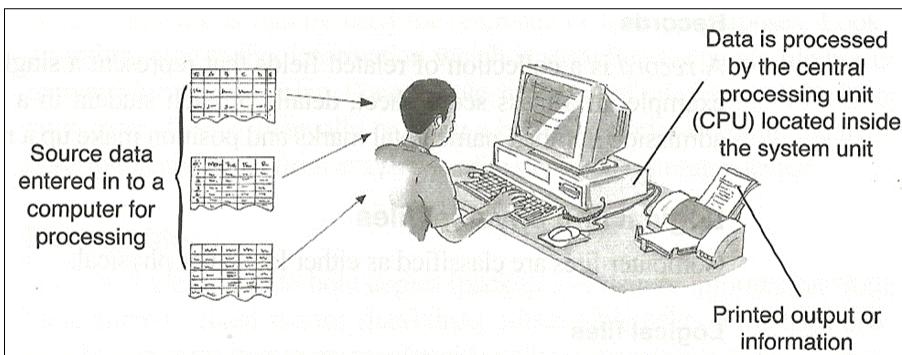
Mechanical data processing

A manual typewriter

- Mechanical systems are those where staff use various mechanical machines like calculators, typewriters, cash registers duplicating machines etc to perform operations.

Electronic data processing

Data is processed by the central processing unit (CPU) located inside the system unit

Source data entered in to a computer for processing

Printed output or information

- In this system, data is manipulated using electronic machines to produce information. This method is faster and more accurate than the manual and mechanical where large volumes of data are to be processed. This method involves the use of computers and other electronic machine like mobile phones, washing machines, modern digital TVs etc.

Factors determining the methods of data processing

- Size and type of business

- Timing aspects

- Link between application

Computer files

- A file is a collection of related records that give a complete set of information about a certain item or entity. A file can be stored manually in a file cabinet or electronically in a computer storage device.

Advantages of computerized filling system

- Information takes up less space than the manual system.
- Enhance data integrity and reduces duplication.
- Offers faster access and retrieval of data.
- It's much easier to update or modify information.

Elements of a computer file

- A computer file is made up 3 elements.
  - Characters – this is the smallest element in a computer file and refer to a letter, number, symbol that can be entered, stored and output by a computer.
  - Field – a field is a single character or a collection of characters that represents a single piece of data. For example in employees record, the name of employee is a field.
  - Records – is a collection of related fields that represents a single entity. For example in employees record, details of each employee in a row. E.g. Name, IDNO, sex, Department etc make up a record. Computer files can be viewed in two ways.

Logical file

- This is the way the user views the file in terms of its contents and the processing to be carried upon them.

Physical file

- This is the actual arrangement of the file contents into the storage media surface and how the processing operations are made possible.

Types of computer processing files

- Master file
- Transaction file (movement file)
- Reference file
- Sort file
- Back – up file
- Report file

Master file

- This is the main file that contains permanent records about a particular item or entries. The master file contains both static and dynamic fields. For example a payroll system might keep a master file containing employees personal details (name, address, date of birth, sex etc) which rarely change (static fields) and data describing the earnings and deductions which frequently change (dynamic fields).

Transaction file (movement file)

- This file includes input and output files for holding temporary incoming or outgoing data. It contains data about an organisation activities taking place for some period of time. This file is used to update dynamic data on master files.

Reference file

- They are permanent or semi permanent and are used for reference or look up purposes. They contain records that are fairly static. E.g. price, lists, Pay As You Earn etc.

Sort file

- These files are created from existing transaction or master files. They are used where data is to be processed sequentially. Data or records are first sorted in a particular format for example they are arranged in ascending or descending order of the key field and stored in the sort file.

Back – up file

- These files are duplicate copies of existing files. They are used to supplement the operational files in case of loss by any reason. They are created any time and update is carried out on the master file.

Report file

- They contain sets of records extracted from data in the master files. They are used to prepare reports that can be printed at a later date, for example, list of absentees, overtime etc.

File processing activities 1

- Updating – refers to changing data in master file to reflect the current status.
- Referencing – happens when it is made to particular records to ascertain what is contained therein. The activity does not alter the contents of master file from a file.
- Sorting – this entails arranging of file contents into some pre – determined sequence of the key field, which occupy the same position on the records.

File processing activities 2

- Merging – the process that combines the contents of two or more input files into one output files.
- Matching – the input files records are matched/compared to ensure that records exist in both the files.
- Summarizing – the records of interest from the file are accumulated to form a record in the output file.
- Searching – it entails looking for the record of interests from the file.

File updating 1

- Hit rate – hit rate is used to describe the rate of processing of a master file in terms of its active records. For example, if 600 transactions are processed each day against a master file of 12,000 records, then the hit rate is said to be 5% i.e. 600/12000*100

File updating 2

- Volatility – this is the frequency with which records are added to the file or deleted from it. If the frequency is high, then the file is said to be 'volatile' otherwise 'static'.

- Size – refers to the amount of data stored in the file. Its expressed in terms of the total number of records in the file.

- Growth – files will always grow as new records are added to it.

File organisation methods

- File organisation is the arrangements of records within a particular file. There are four methods of storing files and retrieving them from secondary storage devices.

Sequential file organisation

- In this organisation records are stored and accessed in a particular order sorted using key field. The key field is used to search for a particular record. Searching commences at the beginning of the file and proceeds to the 'tail' of the file until the record is reached. Mainly used with magnetic tapes.

Advantages of sequential file

- Simple to understand the approach.
- Easy to organise, maintain and understand.
- Reading a record requires only the key field.
- Inexpensive input/output media and devices are used.

Disadvantages of sequential file

- Entire file must be accessed even when the activity rate is very low.

- Random enquires are impossible to handle.

- Data redundancy is typically high.

Random or direct file organisation

- In this organisation records are stored randomly but accessed directly. To access a file stored randomly, a record key is used to determine where a record is stored on the storage media. Used in magnetic and optical disks.

Advantages of random or direct file

- Records are quickly accessed.

- File update is easily achieved.

- They do not require the use of indexes.
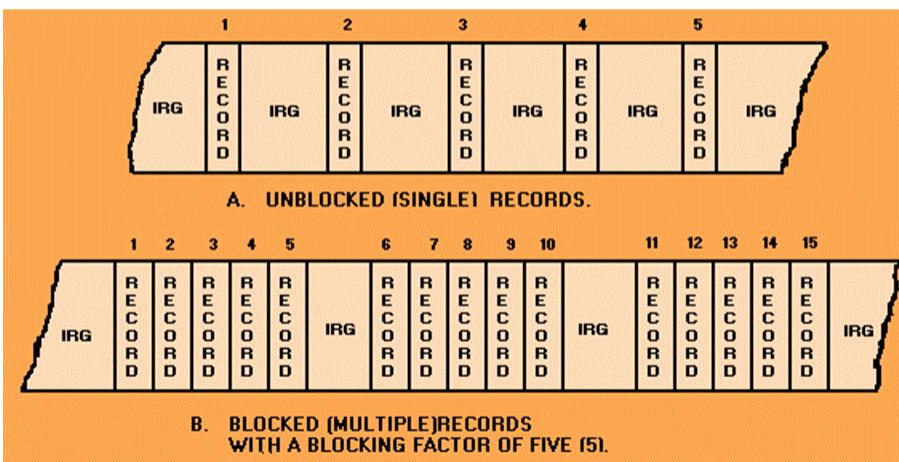
Disadvantages of random or direct file

- Data may be accidentally erased or over written. Unless special precaution taken.
- Expensive hardware and software resources are required.
- Relative complexity of programming.
- System design around it is complex and costly.

Serial file organisation

Records on a magnetic tape

A. UNBLOCKED (SINGLE) RECORDS.

B. BLOCKED (MULTIPLE) RECORDS WITH A BLOCKING FACTOR OF FIVE (5).

- The records are laid out contiguously one after the other in no particular sequence. The records are stored one after the other in the same order they come into the file and there exists no relationship between contiguous records. Used with magnetic tapes.

Indexed sequential file organisation

- This method is similar to sequential method, only that an index is used to enable the computer to locate individual records on the storage media. Used with magnetic disk.

Advantages of indexed sequential file

- Records can be accessed sequentially or randomly.

- Records are not duplicated.

- Accessing of records can be fast if done randomly.

Disadvantages indexed sequential file

- Storage medium is rather expensive.

- Accessing records sequentially is time consuming.

- Processing records sequentially may introduce redundancy.

Electronic data processing modes

- On-line processing
- Time sharing processing
- Real time processing
- Multi-programming / multi-tasking
- Distributed processing
- Batch processing
- Multi-processing
- Interactive processing

On-line processing

- In this processing the result of data processing is available immediately. All the computer peripherals and equipments are under the direct control of the central processor. As soon as the input transactions are available, they are processed to produce the required information. The user can interact with the system at any time of processing using input/output facilities.

Application of on-line processing

- Banking

- Stock exchange

- Stock control

- Water/electricity billing

Advantages of on-line processing

- Files are maintained up to date.

- Information is readily available for current decisions.

- File enquiries possible through the terminals (work station)

Disadvantages of on-line processing

- System are complex to develop.

- Are costly in terms of hardware, software, storage media etc.

Time sharing processing

- In this mode the central processor serves two or more users with different processing requirements (tasks). The central computer is available to a variety of uses, connected to the central computer via communication links who may wish to process their data and receive information. The processor time is divided out appropriately among the user task is into time slices. Control is switched from one job to another, under the influence of the operating system. The time slice is equal for all jobs and it's assigned to jobs waiting in a queue. In one turn, only one slice is allocated to the job. The jobs in the queue are assigned the time slice from the head of the queue towards the tail. Jobs that require more time slices are chained to the tail queue. New jobs into the queue are added at the tail of the queue. The wholly processed jobs output results. The incomplete jobs go back to the tail of the queue to await their next turn.

Application of time sharing processing

- Bureau

- Learning institutions.

- Companies.

Advantages of time sharing processing

- Better services to the users, for the information/output is fast.
- Files enquiries is possible since files are held on-line.
- User interaction is possible through terminals.

Disadvantages of time sharing processing

- User has no control over the central computer.

- Not reliable with regard to data security.

- Response time is slow where there are many tasks.

Real time processing

- In this mode, computer processes the incoming data as soon as it occurs, updates the transactions file and gives an immediate response that would affect the events as they happen. The main purpose of a real time processing is to provide accurate, up to date information hence better services based on a real situation.

Application of real time processing

- Air reservation

- Hotel reservation

- Chemical plant processing

Advantages of real time processing

- Information is readily available for instant decision.

- Provides immediate control / information.

- Its fast and reliable.

Disadvantages of real time processing

- Requires complex and expensive operating system.

- They are not easy to develop.

- Require Front End Processors (FEPS) to relieve the ventral processor.

Multi-programming / multi-tasking

- In this processing more than one program are executed apparently at the same time by a single central processing unit. The operating system allocated each program a time slice and decides what order they will be executed.

Advantages of multi-programming / multi-tasking

- Increases the productivity of the computer by reducing CPU idle time.

- Reduces the incidence of peripheral bound operations.

Disadvantages of multi-programming / multi-tasking

- Requires more expensive CPUs.

- Operating system is complex and more difficult to operate.

Distributed processing

- This processing refers to dividing (distributing) processing tasks to two or more computers that are located on physically separate sites but connected by data transmission media. For example, a distributed database will have different tables of the same database residing on separate computers and processed there as need arises.

Application of distributed processing

- Banks where customers can be served from different branches but information is updated at the head branch.

Advantages of distributed processing

- Less risk to system breakdown.

- In case of data loss, only a small portion of the data is lost.

- Reduction of the load on the host. computer hence faster processing.

Disadvantages of distributed processing

- Expensive due to extra communication costs.

- More sophisticated software required to maintain data integrity.

Batch processing

- This is processing mode where the transactions are accumulated over a period of time and then proce3ssed at a pre-specified time to produce a batch of output. The data to be processed e.g. daily, weekly, monthly etc then processed at once.

Application of batch processing

- Payroll processing.

Advantages of batch processing

■ Simple to develop.

■ Timing reports is not a necessity.

■ The unit cost of processing is low.

Disadvantages of batch processing

- Time lag between origination of the transactions and the information availability.

- Late information is not suitable in situations where instant decisions are requires.

- Its difficult to provide the desired priority scheduling.

Multi-processing

- This refers to the processing of more one task at the same time on different processors of the same computer. In such systems, a computer may contain more than one independent central processing unit which works together in a coordinated way. At a given time the processor may execute instructions from two or more different programs or from different parts of one program simultaneously.

Interactive processing

- In this processing, there is continuous dialogue between the user and the computers. As the program executes, it keeps on prompting the user to provide input or respond to prompts displayed on the screen.

Factor to consider in selecting the data processing mode

- The need for direct information retrieval and/or file interrogation.
- The control over the resources e.g. files, input/output devices.
- The cost of acquiring the relevant hardware, software, media etc.
- The optimization of the processing time.
- The time factor of the information for decision making by mangers.

Review questions 1

1. Define the following terms:
   A) Data processing.
   B) Data processing style.
2. Using an illustration, describe the four primary stages of the data processing cycle.
3. Outline the stages of data collection.
4. You may have come across the term Garbage In Garbage Out (GIGO). What is its relevance to errors in data processing.
5. Explain the two types of transcription errors.
6. State three types of computational errors.
7. Define the term data integrity.
8. Give three factors that determine the integrity of data.
9. State at least five ways of minimising threats to data integrity.

Answer of review questions 1-1

1. A). Data processing refers to the transformation of raw data into meaningful output (information).

   B). Data processing cycle are the stages that data goes through during its transformation into information. The output may be used later as input to another processing task.

2. Data collection → data input → data processing → output.

3. A). Data creation.

   B). Data transmission.

   C). Data preparation.

   D). Media conversion.

   E). Input validation.

   F). Sorting.

Answer of review questions 1-2

4. The accuracy of data entry determines the accuracy of the output.

5. A). Wrong entry due to misreading the source data e.g. entering C instead of G (transcription error).

   B). Transposition – interchanging values e.g. 396 instead of 369.

6. Overflow and underflow.

   Truncation.

   Rounding.

7. A). Data integrity refers to accuracy and completeness of data.

   B). Accuracy, timeliness and relevance.

   C). Banking up, enforcing security, using input devices that capture data directly from the source etc.

Revision questions
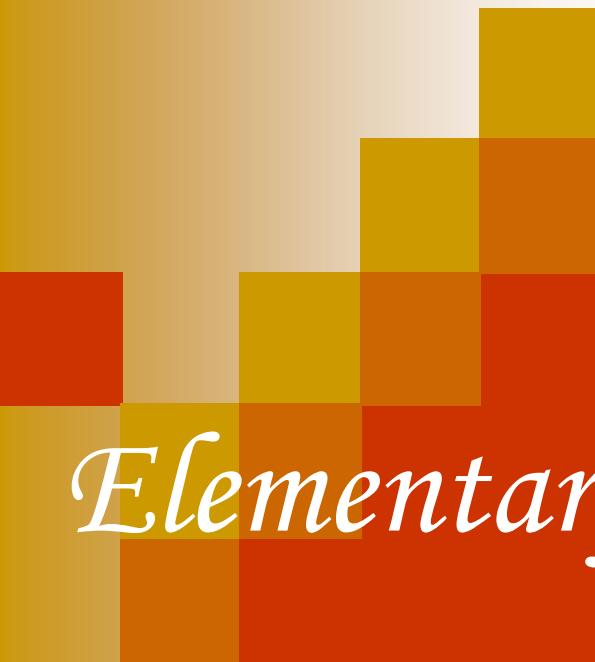
1. Distinguish between data and information.
2. Describe types of data processing methods.
3. State at least four advantages of storing data in computer files over the manual filing system.
4. State elements that make up a computer file.
5. Differentiate between a logical file and physical file.
6. Explain at least six types of computer files.
7. Distinguish between
   A) Sequential and serial file organisation methods.
   B) Random and indexed-sequential file organisation methods.
8. Describe at least five types of electronic data processing modes.

Answer of revision questions 1

1. Data refers to raw facts that do not have much meaning while information refers to the meaningful output after processing.

2. Data processing methods: manual, mechanical and electronic.

3. A). Takes up less space.

   B). Easier to update or modify.

   C). Faster access and retrieval.

   D). Enhances integrity.

4. A). Characters.

   B). Field.

   C). Record.

5. A logical file is viewed in terms of what data items and processing operation may be performed on the data while a physical file is viewed in terms how data is stored and how it is to be processed.

6. A). Master file.

   B). Transaction file.

   C). Reference file.

   D). Backup file.

   E). Report file.

   F). Sort file.

Answer of revision questions 2

1. In sequential file organisation, records are stored in a sorted order using a key field while in serial file organisation, records are not sorted in any way.
2. In random file organisation, records are stored randomly but accessed directly. On the other hand, in indexed-sequential, records are stored sequentially but accessed directly using an index.
3. On-line distributed – the data being processed is accessed, processed and updated in real time.
4. Time-sharing-competing computers are allocated time slots to use computer resources.
5. Batch processing – all the available jobs are processed in bulk all at once.
6. Interactive processing – the computer presents the user with input screens / the user inputs data as processing continues.
7. Real time – data being processed is updated online and the results thereof affect current processing logic.
8. Multi-tasking – several computer programs / tasks run concurrently at the same time sharing resources in shifts.
9. Multiprocessing – several programs / tasks run concurrently on different processors.
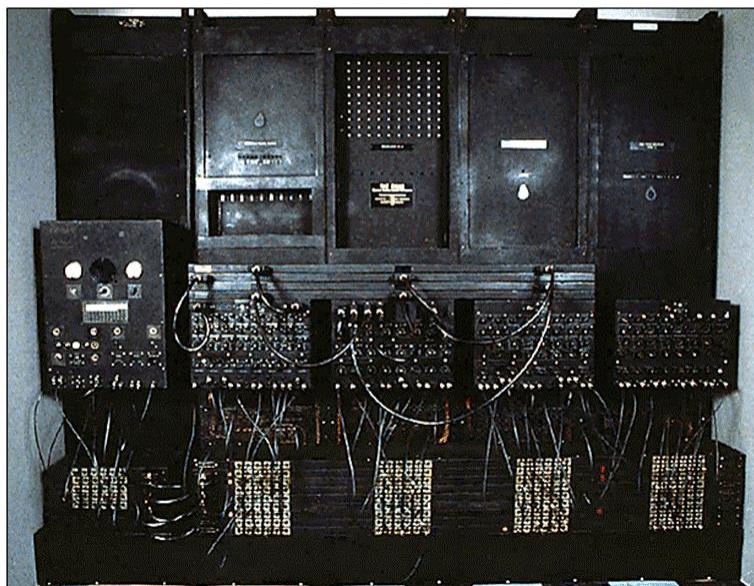
Chapter 3
Elementary programming principles

Introduction

Electronic Numeric Integrator and Calculator (ENIAC)

- Human beings have evolved from the stone age to a highly sophisticated and advanced society by inventing things like the wheel, fire, transistors and today's ultra modern devices like computers. The idea of computers started way back in the nineteenth century. The first generation computers called Electronic Numeric Integrator And Calculator (ENIAC 1) were operated by plugging wires into a control panel that resembles the old telephone switchboards.

Computer programming

- A program – is an organised list of statements (instructions) that when executed, causes the computer to behave in a predetermined manner or carry out a defined task.

- Programming – refers to the process of developing computer (instructions) programs used to solve a particular task.

- A computer program – is designed using a particular programming language. Each language has a special sequence or order of writing characters usually referred to as syntax.

Terms used in programming

- Source program
- Object code
- Translator
- Assembler
- Interpreter
- Compiler

Source program

- This refers to the program code that the programmer enters in the program editor window that is not yet translated into machine-readable form.

Object code

- This refers to the program code that is in machine-readable. i.e. a source code that has been translated into machine language.

Translator

- These are programs that translates/convert the source program into object code. E.g. assemblers, compliers, interpreters etc.

Assembler

- An assembler translates a program written in assembly language into machine language,

Interpreter

- This translates the source program line-by-line, allowing the CPU to execute one line before translating the next. The translated line is not stored in the computer memory, hence every time the program is executed, it has to be translated.

Complier

- This translates the entire source program into object code. The compiler translates each high-level instruction into several machine code instructions in a process called COMPILATION and produces a complete independent program that can be run by the computer as often as required without the original source program being present.

Difference between the interpreters and compilers

Interpreters	Compilers
Translates the source program one statement at a time.	Translates the entire source code at once before execution.
Translates the program each time it is run hence slower than compiling.	Compiled program (object code) can be saved on a storage media and run as required, hence executes faster than interpreted programs.
Interpreted object code takes less memory compared to compiled program.	Compiled programs require more memory as the object file are larger.

Level of programming languages

- There are 2 major levels namely;

  - Low-level languages.

  - High-level languages.

Low-level languages

- These languages are classified as low because they can be easily understood by the computer directly or they require little effort to translate into computer understandable form. These languages are hardware oriented and therefore they are not portable. i.e. a program written for one computer cannot be installed and used on another.

High-level languages

- These 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. These languages are machine independent. This means that a programmer concentrates on problem solving during a programming session rather than how a machine operates.

Types of low level languages

- Machine language (First generation languages).

- Assembly languages (Second generation languages)

Machine language (First generation languages)

- In this language, instructions are written using binary logic. Given that data and instructions are in binary form, many lines of codes are needed to accomplish even a simple task like adding two numbers i.e. a program written in this language look like this.

  11100110   00000011   19999991

  00001111   10001101

  10001111   11111111   10000011

  The program code is hard for human to understand but it's easily understood by computers.

Assembly languages (Second generation languages)

- This language is close to the machines vocabulary rather than human beings vocabulary. It was developed in order to overcome the difficulties of understanding and using machine language. This language helps the programmers to write programs as a set of symbolic operation codes called Mnemonics. Mnemonics are basically shortened two or three letter words. A sample program written in Assembly language.

MOV   AX,   15   (move 15 to register Ax)

SUB   AX,   10   (Subtract 10 from the value Ax)

Programs written in this language require an assembler to convert them into machine language.

Classes of high level languages

- Third generation languages (3GLS)
- Forth generation languages (5GLs)
- Fifth generation languages (5GL's)
- Object Oriented Programming Language (OOP)
- Web scripting languages

Third generation languages (3GLS)

- This generation languages is also called structured or procedural languages. A procedural language makes it possible to break a program into components called modules. Each performing a particular task. Structured programming has advantages because its flexible, easier to read and modify.

Examples of third generation programming language

- Pascal – was developed to help in the teaching and learning of structured programming.
- Fotran – was developed for mathematics, scientists of programs with mathematical expressions.
- Cobol – was designed for developing programs that solve business problems.
- Basic – developed to enable students to easily learn programming.
- C – used for developing system software e.g. the operation systems. Its very powerful high level language because of its ability to provide programmer with powerful aspects/features of low level.
- Ada – this language is suitable for developing military, industrial and real – time systems.
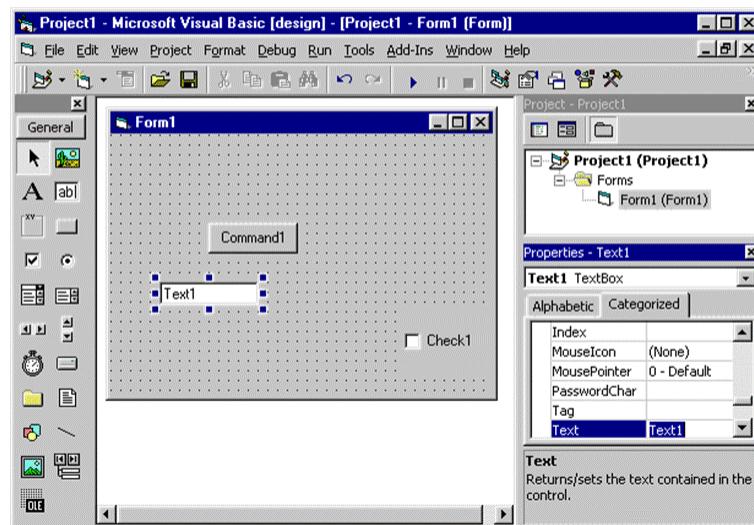
Forth generation languages (4GLs)

- This generation make programming an even easier task than the third generation language because they present the programmer with more programming tools. Examples of such tools are command buttons, forms etc. the 4 GLs are easy to learn and understand because they are user based. The languages syntax (grammar) is natural, near English language and use menus to prompts to guide a non-specialist or retrieve data with ease.

Examples of 4GLs

The Visual Basic programming environment

- Visual Basic

- Delphi Pascal

- Visual Cobol

- C++

Fifth generation languages (5GL's)

- These languages are designed around the concept of solving problems by enabling the computer to depict human like intelligence. These programs are designed to make the computer solve the problem for the programmer rather than programmer spending a lot of time to come up with the solution.

Examples of 5GL's

- PROLOG

- MERCURY

- LISP

- OCCAM

Object Oriented Programming Language (OOP)

- The concept behind OOP languages is to look at a program as having various objects interacting to make up a whole. Each object has a specific data values that are unique to it (called state) and a set of the things it can accomplish called (functions or behaviour). This process of having data and functions that operate on the data within an object is called **Encapsulation**. Several objects can then be linked together to form a complete program. OOP has greatly contributed to development of **Graphical user interface operation system** and **application programs.**

Examples of OOP

- Java

- Simula

- Small talk

Web scripting languages

- These languages are used to develop or add functionalities on web pages. Web pages are hypertext documents created in a language called **HyperText Markup Language** (HTML). The languages consist of markup tags that tell the Internet browser that the file contains HTML-code information and is distinguished by a file extension of HTML. The markup tags define the various components of a World Wide Web document such as heading, tables, paragraphs, lists etc. HTML does not have the declaration part and control structures, hence it's not considered as a true programming language. Due to its simplicity, it has many limitations and cannot be used alone when developing functional websites. Some special blocks of codes called **Scripts** may be inserted in HTML pages using scripting languages like JavaScript, VBScript etc in order to add functionality to HTML pages.

Examples of HTML tags and their meanings

Tag	Meaning
<HTML> </HTML>	Marks the beginning and end of a HTML document. All other tags and text fall between these two tags.
<HEAD></HEAD>	Marks the header part of the document.
<TITLE> </TITLE>	Gives title of te web page. Text between this tags appears in the title bar when the page is browsed.
<BODY></BODY>	Marks the body part of the document.
<CENTER></CENTER >	Centres text and objects on the web page.
<B></B>	Bolds the text on the web page.
<I></I>	Italicise the text.
<H1> </H1>	Sets size of text on the web page with H6 displaying the smallest and H1 the largest size.
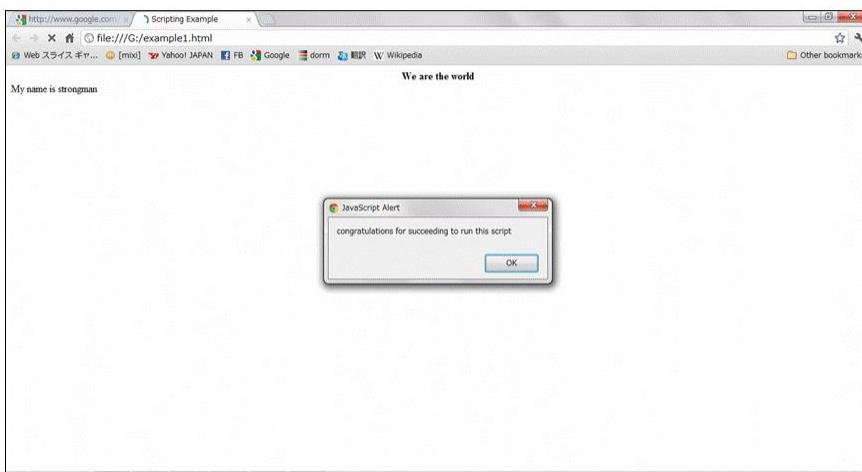
Creating a script using Javascript 1

- Javascript is a popular scripting language. Before writing your HTML program with a script inserted, make sure that you have the latest browser software installed on your computer. Older browsers may not have support for Javascript. If you are using Internet Explorer, it should be version 5.0 and above.

Creating a script using Javascript 2

- Open Notepad and key in the following program. Do not write the numbers to the left of each line of code.
  1. <HTML>
  2. <HEAD>
  3. <TITLE>Scripting Example</TITLE>
  4. </HEAD>
  5. <BODY>
  6. <HI><CENTER><B>we are the world</B></CENTER><HI>
  7. <SCRIPT>LANGUAGE='JavaScript'>
  8. document.write('My name is strongman');
  9. alert('congratulations for succeeding to run this script');
  10. </SCRIPT>
  11. </BODY></HTML>

Creating a script using Javascript 3

- After typing the entire program, save your file on the desktop as Example.html and then close the notepad. Notice that the icon to your file on the desktop look like that of the default web browser in your computer.

- To view the web page, double click the icon of the file Example.html on the desktop. Figure right shows as open webpage.

Explanations of JavaScript

Line1	The tag <HTML> marks the beginning of the HTML document.
Line2	The <TITLE></TITLE> tags show the title of the web page. The text between this tags will appear in the title bar of the running HTML document. Notice that the title is written in the header section i.e. between <HEAD> and </HEAD> tags.
Line5	It marks the beginning of the body section. Anything between <BODY> and </BODY> will be executed and displayed when the webpage starts running.
Line6	This line will display the text "We are the world" on the screen. The text will be large i.e. size H1 it will be centred on the screen. The text will also be bolded.
Line7	It marks the start point of the script. Notice the line 'LANGUAGE='JavaScript' which tells the browser that the script will be written in JavaScript language.
Line8	The statement document.write tells the browser using JavaScript to write whatever is in the brackets. Notice that in JavaScript, the end of a statement is marked by a semicolon(;).
Line9	The word alert displays a message box on the screen with an OK button. Clicking the button makes the dialog box to disappear. The text in the brackets appears in the dialog box.
Line10	Closes the script.
Line11	Marks the end of the body and the HTML code.

Advantages of low-level languages

- The CPU understands machine language directly without translation.
- They are stable and hardly crash or breakdown once written.
- Running a program is fast, no compilation is needed.
- They are economical in terms of the amount of memory they use.

Disadvantages of low-level languages

- They are difficult and cumbersome to use and learn.
- Require highly trained experts to both develop and maintain programs.
- Debugging programs is difficult.
- They are machine dependent.
- The programs are long.

Advantages of high-level languages

- The programs are portable (not machine dependent).
- They are user friendly and easy to use and learn.
- They are more flexible.
- They are easy to debug.
- They provide better documentation.
- Require less time to code.
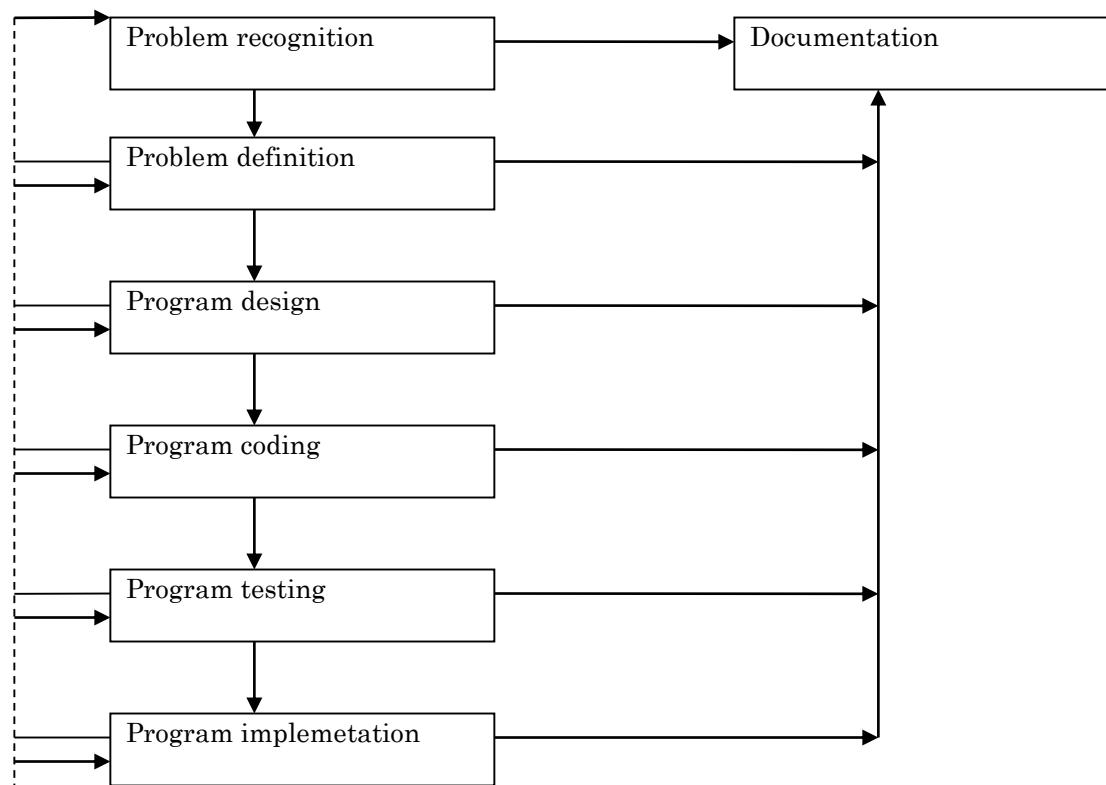
Disadvantages of high-level languages

- Program executed more slowly.

- Require larger CPU storage capacity for compilation.

- They have to be translated to machine readable form before

Program development

- There are six stages of program development. They include:

Problem recognition		Documentation
Problem definition		
Program design		
Program coding		
Program testing		
Program implemetation		

Problem recognition

- This refers to the understanding and interpretation of a particular problem. To understand a problem one has to look for key words such as compute, evaluate, compare etc. a programmer identifies problems in the environment and seeks to solve them by writing a computer program that would provide the solution.

Circumstances that can cause the programmer to identify a problem

- Opportunity to improve the current program.

- A new directive given by the management requiring a change in the status quo.

- Problems or undesirable situations that prevent an individual or organisation from achieving their purpose.

Problem definition

- At this stage the programmer tries to determine or define the likely input, processing and expected output using the keywords outlined at the problem recognition stage. The boundaries of the expected program are established and if several methods to solve the same problem are identified the best alternatives should be chosen. At the end of the stage requirement documentation for the new program is written.
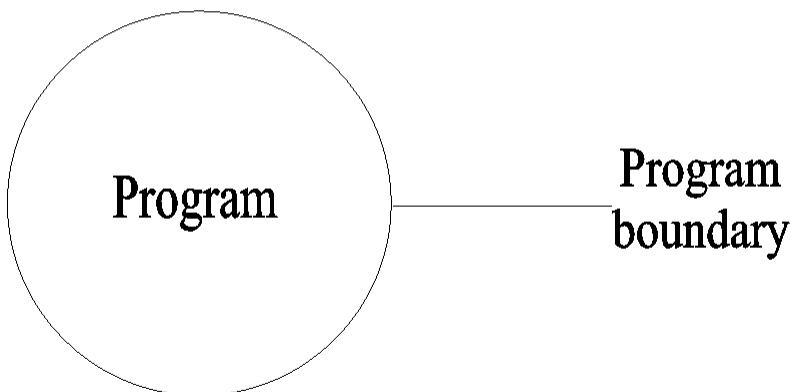
Program design
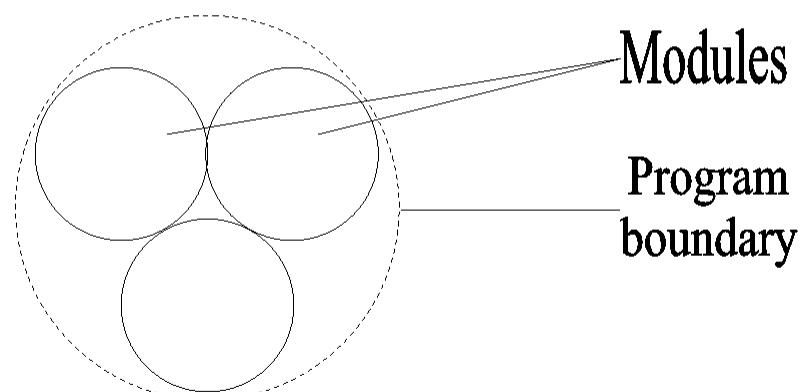
- This is the actual development of the program's processing or problem solving logic called **algorithm.** (A limited number of logical steps that a program follows in order to solve a problem). The programmer comes up with an algorithm after analysing the requirements specifications. Some of the programs are made up of one large block code. i.e. they are Monolithic while others are made of several units called modules which work together to from the whole program. In modular programming, each module performs a specific task. This approach makes a program flexible, easier to read and debug. This phase enable the programmer to come up with models of the expected program. The model show the flow of events and data throughout the entire program from input of a program.

Monolithic and modular program

A non-modular program (monolithic)

A modular program

Program

Program boundary

Modules

Program boundary

Program coding

- This is the actual process of converting a design model into its equivalent program. This is done by creating the program using a particular programming language. The end result of this stage is a source program that can be translated into machine readable form for the computer to execute and solve the target problem.

Program in Pascal and C++

Program in Pascal	Program in C++

Program AreaCircle (input, output);
Const Pi = 22/7;
Var
Radius, Area: real;
Begin
WriteIn ('Enter the radius');
ReadIn (radius);
Area: = Pi *Radius *Radius;
WriteIn ('The area is', Area)
End.
	#include<iostream.h>
	double radius, area;
	double pi = 22/7;
	main ()
	{
	cout<<'Enter the radius';
	cin>>radius;
	area = pi *radius *radius
	cout<<'The area is', area;
	return 0;
	}

Explain the program codes 1

Pascal code	C++ code	Explanation
Program Areacircle (input, output);	#include<iostream.h>	The header of the programs. The statements in () and <> shows that the user inputs data via the keyboard and the program display information on the screen.
Const Pi = 3.142;	double pi = 3.142;	A constant has been declared with a name pi and value 3.142.
Var Radius, area:real	double area, radius;	Real variables with fractional part have been declared.

Explain the program codes 2

Pascal code	C++ code	Explanation
Begin	{	Marks the beginning of the program body or executable statements.
WtiteIn	cout<<'Enter radius':	Display on the screen the string between inverted commas.
ReadIn (Radius)	cin>>radius;	Displays a blinking cursor that tells the user that an input is needed before the program can continue.

Explain the program codes 3

Pascal code	C++ code	Explanation
Area: = Pi * Radius * Radius	area = pi * radius * radius;	Calculates the area. Notice the assignment statement in Pascal is : = while in C++ it is =
Writeln ('The area is', Area);	cout<<"The area is', Area<<"\n";	Display the value stored in the variable Area.
End.	return 0; }	Marks the end of the program

Program Testing and Debugging

- After coding the program has to be tested and the errors detected and corrected. Debugging refers to detection and correction of errors that may exist in the program. Program testing involves creating test data designed to produce predictable output. They are two types of errors (bugs) that can be encountered when testing.

Types of errors (bugs)

- Syntax errors – they occur as a result of improper use of language rules. E.g. grammar mistakes, punctuation, improper naming of variables etc. these errors are detectable by translator and must be corrected before the program runs.

- Logical errors – they are not detectable by the translator. The program rules but gives wrong out put or halts during execution.

Methods of errors detection

- Desk checking / dry – run.
- Using Debugging utilities.
- Using Test Data.
- Implementation and maintenance Implementation.
- Review and maintenance.
- Program documentation.

Desk checking / dry – run

- It involves going through the program while still on paper before entering it in the program editor.

Using Debugging utilities

- In the program editor, you can run the debugging utilities during translation to detect syntax errors.

Using Test Data

- The programmer carries out trial runs of the new program. At each run he enters various data variations and extremes including data with errors to test whether the system will grid to a halt. A good program should not crash due to incorrect data entry but should inform the user about the anomaly.

Implementation and maintenance implementation

- This is the actual delivery and installation of the new program ready for use, creating data files and train people to use the system. The new system will change the way things are done hence it should be reviewed and maintained.

Review and maintenance

- This stage is important because of the errors that may be encountered after implementation. A program may fail due to poor use, hence proper training and post implementation support of users will reduce chances of having them entering invalid data that crash the program.

Program documentation

- This is writing of support materials explaining how the program can be used by users, installed by operators or modified by other programmers. All stages of program development should be documented in order to help during future modification of the program.

Types of documentation

- User oriented documentation – these type enables the user to learn how to use the program as quickly as possible and with little help from programmer.

- Operator oriented documentation – meant for computer operators. E.g. Technician. It helps them to install and maintain the program.

- Programmer oriented documentation – written for skilled programmers. It provides necessary technician information to help in future modification of program.

Development of algorithms

- **Algorithms** – refers to a limited number of logical steps that a program follows in order to solve a problem.

- **Pseudocode** – refers to a set of statements written in a readable language (English – like) but expressing the processing logic of a program.

Guidelines for designing a good pseudocode

- The statements must be short, clear and readable.
- Pseudocode lines should be clearly outlined and indented clearly.
- It should show clearly the start and stop of executable statements and control structures.
- Statements must not have more than one meaning.

Examples of Pseudo code

Solution

START

Print "Enter two numbers"

Input x,y

Sum – x + y

Average = sum / 2

PRINT sum

PRINT Average

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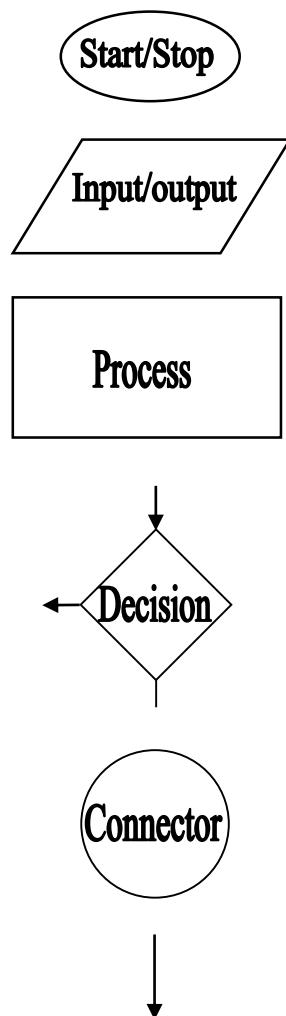
- Write of Pseudo code that can be used to prompt the user to enter two numbers, calculate the sum and average of the two numbers and then display the output on the screen

Program flowcharts

- A flowchart is a diagrammatic representation of a program in algorithms. It uses statements and symbols that have specific meaning. The symbols are of different standard shapes linked to show the order of processing. Each shape contains note stating what the operation is.

Most common program flowcharts

- **Ellipse:** denotes the beginning and end of the program algorithm.
- **Parallelogram:** used to denote an input or output operation. For example, READ A,B, PRINT SUM =A+B
- **Rectangle:** indicates that a processing or data transformation is taking place. For example SUM =A+B.
- **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.
- **Connector:** used as a connecting point or interface for arrows coming from different directions.
- **Arrow:** used to indicate the direction of flow of the program logic.
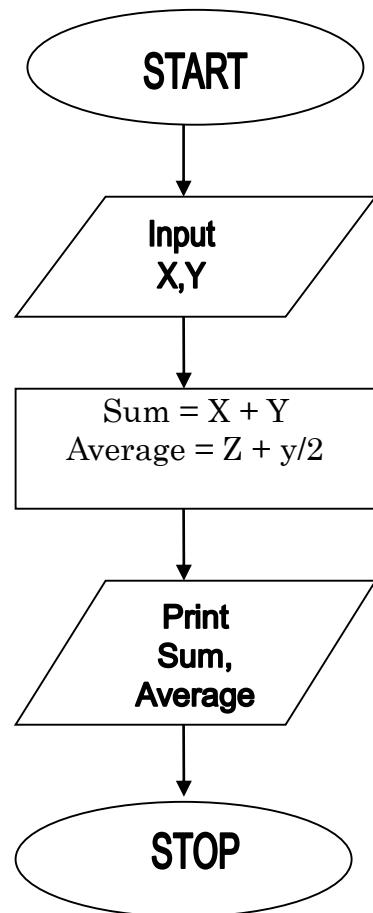
Guidelines for drawing a flowchart

- There should be only one entry and one exit point of a program algorithm.
- Use correct symbol at each stage in the flowchart.
- Avoid crossed flow lines.
- Be as neat and tidy in drawing as possible.
- General direction of flow in any flowchart is from top to bottom, left to right.

Examples of flowchart

Solution

```
        ┌─────────────┐
        │    START    │
        └─────────────┘
               │
               ▼
          ╱─────────╲
         ╱  Input    ╲
         ╲   X,Y     ╱
          ╲─────────╱
               │
               ▼
      ┌──────────────────┐
      │   Sum = X + Y    │
      │ Average = Z + y/2│
      └──────────────────┘
               │
               ▼
          ╱─────────╲
         ╱   Print   ╲
         ╲   Sum,    ╱
          ╲ Average ╱
               │
               ▼
        ┌─────────────┐
        │    STOP     │
        └─────────────┘
```

■ 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.

Types of flowchart

- System flowchart – it's a chart that depicts the systems as a whole with only subsystems or major elements shown,

- Program flowchart – this chart shows the sequence of operations as carried out by a computer program.

Advantages of flowchart

- Gives programmer good visual reference of what the program will do.
- Serves as a program or system documentation.
- It's easy to trace through from the start to find the action resulting from a set of condition.
- Allows programmers to test alternative solutions to a problem without over coding the program.

Disadvantages of flowchart

- There are so many different ways to draw them.
- it's difficult to remain neat and uncluttered if the logic is complex.
- Constructing a flowchart is time consuming.
- They take up considerable space.
- They are difficult to amend without redrawing.
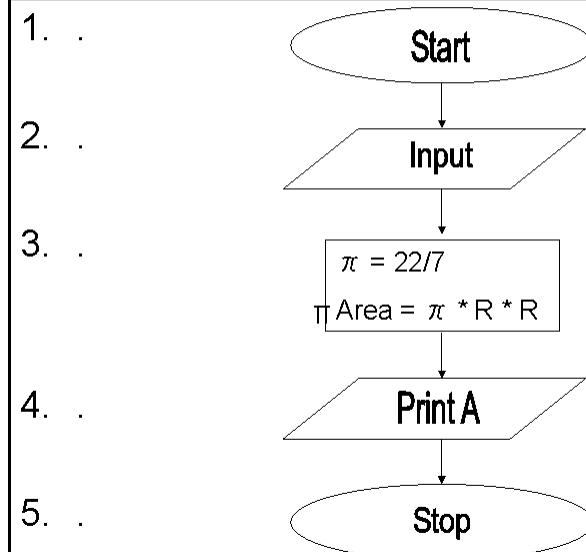
Comparison between a pseudocode and a flowchart

Method 2: Pseudocode
Begin
Set π to 22/7
Prompt the user for the radius (R)
Store the radius in a variable (R)
Set area (A) to π * R * R
Store the area in a variable (A)
Print A
Stop

Method 1: Flowchart
1. .
2. .
3. .
4. .
5. .

Program control structures

- They are blocks of statements that determine how statements are to be executed. There are three control structures namely.

  ☐ Sequence
  ☐ Selection decision
  ☐ Iteration (looping) repetition

Sequence

- In this control structure, the computer reads instructions from a program file starting from the first top line and proceeding downwards one by one to the end. Hence sequential program execution enables the computer to perform tasks that are arranged consecutively one after another in the code.

Example of how a sequential programs code execute

The program file reader reads

Begin   {Procedure name}   sequentially

Action 1   statement

Action 2   by

Action n   statements

End   {Procedure name}   to the end

of the file

Selection/decision

■ This structure is used to branch, depending on whether the condition returns a value of True of False (Yes or No).

For example
    IF   <   Condition   >
        Then Action 1
        Else   Action 2
    Endif.

■ There are 4 types of selection controls used in high-level programming.
    □ IF……….THEN
    □ IF……THEN……ELSE
    □ Nested IF
    □ CASE SELECTION

IF………THEN

General format

IF < condition > Then

Statement:

EndIf

If Mark > = 80 then

Print "Reward"

EndIf

- This is used if only one option is available. All other options are ignored. For example if a school wants to reward only the students who have mean mark of 80 and above, it will be reward only those students who have attained 80% and above and ignore the rest.

IF………THEN………ELSE

General Format

IF < Condition > THEN

    Statements 1

ELSE

    Statements 2

EndIf

The algorithem will be:-

IF fault = Serious THEN

    Print "RED CARD"

ELSE

    Print "yellow Card"

EndIf

- Used when there are two available options. For example in a football match a player is given a RED CARD if he does a very serious mistake otherwise he is given a Yellow Card.

Nested IF

- This is used where two or more options have to be considered to make a selection. For example, to award grades according to the marks as follows

  a. 80 marks   Grade A
  b. 60 marks   Grade B
  c. 50 marks   Grade C
  d. 40 marks   Grade D

General format of Nested IF

General Format

IF <Conditions> Then
Statement
ELSE
IF <Condition> Then
Statement
ELSE
IF <Condition>Then
Statement
Endif
Endif
End

The pseudo code segment will be:-

If Marks = 80 THEN
Grade = "A"
ELSE
If Marks = 60 THEN
Grade = "B"
ELSE
If marks = 50 THEN
Grade = "D"
Endif
Endif
Endif
Endif
End

CASE SELECTION

General format
```
CASE x   of
    Label 1 : Statement1
    Label 2 : Statement2
    Label n : Statement n-1
ELSE
    Statement n
End case
```

Example:-
```
CASE            average OF
80…100:         Grade="A"
70…79:          Grade="B"
60…69:          Grade="C"
40…49:          Grade="E"
ELSE

                Grade="F"

End Case
```

■ It's an alternative to the nested IF. This selection is preferred to the Nested if in order to reduce the many line s of codes. Case selection can only be expressed using integers and alphabetic characters only. The Boolean expression should be CASE integer OF or CASW char OF

Iteration (looping) repetition

- This is designed to execute the same block of code again and again until a certain condition is fulfilled. Iteration is important in situations where the same operation has to be carried out on a set of data many times. There are three main looping controls.

  - ☐ THE-WHILE-DO LOOP
  - ☐ REPEAT……….UNTIL LOOP
  - ☐ The FOR LOOP

THE-WHILE-DO LOOP
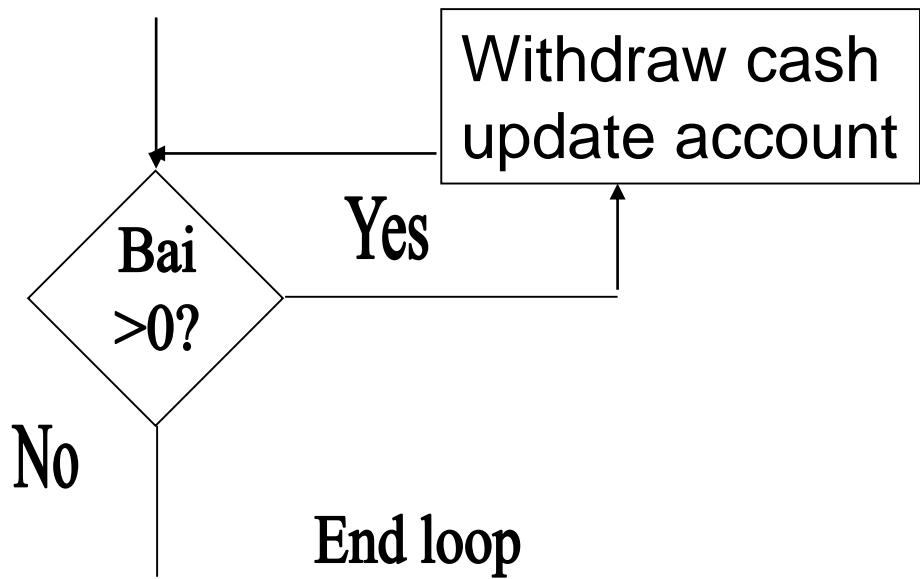
- This repetitive structure tests the condition first before executing the successful code if and only if the Boolean expression returns a true value.

Example

To withdraw money using an ATM a customer must have a balance in his/her account.

General format of THE-WHILE DO LOOP

Flowchart

Withdraw cash update account

Bai >0?

Yes

No

End loop

General format

While <condition> Do

Statement

End while

Pseudo code segment

While   balance >0 do

Withdraw cash

Update account

End while

REPEAT………UNTIL LOOP

■ In this structure the code is executed before testing the condition. The repeat loop stops when the Boolean statement returns a value. For example in the case of ATM discussed above the client can withdraw money until balance is zero.

Example
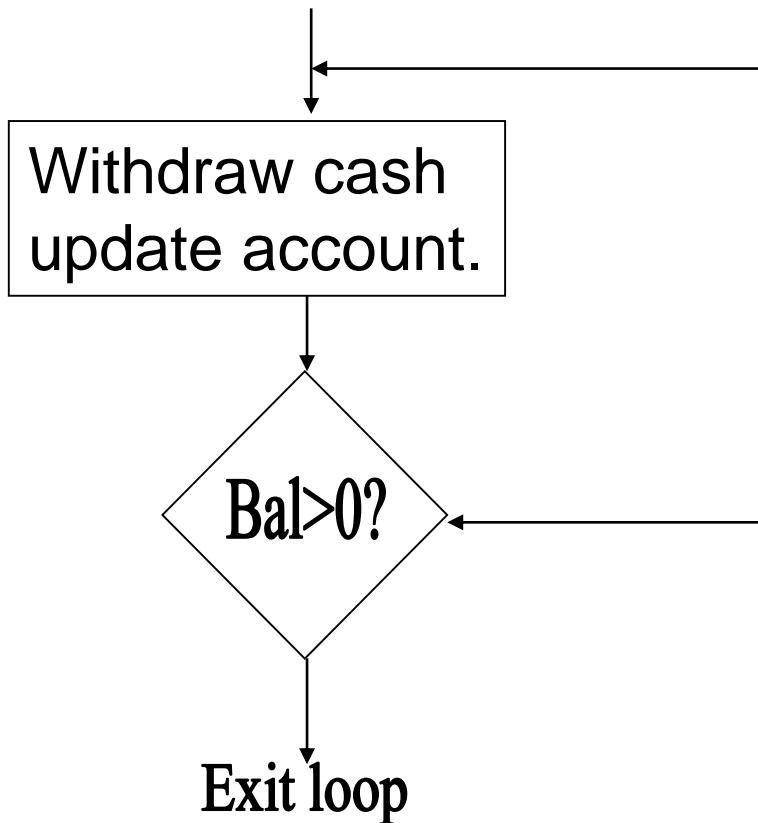
Pseudo code segment.

REPEAT

Withdraw cash

Update account

Until balance ≤ 0;

General format of REPEAT…………UNTIL LOOP

Flow chart

Withdraw cash update account.

Bal>0?

Exit loop

General format

REPEAT

Statement

UNTIL <condition>

The FOR LOOP

- This structure is used where execution of the chosen statements has to be repeated a predetermined number of times. For example if a program is to calculate the sum of ten numbers provided by the user. The FOR LOOP can be used to prompt the user to enter the 10 numbers at most ten times. Once the numbers are entered the program calculates and displays the sum.

Pseudo code of the FOR LOOP

Pseudo code

FOR   count = 1 to 10 DO

  writeln "Enter a number (N)"

Readln   N

     Sum = sum + N

  End FOR

  Display Sum

The counter has to be set to a start value and sometimes to an end value.

General format of the FOR LOOP

General format

1. format for the FOR LOOP that counts from lower limit.
   For loop variables = lower limit To upper limit Do
   Statements
   EndFor

2. Format for the 'For' loop that counts from upper limit down to lower limit
   For loop variable = Upper limit Down To Lower Limit Do
   Statements
   EndFor
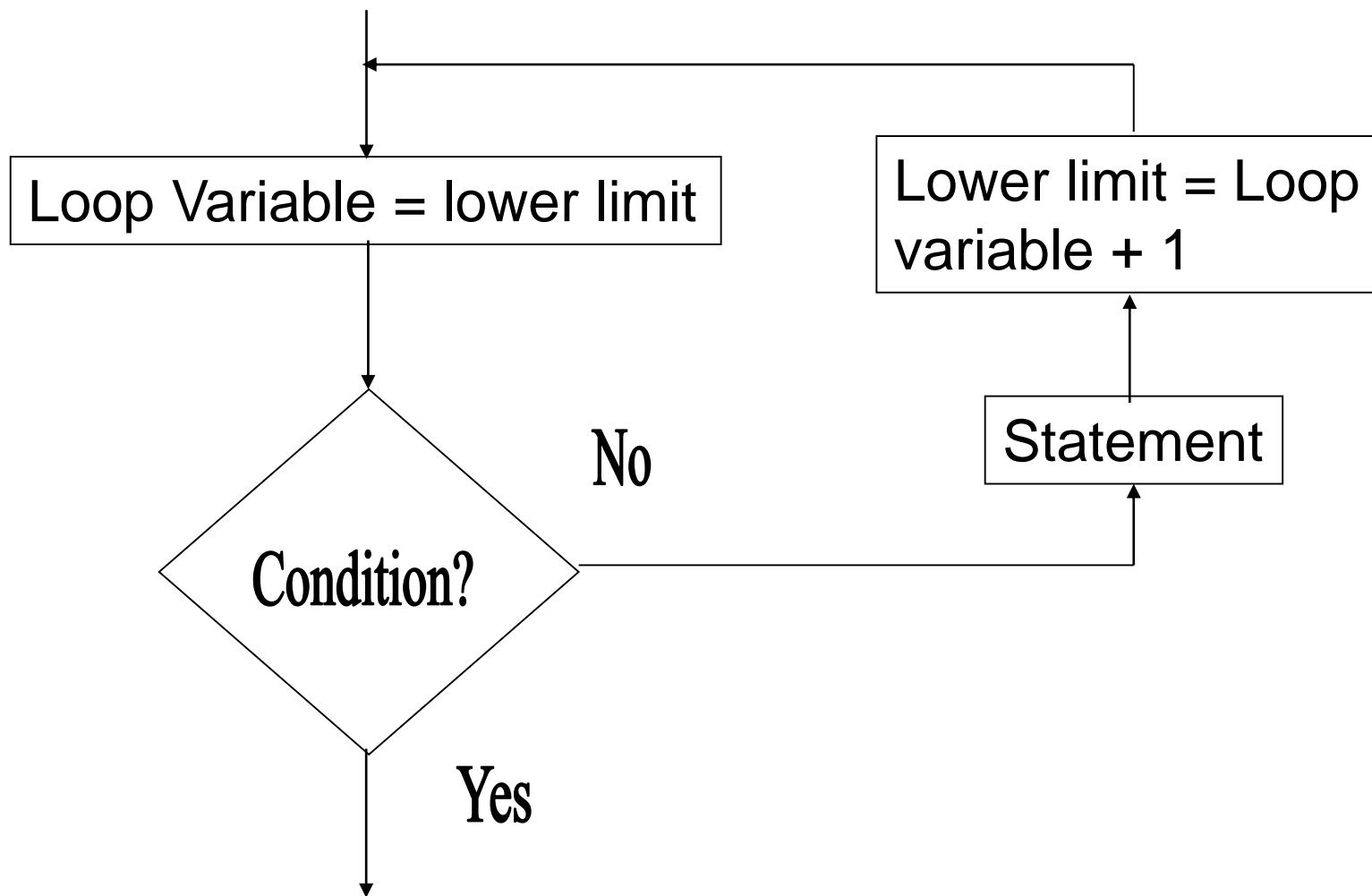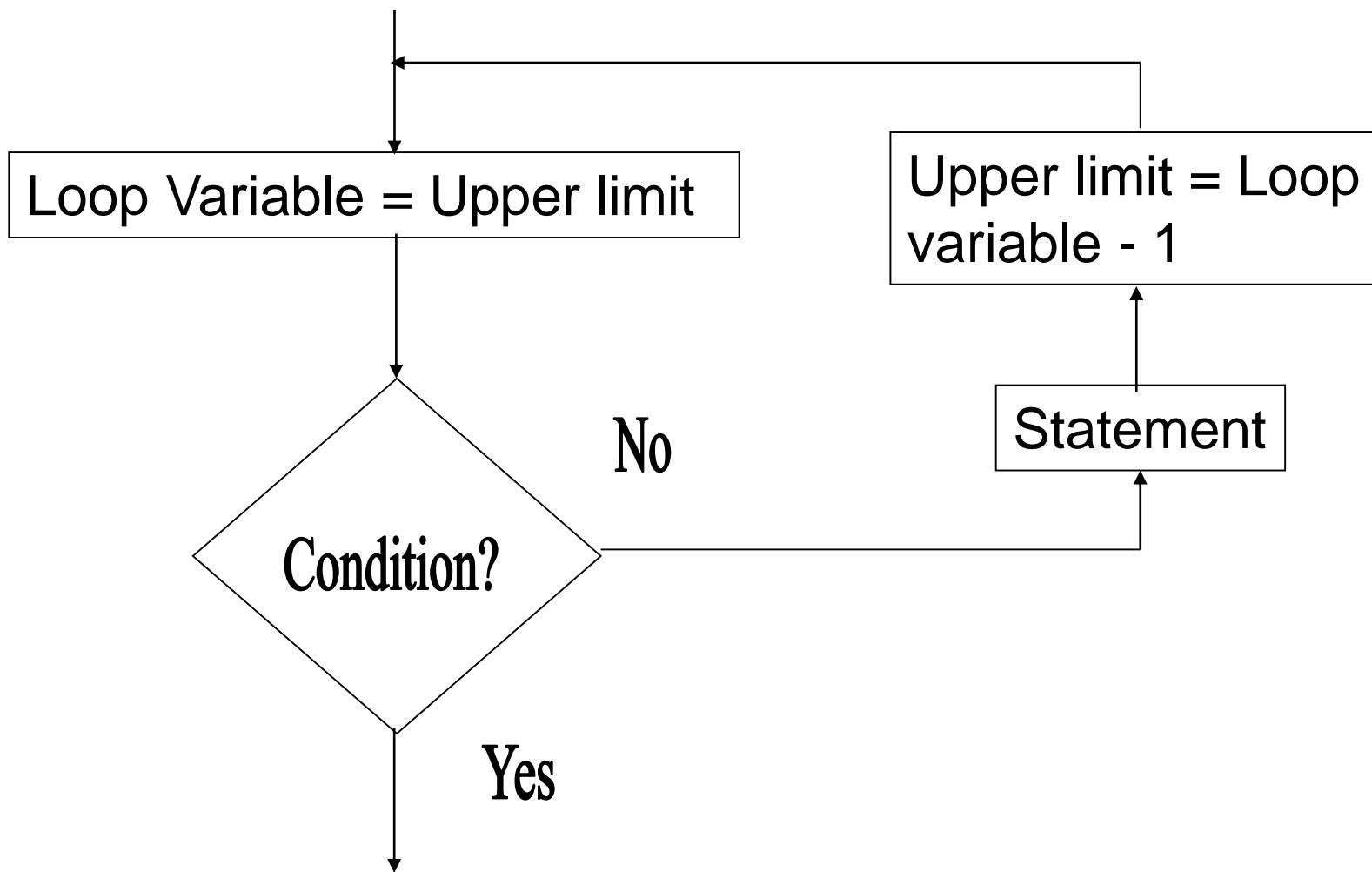
Flowchart for a For Loop that counts upward

Flowchart for a For Loop that counts downwards

Examples of complex pseudo codes 1

- Ushirika Society pays 5% interest on shares exceeding Kshs 10,000 and 3% on shares that do not meet this target. However no interest is paid on deposits in the member's bank account. Design a pseudo code for a program that would:
  a. Prompt the user for shares and deposit of a particular member.
  b. Calculate the interest and total savings.
  c. Display the interest and total savings on the screen for a particular member.

Pseudo code of examples 1

Pseudo code

Start

Input Name, share, Deposit

If share > 10000 THEN

Interest = 0.05*shares

ELSE

Interest=0.03*shares
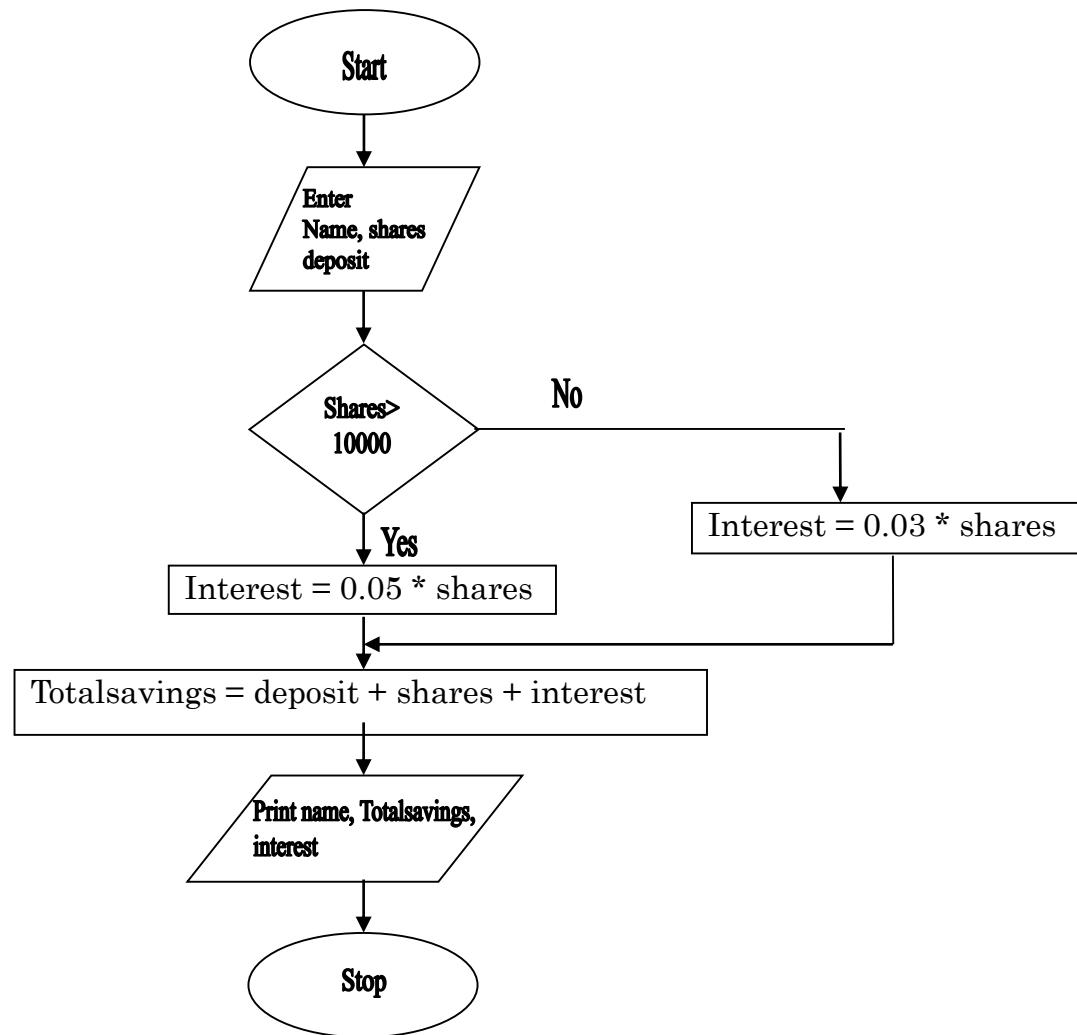
Endif

Total savings = deposits + Interest + shares

Print Name, Totalsaving, Interest

Stop

Flowchart of example 1
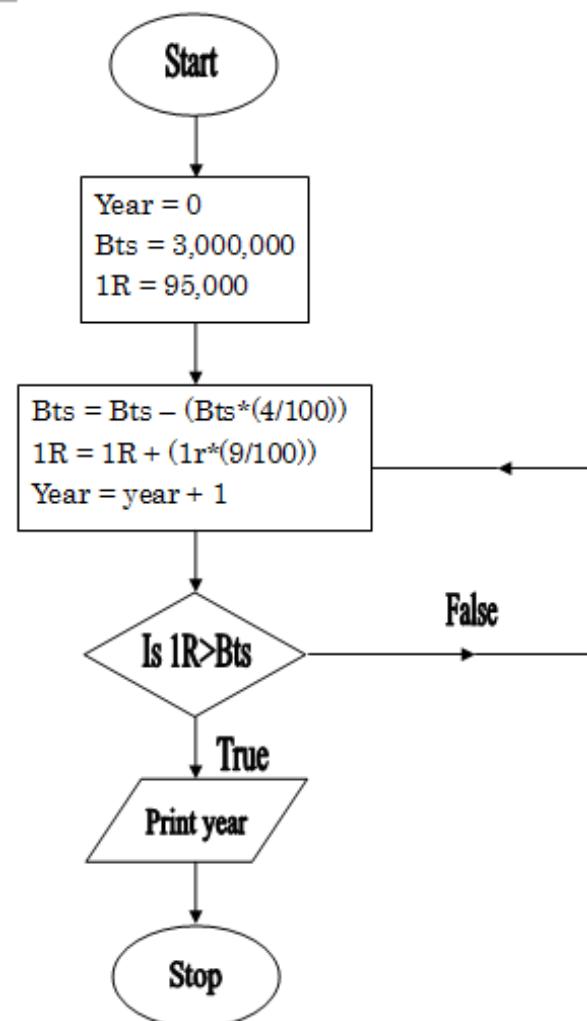
Examples of complex pseudo codes 2

- Botswana has a population of 3,000,000 but this is falling by 4% each year. The island of Reunion has a population of 95,000 but this is increasing by 9% each year.

    a. Draw a flowchart that predicts the year in which the population of reunion will be greater than that of Botswana if the trend continues.

Solution of example 2

Pseudo code of example 2

Pseudo code

Start

Bts: = 3,000,000

1R: = 95,000

Year: = 0

REPEAT

   Bts = Bts - (Bts * (4 / 100))

1R = 1R + (1R * (9 / 100))

Year = year + 1

UNTIL 1R > Bts

Print year

Stop

Examples of complex pseudo codes 3

- Write a program that will allow the input of name of student marks obtained in 5 subjects (maths, English, Kiswahili, computer, biology). The program should calculate the total and average marks for each student and assign the grades depending on the average marks obtained as follows.

  80 – 100   A
  70 – 79   B
  60 – 69   C
  50 – 59   D
  Below 50   E

  The program should then display each student's name, total marks and average grade.

Pseudo code of example 3

Pseudo code

START

REPEAT

Print "Enter name and subject marks"

Input Name, maths, English, Kiswahili, Computer, Biology

Sum =

Maths + English + Kiswahili + Computer + Biology

AVG = sum / 5

If (AVG$\geq$80) AND (AVG$\leq$100) THEN

Grade = 'A'

If (AVG$\geq$70) AND (AVG$\leq$79) THEN

Grade = 'B"

If (AVG$\geq$60) AND (AVG$\leq$69) THEN

Grade = 'C'

If (AVG$\geq$50) AND (AVG$\leq$59) THEN

Grade = 'D'

ELSE

Grade = 'E"

Endif

Endif

Endif
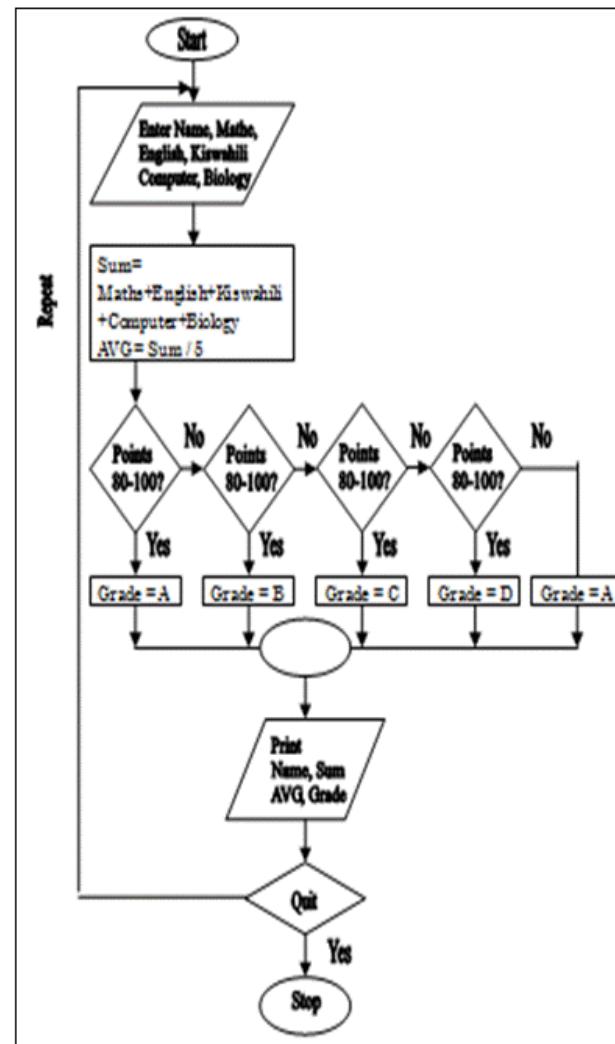
Endif

Print Name, Sum, AVG, Grade

Until count = Number of students

Stop

Flowchart of example 3
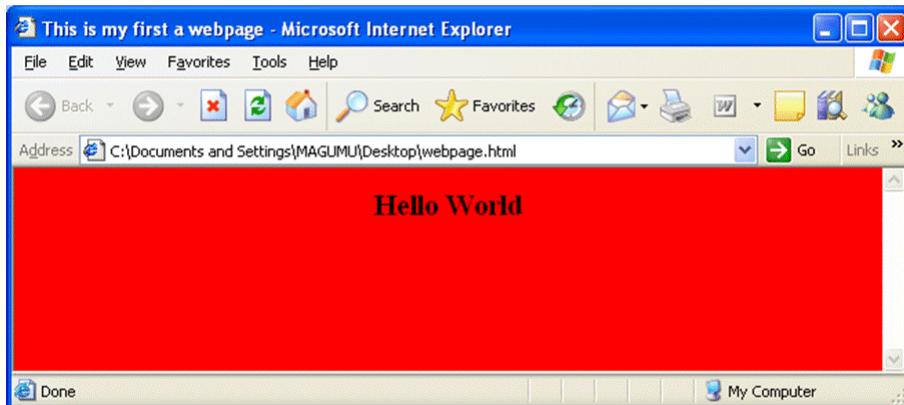
Practical activity 1-1

1. Open a text editor program on your computer like NotePad or WordPad.
2. Type the following program exactly the way it is in the editor.

```
<HTML>
<HEAD><TITLE>This is my first a webpage</TITLE></HEAD>
<BODY bgcolor = "red">
<H1><CENTER><B>Hello World</B></CENTER></H1>
</BODY>
</HTML>
```

3. Save your work as webpage.html on the desktop. Make sure that the *Save As Type* box reads "All Files" before clicking the save button in order to avoid saving a text file with two extensions i.e. webpage.html.txt.

Practical activity 1-2

4. Close your text editor. Notice that your file on the desktop has the icon of the default web browser installed on your computer. Double click it to view the web page! Figure right shows a sample of the display expected when the file is loaded to the browser. If you have a colour monitor, it should look as below only that it will have black bold text on a red background!

5. Check your program and change your background to blue, save then click the refresh button. What happens?

Review questions 1

1. Define the term computer program.
2. What is programming?
3. State three advantages of high level languages over low level languages.
4. List four examples of languages and for each, state its most appropriate application area.
5. Why is an executable file unique when compared to any other file?
6. Differentiate between a compiler and an interpreter. Why did early computers work well with interpreters?
7. List the various examples of programming languages per generation.
8. State one advantage of machine language over the other languages.
9. Write the following in full
   A) HTML
   B) OOP
10. Distinguish between source program and object code in programming.
11. Define the term encapsulation as used in object oriented programming.

Answer of review questions 1-1

1. A computer program is a set of instructions that directs a computer on how to process a particular task.
2. Programming is the writing of computer programs.
3. A). Easier to learn and use.
   B). Portable.
   C). Flexible.
   D). Easier to correct errors.
4. A). FORTRAN – used for developing business oriented programs.
   B). COBOL – designed for developing business oriented programs.
   C). Pascal – mostly used as a teaching tool in structured programming.
   D). C – mostly used to create system software such as operating systems.
   E). Ada – suitable for developing real-time and industrial systems.
5. It is the main file that is used to start the program.
6. A complier translates the entire source program into machine readable form while an interpreter translates the source program to machine language line-by-line. Interpreters were used on early computers due to memory limitations.

Answer of review questions 1-2

7. A). First generation – machine languages.

   B). Second generation – assembly languages.

   C). Third generation – Pascal, C, COBOL, BASIC, FOTRAN, Ada etc.

   D). Fourth generation – Visual Basic, Delphi Pascal, Visual COBOL etc.

   E). Fifth generation languages – PROLOG, Mercury, LISP, OCCAM etc.

8. A). The processor executes them faster.

   B). They are more stable.

9. A). HTML – Hypertext Markup Language.

   B). OOP – Object Oriented Programming language

10. Source program refers to a program that is not yet translated while object code refers to the program code that is in machine readable form.

11. Encapsulation refers to the process of joining data and instruction together to form an object.

Review questions 2

1. Give one advantage of compiling a program rather than interpreting it.
2. Outline at least six stages of program development in their respective order.
3. Highlight two advantages of monolithic programs.
4. State two advantages of modular programming.
5. In what stage of the development does program documentation fall? Justify your answer.
6. Differentiate between a flowchart and pseudocode.
7. What is a program bug?
8. Explain why it is important to test a program before implementing it.

Answer of review questions 2-1

1. A). Interpreters translate the source program one statement at a time while compilers translate the entire source code at once before execution.

   B). Interpreters translate the program each time it runs hence slower than compiling. Compiled programs can be saved on a storage media. This means that it does not require any further transaction every time the program is run hence executes faster than interpreted programs.

   C). Interpreters translation takes less memory while compiled programs require more memory as the object code files are larger.

2. A). Problem recognition.

   B). Problem definition.

   C). Program design.

   D). Program coding.

   E). Program testing and debugging.

   F). Implementation.

Answer of review questions 2-1

3. A). They are difficult to test and debug.
   B). They are difficult to modify or upgrade.
4. A). They are easy to test and debug.
   B). They are easy to modify and upgrade.
5. Documentation is done at all stages of program development.
6. A flowchart is a graphical representation of step-by-step processing logic of a program while a pseudocode is a set of structured-English like statements that describe the step-by-step processing logic of a program.
7. Bugs refers to program errors.
8. To correct any syntax or logical errors.

Review questions 3

1. State the factors you would consider when
    A) Writing a pseudocode
    B) Drawing a flowchart
2. Using illustrations, explain at least six symbols used in flowchart design.
3. Give one advantage of pseudocodes over flowcharts.
4. Write a pseudocode for a program that would be used to solve the equation

    $E = MC^2$

5. Draw a flowchart that would be used to classify animals according to sex. If a letter M is input the program should display 'Male' otherwise it should display 'Female'.

Answer of review questions 3-1

1. A). Pseudocode

   1). Statements must be short and clear.

   2). Statements should be unambiguous.

   3). Should shoe start and stop.
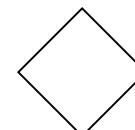
   4). Words used should be suggestive such as READ, PRINT, OUTPUT etc.

   B). Flowchart
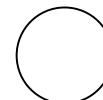
   1). A flowchart should have one entry and one exit point.

   2). Use correct symbols.
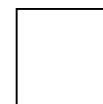
   3). The flow should be clearly indicated using arrows.
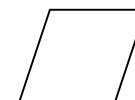
2. ◇ Decision – used to represent a decision construct.

   ○ Connector – connects logic flows.

   ▢ Process – indicates a transformation of data.
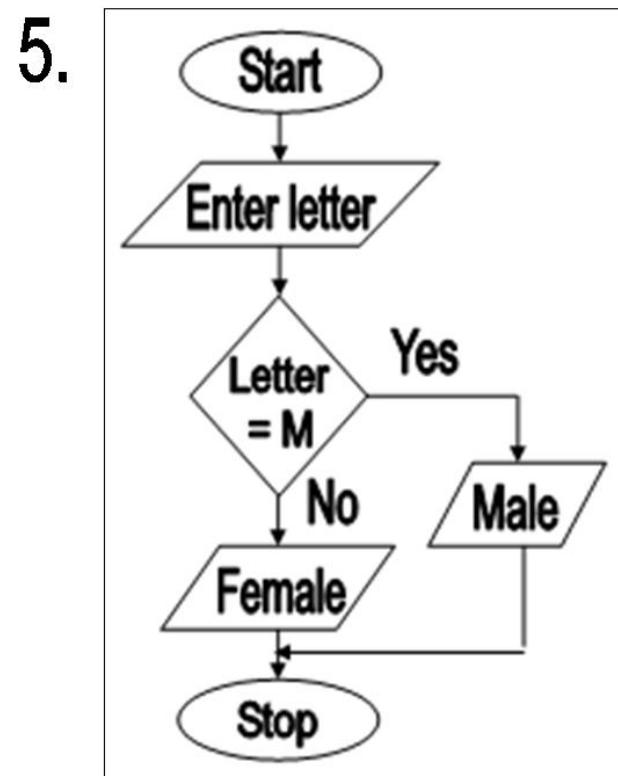
   ▱ Input/output – shows program input/output.

   ⬭ Ellipse – indicates start/stop of algorithm.

Answer of review questions 3-2

3. Pseudocodes are easier to write and understand.

4. Start

   Enter values form M and C

   E=M*C*C

   Display E

   Stop

5.

Review questions 4

1. Define the term selection in relation to program control structures.

2. State three types of control structures used in programming.

3. Differentiate between looping and selection.

4. In what circumstance does looping depend on decision?

5. List four selection controls used in writing a program.

Answer of review questions 4

1. Selection refers to a situation where further program execution depend one making a decision from two more alternatives.

2. Sequence, decision and iteration.

3. Looping refers to a situation where the same block of code is executed repeatedly until a certain condition is fulfilled, while selection refers to a situation where further program execution depends on making a decision from two or more alternatives.

4. Looping may depend on decision if a condition has to be met before the looping terminates or continues.

5. A). IF…THEN

   B). IF….THEN…ELSE

   C). Nested IF

   D). Case

Revision questions 1

1. State three reasons why high level languages are better than low level languages.
2. Explain why HTML is not true programming language.
3. List three examples of forth generation programming languages.
4. Describe the term structured programming.
5. Define the term structured programming.
6. A). State two reasons for documenting all the stages of program development.

   B). Give three types of program documentation in reference to programming.
7. Write HTML code that would display a green background, with the words 'Hello' on the screen.
8. Draw a flowchart showing the general flow of the following:
   A) The nested IF selection.
   B) The FOR loop.
   C) REPEAT….UNTIL loop.

Answer of revision questions 1-1

1. A). They are portable
   B). They are user friendly
   C). They are more flexible
   D). They are easier to debug
2. HTML is not a true programming language because it does not have a true format of a programming language. For example, it does not have a declaration part and control structures.
3. A). Visual Basic
   B). Delphi Pascal
   C). Visual COBOL
4. Fifth generation programming languages are designed in a way that enables programmers design programs that emulate human intelligence.
5. In structured programming, a program is broken down into components called modules each performing a particular task.
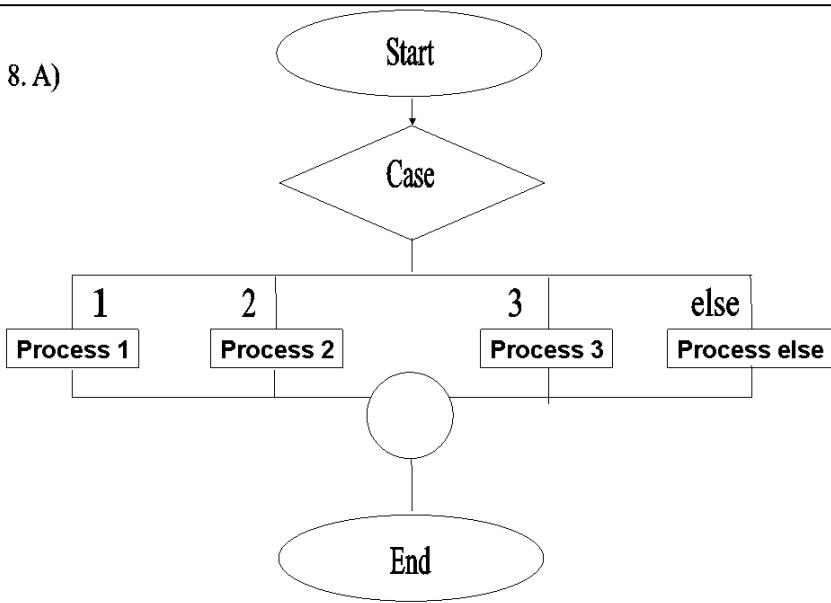
Answer of revision questions 1-2
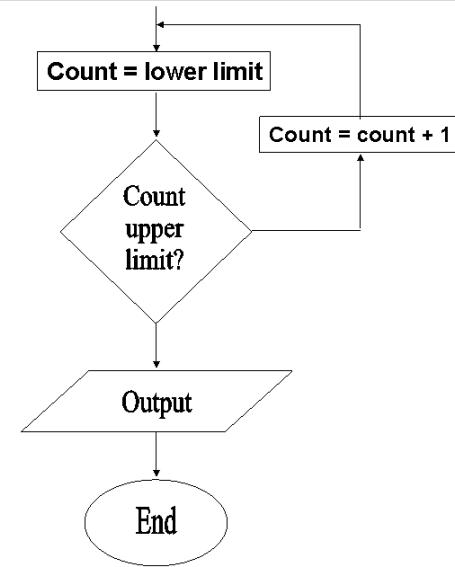
6. A). Reasons for documenting
   i. There may rise a need to revise or modify it.
   ii. Other people may think of ways of improving your program by studying at the documentation.

   B). Types of documentation
   i. User-oriented.
   ii. Operator-oriented.
   iii. Programmer-oriented.

7. <HTML>
   <HEAD><TITLE>Hello program </TITLE></HEAD>
   <BODY bgcolour = "blue">
   <H3><CENTRE><B>Hello</B></CENTER></H3>
   </BODY>
   </HTML>

Answer of revision questions 1-3

8. A)

Start

Case

1 — Process 1

2 — Process 2

3 — Process 3

else — Process else

End

8. B)

Count = lower limit

Count = count + 1

Count upper limit?

Output

End
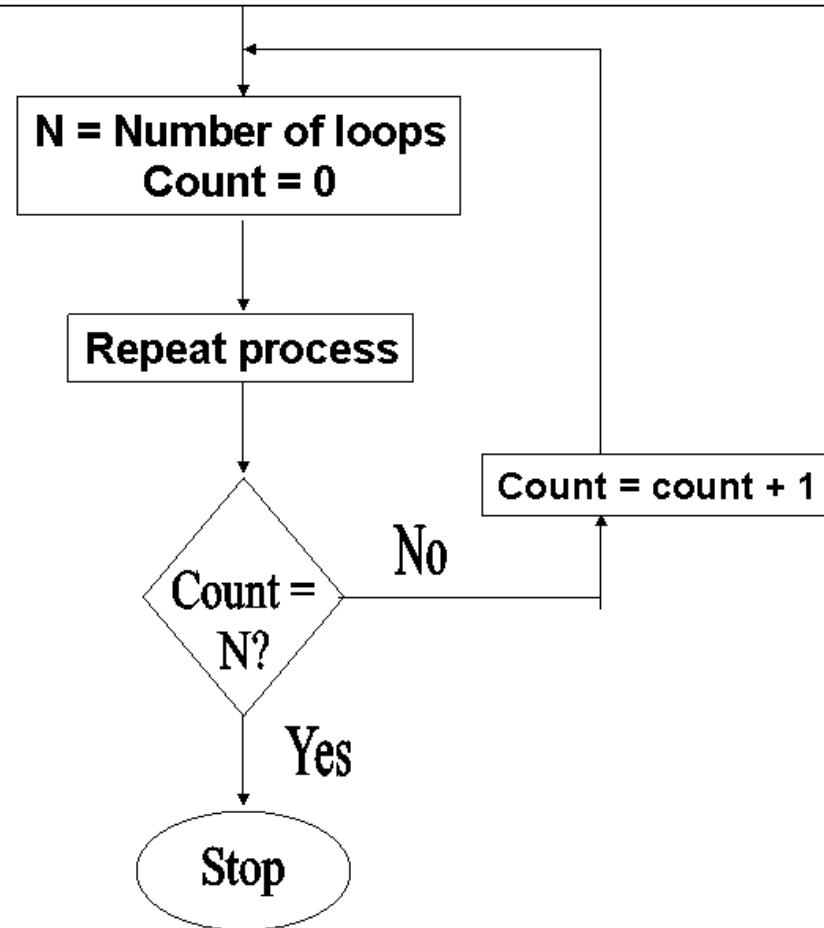
Answer of revision questions 1-4
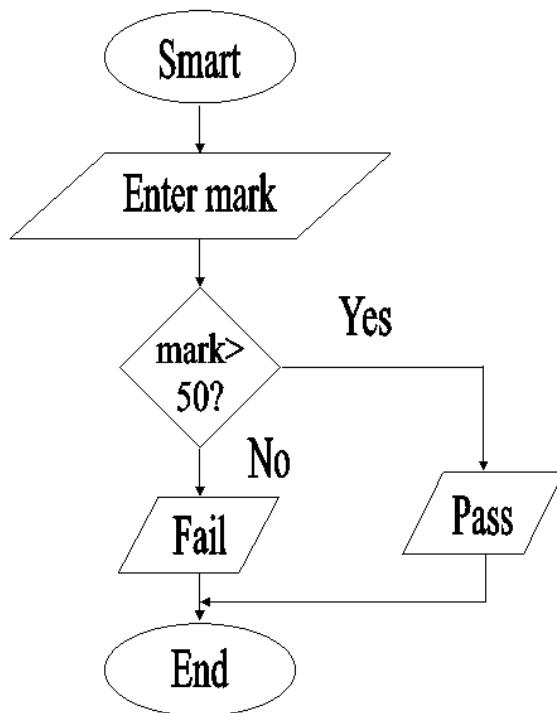
Revision questions 2
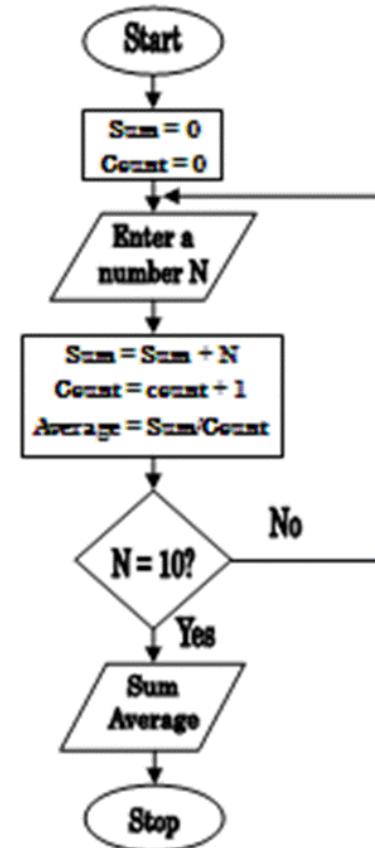
1. Write a program algorithm that would enable the user to enter student marks. The program should then determine whether the mark entered is a pass or fail given that the pass mark is 50.

2. Draw a flowchart for a program that is to prompt for N numbers, accumulate the sum and then find the average. The output is the accumulated totals and the average.

3. Write a pseudocode that reads temperature for each day in a week, in degree Celsius, converts the Celsius into Fahrenheit's and then calculate the average weekly temperatures. The program should output the calculated average in degrees Fahrenheit's.

Answer of revision questions 2-1

Answer of revision questions 2-2

3. START
    Input temperature in ℃
    et initial day to 0
        WHILE initial day 6 DO
        Input temperature in ℃
        F=32+(9C/5)
        Initial day=initial day + 1
        Cummulative = cummulative + F
        END WHILE
    Average weekly temperature
    = cummulative/7
    Print Average
    STOP

Revision questions 3

1. A lady deposits 2,000 shillings in a Microfinance company at an interest rate of 20% per annum. At the end of each year, the interest earned is added to the deposit and the new amount becomes the deposit for that year. Write an algorithm for a program that would track the growth of the deposits over a period of seven years.

2. Mukumu took a loan of Ksh 400,000 from a local bank at an interest rate of 10% payable in four years. Assuming you wish to develop a computer program that will keep track of monthly repayments:

   A) Identify the input, processing and output requirements for such a program.

   B) Design the algorithm for the program using a simple flowchart and pseudocode.

Answer of revision questions 3-1

1. START
   Input initial deposit
   Input interest rate
   Set deposit to initial deposit
   Set year to 0
       WHILE year $\leq$ 7 DO
       Interest = Deposit * interest rate
       Total = Deposit + interest
       Deposit = Total
       Year = Year + 1
       END WHILE
   PRINT Deposit, Year
   STOP

Answer of revision questions 3-2

2. A).

Inputs – initial amount borrowed

- Interest rate

- Number of years

Processing – the equation to calculate monthly repayments.

Output – monthly repayments calculated by the process

Answer of revision questions 3-3

2. B).
START
    Input initial loan borrowed

    Input interest rate

    Input number of years
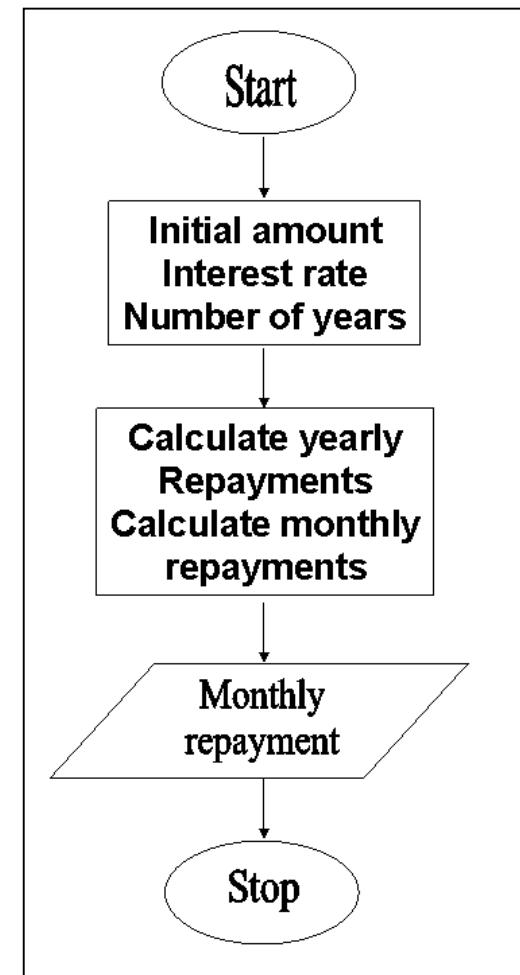
    Calculate yearly repayments
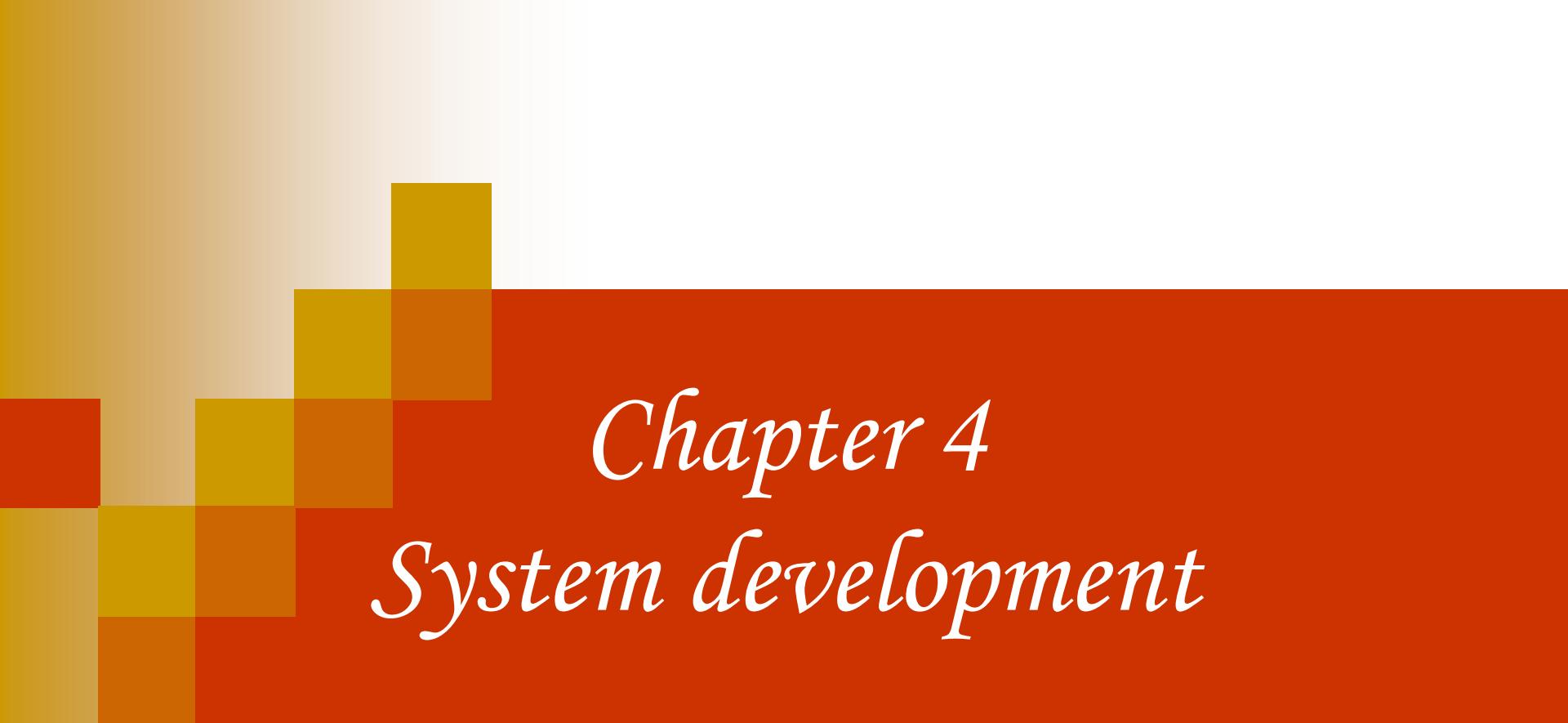
    Monthly repayments

    = yearly repayments/12

    Display monthly repayments

STOP

Chapter 4
System development

A system

- Is a set of items linked together to carry out a given task in order to achieve one or more objectives. A system is described as being either soft or hard.

Soft systems and hard systems

- Soft systems – human activity are said to be soft systems. They are said to be soft because:
  - ☐ Their boundaries may be fluid or keep on changing.
  - ☐ Their goals and objectives conflict.
  - ☐ It's difficult to precisely define exact measures of performance for them.
- Hard systems – these are systems whose goals and objectives are clearly defined and the outcomes from the systems processes are predictable and can be modeled accurately.

System classification

- Deterministic system – these are systems whose output are known precisely by their inputs e.g. a computer system.
- Probabilistic system – these are systems whose output can only be predicted but not knows precisely e.g. business and economic system organisation.
- Cybernetic systems – these are self controlling systems which adapt to the environment and regulate their own behaviour by accessing the feedback. They access their output and adjust the input e.g. human beings, plants systems

Characteristic of systems

1. Holistic thinking
2. Sub systems
3. Processor
4. Objectives/purpose
5. Environment
6. Boundaries
7. Interdependence
8. Feedback
9. Control
10. Inputs and Outputs
11. System Entropy

Holistic thinking and sub systems

- Holistic thinking – in this, a system is considered as a whole. A combination of various components that makes up a system creates a complex whole whose overall goals are more sophisticated than those of the individual components.

- Sub systems – a system is made up of different components (subsystem). Therefore a system does not exist in solitude but it may be a component of a large system.

Processor and objectives/purpose

- Processor – this is an element of a system that does the actual transformation of input into output.

- Objectives/purpose – objectives of a system may be real or stated. An organization should state one objective and operate to achieve another user must know the major objective of a system.

Environment and boundaries

- Environment – the environment is the system within an organisation operates. Environment provides a reason for the existence of the system.

- Boundaries – system boundaries are the external element whose changes in attitudes, behaviour and property affect their state and are affected respectively i.e. it's the limit of system.
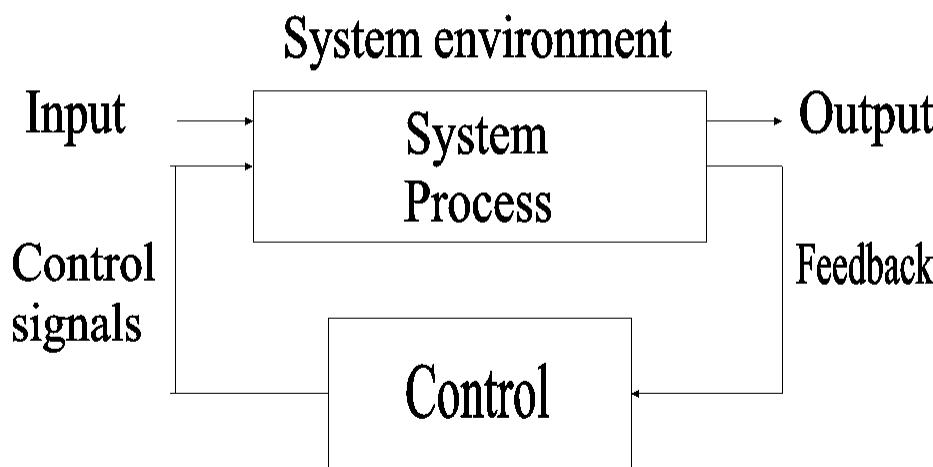
Interdependence and feedback

- Interdependence – for a system to be effective and efficient, its components or subsystems must be coordinated and linked together.

- Feedback – a feedback is a check within a system to see whether predetermined goals are being met. The checks are conducted on a system to determine deviation.

Control and inputs and outputs

System control

- Control – this is the process by which the system adapts to changes in the environment in order to give the expected level.
- Inputs and outputs – a system communicates with its environment by receiving inputs and giving outputs.

System Entropy

- Entropy means decay. System 'decay' naturally over time, hence it slowly becomes useless to the user either due to improvement in technology, new management or change in user requirements. Therefore it must be reviewed in order to improve it or develop a new one.

Open and closed system

- An open system receives input from and gives output to the environment while a closed system does not interact with the environment.

Information system

- This is a set of persons, procedures and technologies and other resources that collects, transforms and disseminates information in the system e.g. organisation.

Purpose of information system

- To provide the people within an organisation with information to assist in decision making process.
- To provide responsive service to meet user needs.
- To enable sharing of information.
- To facilitate interaction and dissemination of data throughout the various level of organisation.

Why develop new information system?

- New changes in organisation – management can introduce new changes, and they have to be implement to meet the needs of the organisation.

- Problems in the current system – the undesirable circumstances in the current system that prevent the organisation from meeting its goal.

- New opportunities – if an opportunity presents itself that can enable the organisations improve its services.

Theories of system development

- The most common theories include:

  ☐ Traditional Approach.

  ☐ Rapid Application Development (RAD).

  ☐ The Structured Approach.

Traditional Approach

- This approach relies mostly on the skills and experience of individual staff members carrying out of the project. In this approach, the manual system is replaced with a computerized one without change in overall structure of the former system. Hence the weakness of the former system are not addressed, they are forwarded to the new system.

Rapid Application Development (RAD)

- RAD model evolved from the theory that business today heavily rely on information technology, hence, development and implementation of information system needs to be quick enough for the organisation to maintain a competitive advantage in the market place.
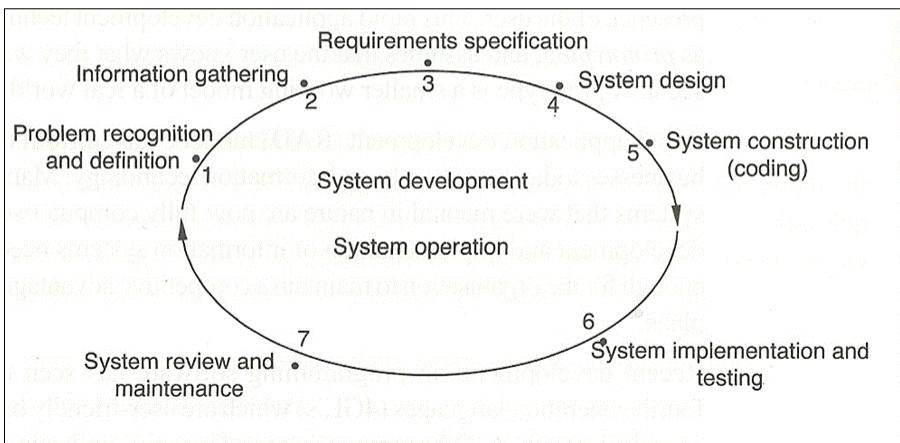
Structured Approach

- In this approach a system development defines a set of stages that should be followed when developing a system.

Stage of system development

The system development life cycle

- Problem recognition and definition
- Information gathering
- Requirements specification
- System design
- System construction
- System implementation

Problem recognition and definition

- Problem recognition is the initial stage where the analyst decides if the project is really necessary to look into, and if its necessary to start a new project. In problem definition, the problem must be thoroughly defined. A through survey of the problem is undertaken by identifying the problem, understanding it, and finding out of there is a possibility of getting a solution. Before a new system is designed, a special study will be carried out to establish the costs and benefits of a new system. The study is called **feasibility study.** Feasibility study of a system is accessed in four ways.
  - Operational feasibility
  - Technical feasibility
  - Economic feasibility
  - Schedule feasibility

Operational feasibility and technical feasibility

- Operational feasibility – this establishes the extend to which the users are comfortable or happy with the proposed or new system.

- Technical feasibility – this establishes whether the technology available is sufficient or can be upgraded for the new system.

Economic feasibility and schedule feasibility

- Economic feasibility – it establishes whether the technology available is sufficient or can be upgraded for the new system.

- Schedule feasibility – it establishes whether the development of the proposed system will be accomplished within the available time.

Information gathering

- The system analyst gathers data about the current system and the users needs from both internal and external sources. Some of the methods used to collect or gather data include:

  - Record inspection
  - Interview
  - Questionnaires

Record inspection and interview

Example of an interview

```
                    INTERVIEW TITLE
BRIEF INTRODUCTION          Interviewr:……….
                            Interviwee:……….

Interviewer:  Hello……….
Interviewee: Hello. Welcome to my office.
Interviewer:  Thank you. Please call Pat. I would
like to ask you a few questions about the library
information system that we are developing
Interviewer: …………………………………………….…
………………………………………………………..………
```

- The documents maintained in connection to the system are referenced. This forms a good starting point for the analyst to gain an overall insight of the existing system.

- Interview – this is a face to face communication between the analysts as the interviewer and the affected personnel as interviewee. The questions are geared to getting the information needed by the analysts.

Advantages of interview

- The analysts is accessible to first class information.

- Make interviewee feel as part of the system.

- Give adequate opportunity for interviewee, an opportunity to put forward opinions.

Disadvantages of interview

- Make the interviewee feel they are being summoned by the analysts.
- Some interviewees give answers which are not actually proven facts but opinions.
- The interviewee may not fully open on some issues that may be personal.
- The interviewee may be suspicious of the intentions of the proposed system for fear of being displaced by new system.

Questionnaires

An example of a questionnaire

QUESTIONNAIRE
BRIEF INTRODUCTION   Date:….......
……………………………………………………………
…………………………………………………………
QUESTIONS
1.   How long have you worked as a library attendant:
☐1 yr.   ☐ 2 yrs.   ☐over 2 yrs.
2.   How long does it take to rearrange books on the shelves?
☐days   ☐weeks   ☐months

- A questionnaire is a document containing a number of standard questions set to be asked to a larger number of people in order to gather information from them. The analyst is able to collect information from several key people in the organisation who may be affected by the current proposed system.

When the use questionnaire

- When limited amount of information is required from a large number of people.

- There is a large number of people involved in the system project.

- The people you need to gather information from are vastly despersed.

Advantages of questionnaires

- Answers filled in less tense environment.
- Since they are filled and returned in privacy, more sincere responses are possible.
- Questionnaire provides many people with the opportunity to provide input and suggestion.
- Questionnaire saves time especially where respondents are geographically scattered.

Disadvantages of questionnaires

- It's a slow way of facts gathering.
- Designing questionnaires requires expertise.
- Some questions can be easily misunderstood hence ambiguous replies will be received.
- Analysis of questionnaires requires a lot of time.

Observation

- This requires the observer to participate or watch closely as a person performs activities in order to learn about the system. This method gives the analysts first hand experience about the problems and expenses to the system requirements.

Advantages of observation

- Concepts that are too difficult for non-technical staff to explain can be observed.
- It helps the analysts become acquainted with operational personnel.
- It helps the analysts acquire 'know how' needed to assist in changes that have been recommended.
- Can correct 'hazy' understanding or erroneous impressions.

Disadvantages of observation

- It's costly and time consuming.

- Staff members are not comfortable when they are being watched.

Automated method

- This is mostly used when actual data is required but difficult to get through interviews, observation or questionnaires. Such data is collected using devices that automatically capture data from the source. E.g. video, cameras, tape recorders etc.

Requirements specification

■ The system analyst must come with the detailed requirements for the new system. The requirements specifications include:

- ☐ Output specification
- ☐ Input specification
- ☐ File/data stores
- ☐ Hardware and software requirements

Output specification

- The output specifications entails activities like generation of reports used to facilitate decision making. The output can be in the form of reports either in hardcopy or softcopy. The quality of those reports is important. Formality of reports is important so that good quality reports are produced for easy understanding.

Input specification

- The input to the system in necessary because the contents of the input are used to maintain the master files. The following should be considered in input specification.

  - ☐ The contents and volume of input.
  - ☐ The mode and devices of input selected.
  - ☐ Layout and sequence of input.

File/data stores

- File requirement involves making an informed decision on file required to store data and information in the system. The analyst should identify the number of files that will be needed by the system and determine the structure of each file.

Hardware and software requirements

- System analyst should specify all hardware and software requirements for the new system. He/she should consider:

  - Economic factors e.g. price
  - Operational factors e.g. reliability
  - User friendliness.

System design

- This involves detailing the physical design of the system, it's the how of the process. Once the logical design is put lined during system analysis, the analyst determines the physical design, which describes the hardware, software and operating procedures required to make system operational. Several tools are used for designing an information system. Examples are flowchart, data flow diagram, requirements.

System construction

- This refers to coding, installation and testing of the modules and their components such as outputs, inputs and files. The purpose of the construction is to develop and test a functional system that fulfills the business and design requirements.

Testing the system

- The system is tested by entering some test data to find out whether its output are as expected. The system is tested using the requirements specifications and the design specification to find out whether it meets all requirements specified.

System implementation

- This involves installing the system in the user's computers, testing the installed system, converting from old system to the new one and training the users. System implementation involves:

  - File creation and conversion.

  - Change over.

  - Staff training

File creation and conversion

- This involves setting up of the master files that are to be used in the support of the new system. The setting can be either from scratch or just to convert the files that were employed in old system.

Training staff

- The training aim at:

    - Convincing the staff of the user department on the effectiveness of the new system.

    - Remove fear of change from them.

    - Enabling the staff to cope with processing task of the new system.

Methods of training

- Demonstration
- Film show
- Manual reference
- Group discussion
- Visits

Changeover

- This is the process of changing over from old system to the new system.

- Type of changeover
  - ☐ Direct changeover
  - ☐ Parallel changeover
  - ☐ Phase changeover

Direct changeover

- In this approach the new system commence live operation and immediately the old system is abandoned.

- Advantage of direct changeover
  - It's a cheap method

- Disadvantages of direct changeover
  - Extremely risky
  - Places an organization in a do or die situation.

Parallel changeover

- In this changeover the operation of the old system and the new system run alongside each other.

- Advantages of parallel changeover
  - Provides a method of training in the new system.
  - Personnel become familiar with the new system prior to actual changeover.
- Disadvantages of parallel changeover
  - Costly to run two systems
  - Cross checking is difficult
  - Requires more staff hence more training required for additional staff

Phase changeover

- In this approach implementation is done only one part of the new system at one time or a step by step.

- Advantages of phase changeover
  - Allow the system to be implemented quickly
  - Less costly
  - Failures of the system are limited
  - Causes minimal disruption
- Disadvantage of phase changeover
  - Problems on ensuring that the first phase is implemented and converted.

System review and maintenance

- System maintenance is the adjustment and enhancement or correction of errors after the system has been implemented. The reviewing involves going through the specification and testing the system after implementation to find out whether it still meets the original specification.

System documentation consists of

- Report of fact finding which outlines methods used to collect data, weakness of the current system and recommendation of the current system.
- Requirements specification which entails the output requirements, input, hardware and software required for the new system.
- System flowchart which shows the overall functionality of te proposed information system.
- Tables or file structures depending on the approach used in system construction.
- Sample test data to test whether the new computerized information system is working as expected.
- User manual which helps the user work with the new system with minimal guidance. The manual contains information like
  - ☐ How to install, start the system
  - ☐ The interface of the system
  - ☐ How to carry out various tasks
  - ☐ Error correction and how to get help

Review questions 1

1. Define the term information system.
2. Differentiate between soft systems and hard systems.
3. List five characteristics of a system.
4. What is system control?
5. Why do we need feedback in a system?
6. Explain the tem system boundary.
7. State and explain three purposes of information systems in organisations.
8. Define the term online analytical processing?
9. Draw a well labelled diagram that depicts a system that is controlled through feedback.
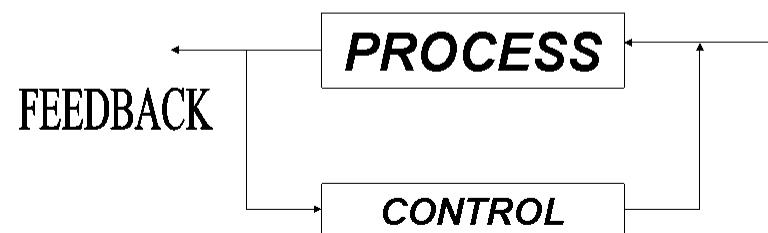
Answer of review questions 1-1

1. An information system is the arrangement of people, data, processes and information that work together to support and improve the day-to-day business operations.
2. In soft systems, goals are not clearly defined unlike in hard systems.
3. A). Holistic thinking
   B). Subsystems.
   C). Boundary and environment.
   D). Purpose.
   E). Process.
   F). System entropy.
   G). Inputs and outputs.
   H). Open and closed systems.
   I). Control
4. A system control is a method by which a system adapts to changes in the environment in order to give the expected output or perform at the expected level.

Answer of review questions 1-2

5. Feedback is needed in order to adjust controls for better performance of the system.

6. System boundary is the scope within which a system component operates.

7. A). Supports information processing tasks.

   B). Supports decision making.

   C). Supports sharing of information.

8. Online analytical processing is whereby a computer based information system is used to capture operational data, analyse and generate reports to be used for supporting decision making process in an organisation.

9. .

Review questions 2

1. Explain three system development theories and methods.
2. State the main disadvantage of Rapid Application Development method.
3. Define the term system development lifecycle.
4. At what stage does the system change hands from the development stage to the users.
5. Highlight three circumstances that necessitate the development of new information systems.
6. Define the term feasibility study as used in system development.
7. State four methods that can be used to gather information during system development.
8. State two advantages of interview over questionnaires.
9. State two disadvantages of interview method when used for information gathering.
10. Give one example of automated information gathering technique.

Answer of review questions 2-1

1. A). Traditional approach.
   B). Rapid Application Development (RAD)
   C). Structure approach.
2. The main disadvantage of RAD is that the system may be weak due to quick development.
3. System development life cycle refers to the steps followed in developing a system for conception to review and maintenance.
4. At the implementation stage.

5. A). New opportunities.
   B). Problems
   C). Directives.
6. Feasibility study is a special study carried out to establish the costs and benefits of a new system.
7. A). Interviews
   B). Questionnaires
   C). Observation
   D). Study of available documents.
   E). Automated methods.

Answer of review questions 2-2

8. A). Non-verbal communication can be used or observed.

    B). Questions can be rephrased for clarity.

9. A). It is difficult to organise for an interview due to time constraint.

    B). The interviewee may not fully open up some sensitive issues due to security concern.

10. An example of automated information gathering technique is the use of a video camera to compare motion pictures.

Review questions 3

1. Outline the four areas that are considered during the requirements specification stage.
2. Explain at least three factors that should be put into consideration during output design.
3. State four factors that may be considered in order to design a good file.
4. Explain why it is important to consider file backup and recovery strategies during file design.

Answer of review questions 3

1. A). Output
   B). Input
   C). Files
   D). Hardware and software requirements.

2. A). Target i.e. the audience.
   B). Frequency of report generation
   C). Quality and format

3. A). Record key field
   B). Type of data
   C). Length of each field
   D). Backup and recovery strategies.

4. Back up and recovery strategy helps in making sure that if the current file gets corrupted, it is possible to reconstruct the original file.

Revision questions 1

1. Outline seven steps followed in structured system development.
2. Explain three tasks that are carried out during system implementation.
3. Outline three disadvantages of questionnaires.
4. Explain the concept of proxemics in interviews.
5. Give an advantages of straight changeover over parallel changeover.
6. Define the term file conversion.
7. Differentiate between a system flowchart and a program flowchart.
8. Draw four system flowchart symbols and explain their functions.

Answer of revision questions 1-1

1. A). Problem recognition and definition.

   B). Requirement specification.

   C). System design.

   D). System construction.

   E). System implementation.

   F). System review and maintenance.

2. A). File conversions.

   B). Staff training.

   C). Project management.

   D). Changeover

3. A). Good questionnaires are difficult to prepare.

   B). The respondent may not understand some questions.

   C). The respondent has the final say on when to fill and return the questionnaire. Most of them end up not being filled or returned hence it is disadvantageous to the researcher.

Answer of revision questions 1-2

4. Proxemics refer to issues related to physical contact such as sitting arrangement during an interview.

5. Straight changeover is cheaper to implement than parallel changeover.

6. File conversion refers to changing files from one format to another.

7. A system flowchart is for the entire system while a program flowchart is for a part or a module in the system. Therefore, several program flowcharts may be derived from a single system flowchart.

8.

Start/stop

decision

Computerised Process

Disk/ master file

Revision questions 2
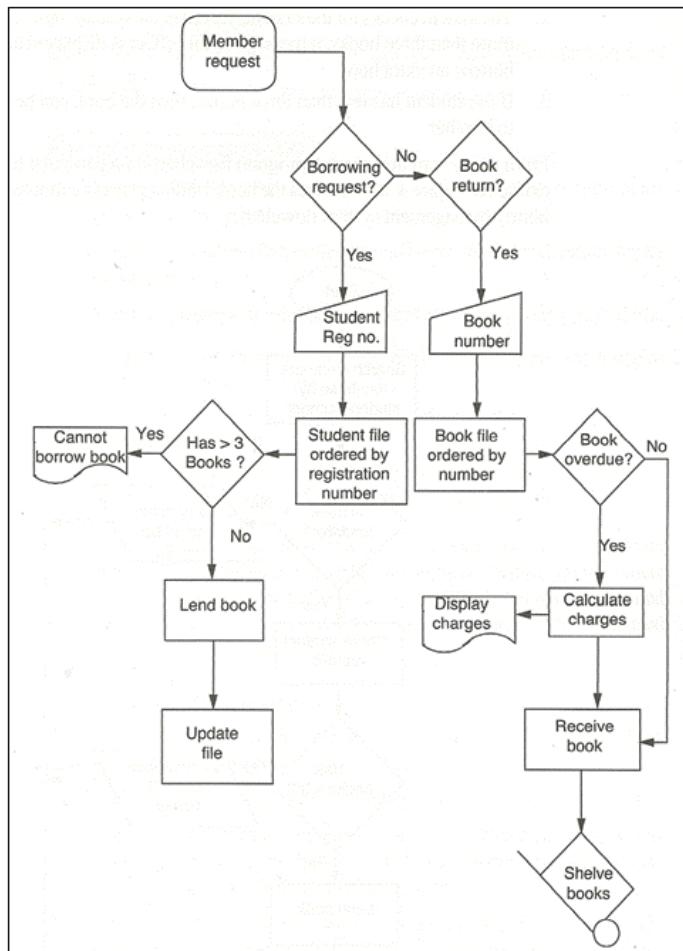
1. Define the term "attribute".
2. Explain the importance of using automated methods in fact finding.
3. Why is observation sometimes disadvantageous when used in fact finding?
4. State one of the disadvantages of the traditional approach in system development.
5. Outline the factors that should be considered when sourcing for hardware and software resources required for a new system.
6. Outline some of the contents of a user manual that would help the user run the system with minimal guidance.

Answer of revision questions 2

1. Entity attributes refer to characteristics of a particular entity.
2. Automated methods of data collection can be used in areas which are not easily accessible.
3. A). The person being observed might alter his or her behaviour.
   B). It may be time consuming.
4. The weaknesses of the former system are not addressed.
5. A). Economic factors.
   B). Operational.
   C). Services from the vendor.
6. A). Installation procedures.
   B). User interface.
   C). How to carry out various tasks.
   D). Test data and expected output
   E). Troubleshooting guide

Practical exercise 1

- Extract the book return module flowchart from the system flowchart depicted in figure right.

END