Assignment No. 01

Page No.
Date

Q. 1) Define Data and Data Types and explain concept Array with example.

→

information

facts Data Graphs

quantities

- Data :

i) Data can be raw facts.

ii) It can be defined as collection of information. gathered by facts, quantities, graphs, analysis, etc.

iii) They may consist of facts, numbers, names, description of things.

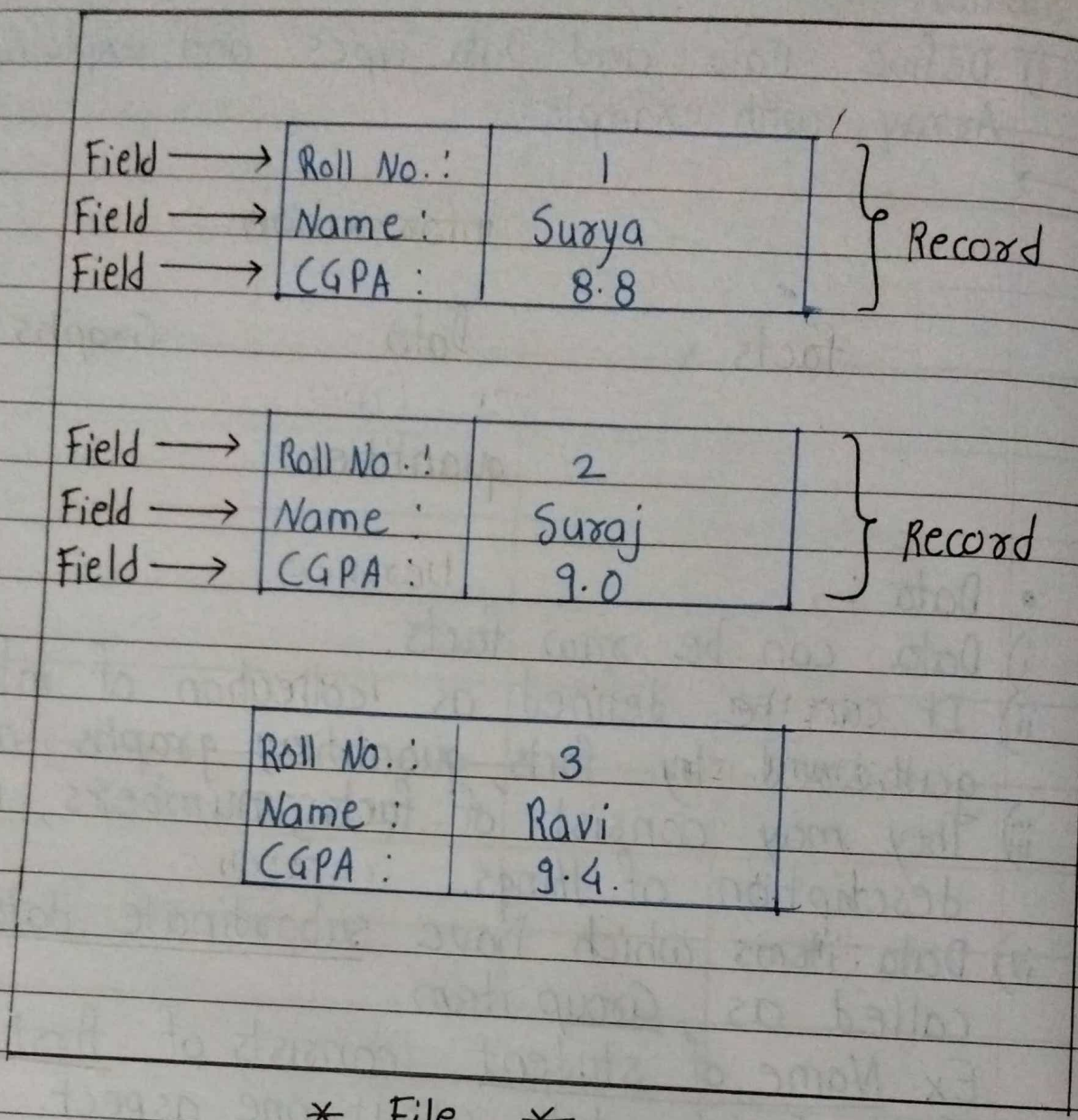
iv) Data items which have subordinate data items are called as Group item.
Ex. Name of student consists of first name & last nam

v) Field : Contains data about one aspect.

vi) Record : can be defined as the collection of various data items. (set of fields)

vii) File : A collection of records involving a set of entities with certain aspects in common and organised for some purpose is called a File.

viii) For example,



* File *

- Data Types :

i) A data type is the most common classification of data.
ii) Through which the compiler gets to know the form or the type of information transmitted between the programmer and compiler, which will be used throughout the code.
iii) It informs the compiler about what type of data is to be stored and also tells how much space it requires in the memory.

Page No.
Date

ii) C++ supports the following data types:
1. Built-in data type
2. Derived data type
3. User-defined data type

Data Types

Primary	Derived	User-Defined
integer	function	class
character	array	structure
boolean	pointer	union
float	reference	enum
double		type def
void		

- Array:

i) Array is a collection of items of same data type stored at contiguous memory locations.
ii) It means an array can contain one type of data only, either all integer or all floating number, etc.
iii) The array has a fixed size meaning once the size is given to it, it cannot be changed.
iv) The array elements can be accessed by its index number (index number starts with 0)

v) There are three types of array.
a) One Dimentional
b) Two Dimentional
c) Multi Dimentional.

vi) Syntax for declaring an array

```
data_type array_name [size];
```

vii) Example:
An array containing names of 3 students.

```
char std_name[3] = {"Surya", "Suraj", "Ravi"};
```

Q.2) Define Data Structure with example.
→

i) Data structure can be defined as the group of data elements which provides an efficient way of storing and organizing data in the computer so that it can be used efficiently.
ii) Data structures are widely used in almost every aspect of Computer Science i.e Operating system, Compiler design, AI and many more.
iii) It plays a vitle role in enhancing the performance of a software or a program as the main function of the software is to store and retrieve the user's data as fast as possible.

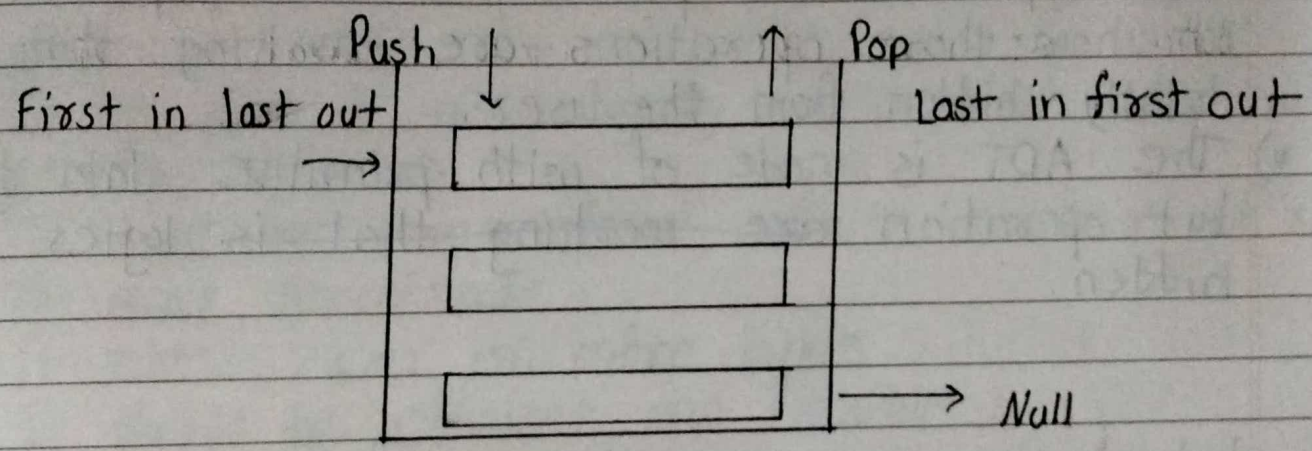
iv) Some examples of Data structures are arrays, Linked list, stack, queue, etc.

- Stack :

i) Stack is a linear data structure which follows a perticular order in which the operations are performed.

ii) Insertion of element into stack is called push and deletion of element from stack is called pop.

iii) The order may be LIFO (Last in First out) or FILO (First in Last out).



→ Operations of Stack :

i) Push () : It adds a new item at the top of stack.

ii) Pop () : It remove the top item of stack.

iii) Peek () : It returns the top item of stack.

iv) isEmpty () : It test whether the stack is empty.

v) size () : It returns the number of items in the stack.

Q.3) Explain Abstract data type (ADT).

→

i) The Data Type is basically a type of data that can be used in different computer program.
ii) It signifies the type like integer, float, etc. the space like integer will take 4-bytes, character will take 1-byte of space. etc.
iii) The abstract data type is special kind of data type, whose behaviour is defined by a set of values and set of operations.
iv) The keywoord "Abstract" is used as we can use these data types, we can perform different operations. But how those operations are working that is totally hidden from the user.
v) The ADT is made of with primitive data types, but operation ~~are working that is~~ logics are hidden.

Let us see some operations of those mentioned ADT

- Stack:

i) is Full ()
ii) is Empty ()
iii) push()
iv) pop()
v) peek()
vi) size()

Page No.
Date

- List :
  i) size ()
  ii) insert ()
  iii) remove ()
  iv) get ()
  v) replace (x, y)

Q.4) State the characteristics of Algorithm.

→

i) It must be correct.
ii) It must be composed of a series of concrete steps.
iii) There can be no ambiguity as to which step will performed next.
iv) It must be composed of a finite number of steps. It must terminate
v) It takes zero or more inputs.
vi) It should be efficient and flexible.
vii) It results in one or more outputs.
viii) It should use less memory space as much as possible
ix) An algorithm states the actions to be executed and the order in which these actions are to be performed.
x)

Q 8) Draw & explain classification of Data Structure.

→

i) Data structure can be defined as group of elements of data which provides an efficient way of storing and organizing the Data in computer.

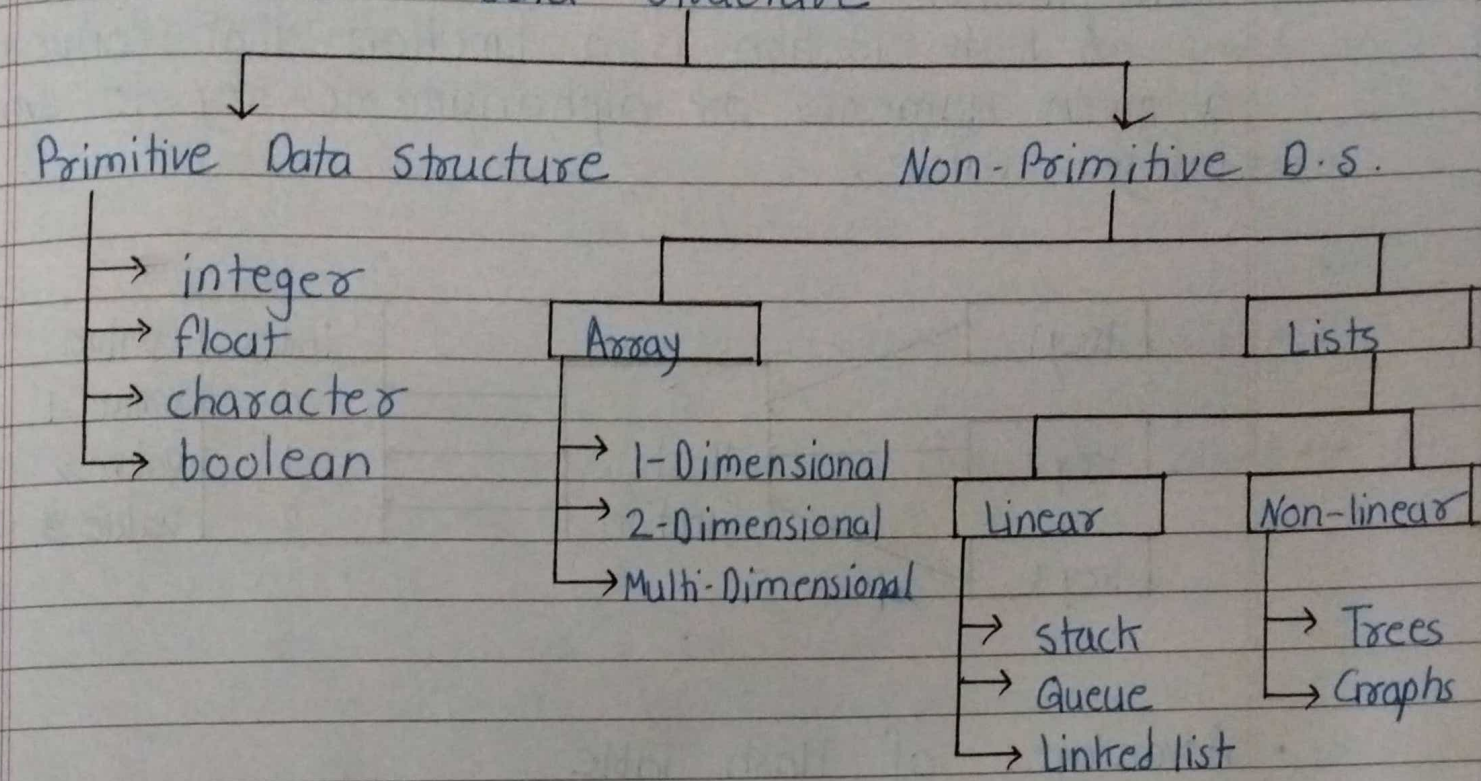
ii) Data structure can be classified into two types.
   a) Primitive Data structure
   b) Non-primitive data structure.

a) Primitive Data Structure:

Data structure which are directly operated upon the machine-level instructions are known as Primitive Data Structure.
Ex. int, float, etc.

b) Non-Primitive Data Structure:

These are more sophisticated data structures.
These are derived from primitive data structure.
Ex. array, graph, stack, etc.



Q. 6) Explain Hash table, Hashing, Hash Function

→

1. Hash Table :

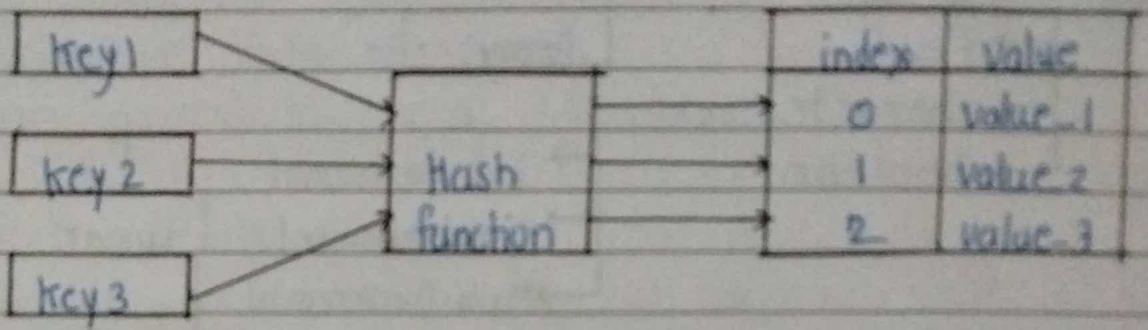
Hash Table is a data structure which stores the data in associative manner in form of an array In the hash table, the data is stored in hash table where each data value has its own unique index value.

2. Hashing :

Hashing is technique to convert a range of key values into range of indices. We are going to use modulus (%) operator to get range of key values.

3. Hash Function:

A hash function is a function that converts a given numeric or alphanumeric key to small integer value.



- Advantages of Hash Table

i) Hash table has high performance when looking up for data insertion and deletion.

ii)

- Disadvantages

i) Can not use null value as a key.

ii) Collision cannot be avoided.

Q.7) Explain Perfect Hashing:

→

i) A perfect hash function is one that maps the set of actual key values to the table without any collisions.

ii) A minimal perfect hash function does so using a table that has only as many slots as there are key values to be hashed.

iii) Perfect and minimal perfect hashing is readily suitable only for static sets, i.e sets in which no deletion and insertion of elements occurs.
1) It must be very easy & quick to compare
2) It must minimize collisions.

iv) There are three perfect hashing technique.
1) Division Method:
Choose a number m smaller than the number of n of keys in K.

- Hash (key) = key % Table_size or
  = key % Table_size + 1

2) Mid square method:
We first square the item and then extract some portion of resulting digits.

Ex. item = 44
$(44)^2$ = 1936
By extracting middle two digits and performing remainder method.
93 % 11 = 5 (11 → size)

3) Folding Method:
i) The folding method for constructing hash functions begins by dividing the item into equal-size pieces
ii) These pieces are then added together to give the resulting hash value.

Q.8) Write short note on.
1. Direct Address Table
2. Open Addressing

→

1. Direct Address Table :

- Direct Address Table is a data structure that has the capability of mapping records to their corresponding keys using arrays.

- In direct address tables, records are placed using their key values directly as indexes.

- They facilitate fast searching, insertion and deletion operations.

2. Open Addressing

- The open addressing is another technique for collision resolution.
Unlike chaining, it does not insert elements to some other data structures.

- It inserts the data into the hash table itself. The size of the hash table should be larger than the number of keys.

There are three types of Open Addressing :
1) Linear Probing
2) Quadratic Probing
3) Double Hashing

Q.9)
→

Data Structure :

i) Data structure can be defined as group of elements of data which provides an efficient way of storing and organizing the Data in computer.

key	key mod 10 = H(key)	index
32	32 % 10 = 2	2
18	18 % 10 = 8	8
23	23 % 10 = 3	3
2	2 % 10 = 2	
	(2+0) mod 10 = 2	
	(2+1) mod 10 = 3	
	(2+2) mod 10 = 4	4
3	3 % 10 = 3	
	(3+1) mod 10 = 4	
	(3+2) mod 10 = 5	5
44	~~(3+~~44 % 10 = 4	
	(4+1) mod 10 = 5	
	(4+2) mod 10 = 6	6
5	5 % 10 = 5	
	(5+1) mod 10 = 6	
	(5+2) mod 10 = 7	7
15	15 % 10 = 5	
	(5+1) mod 10 = 6	
	(5+2) mod 10 = 7	
	(5+3) mod 10 = 8	
	(5+4) mod 10 = 9	9