Date .....
Page .....

Programming Concept :

Programming Elements : —

- **Variable** : —

It is name of memory location which store same value.

$x = 10$

x is a variable . i.e name of memory location where we store a value 10

$y = x + 10$

- **Constant** : —

Any type of value store in any memory location called constant.

$x = 10$

10 is a constant which store in memory location x.

There are following type of constant.

integer - 10
float - 10.01
character - x (' ')
String - xyz (" ")
Boolean - T/f

Date .....
Page .....

- **Data Type :-**

It is tells which type of value can be store in a variable.

- **Array :-**

Array is name of collective memory location where we store data of similar type.
Similar type
the memory location must be continuous.
The element of array access by array name and subscript .in c subscript start from 0

- **Processing of Array :-**

Declaration - int x[10];
Inputting x[0] = 10
Processing - y = x[0] + 10

- **Types of Array :-**

one dimentional (1D)
Two Dimentional (2D)
Three Dimentional (3D)

Date .....
Page .....

Ques:- write an algorithm to enter the marks obtain by ten students and display the marks of each student.

Step 1. start
2. int a[10] = {20,30,40,50,60,70,80,90,100,110}, i;
3. i = 0;
4. for (i; i<10; i++)
5. Print ("%d", a[i]);
6. end.

Ques:- write an algorithm to create an array of ten element and find out so is present or not in array as element.

Step 1. a[10], search, i = 0, k = 0
2. Print "Enter 10 element in array:"
3. for (i=0; i<10; i++)
4. input a[i]
5. Print "Enter Search element".
6. input search
7. for (i=0; i<10; i++)
8. if (search == a[i])
9. k = 1
break;
end if
10. if (k == 1)
11. Print "search is successful"
12. else
13. Print "data not found"
14. end if
15. close

Date .....
Page .....

Ques:- write an algorithm to create an array of 10 element change a value 4 with 40.

```
Step 1.  a[10], old, New, i, k=0, m
     2.  Print "Enter 10 element in array"
     3.  for (i=0; i<10; i++)
     4.  input a[i]
     5.  Print "Enter change value"
     6.  input old
     7.  Print "Enter new value"
     8.  input New
     9.  for(i=0; i<10; i++)
     10  if (a[i]==old)
     11   m=i
     12  k=1
     13  break;
     14. end if
     15  end for
     16  if (k==1)
     17. a[m] =new
     18   else
     19. Print "value not present"
     20. end if
     21. for [i=0; i<10; i++)
     22. Print a[i]
         end for
     23. end.
```

Date .....
Page .....

Ques:- write an algorithm to create an array of 10 element display maximum value.

Step-1. a[10], max = 0, i
2. Print "Enter 10 element in array"
3. for (i=0; i<10; i++)
4. input a[i]
5. for (i=0; i<10; i++)
6. if (max<a[i])
7. max = a[i]
8. end if
9. end for
10. Print max.

Ques:- write an algorithm to insert the marks of 100 students display minimum marks.

Step-1. a[100], min=0, i
2. Print "Enter 100 students marks in array".
3. for [i=0; i<100; i++)
4. input a[i]
5. for [i=0; i<100; i++)
6. if (min>a[i])
7. min = a[i]
8. end if
9. end for
10. Print min

Date .....
Page .....

Ques:- write an algorithm create an array of 10 students and perform deletion on the basis of value input by user.

Step-1 a[10], del, i, pos
2 Print "Enter 10 element in array"
3. for (i=0; i<10; i++)
4. input a[i]
5. Print "Enter deleted data"
6. input del
7. for (i=0; i<10; i++)
8. if (a[i] == del)
9. pos = i
10. break;
11. end if
12. end for
13. for (i = pos; i<10; i++)
14. a[i] = a[i+1]
15. end for
16. for (i=0; i<9; i++)
17. Print a[i]
18. end for
19. end

Date .....
Page .....

Ques:- write an algorithm create an array of element inter a value in any specified position in array.

```
Step-1.   a[10], pos, val, i
2.        Print "Enter 9 element in array"
3.        for [i=0; i<9; i++)
4.        input a[i]
5.        end if
6.        Print "Enter insert on position"
7.        input pos
8.        Print "Enter value"
9.        input value
10.       for (i= 9; i<pos-1; i--)
11.         a[i] = a[i-1]
12.       end for
13        a[pos-1] = value
14.       for (i=0; i<10; i++)
15.       Print a[i]
16.       end if
17.       end.
```

Date .....
Page .....

Ques:- write an algorithm to create two array of 5 element each and perform merge operation i.e (put the elements of both array in third array)

```
step1.  a[5], b[5], c[10], i, j
2.  Print "Enter five element in first array"
3.  for (i=0; i<5; i++)
4.  input a[i]
5.  Print "Enter five element in second array"
    for (i=0; i<5; i++)
      input b[i]
6.    j=0
7.    for (i=0; i<10; i++)
8.    if (i<5)
9.    c[i] = a[i]
10    else
11.   c[i] = b[j]
12.   j = j+1
13.   end for
14.   Print "element of array after merge"
15.   for (i=0; i<10; i++)
16.   Print c[i]
17.   end
```

Date .....
Page .....

Que:- write an algorithm to create two array of five element and put the element of each array in third array in alternate way. starting with element of array a.

```
Step 1.  a[5], b[5], c[10], i, J, K
2.       Print "Enter five element in first array".
3.       for(i=0; i<5; i++)
4.       input a[i]
5.       Print "enter five element in second array".
6.       for(i=0; i<5; i++)
7.       input b[i]
8.       J=0, i=0
9        for(K=0; K<10; K++)
10       if(K % 2 == 0)
11        c[K] = a[i]
12        i = i+1
13       else
14.      c[K] = b[J]
15.      J = J+1
16.      end if
17.      end for
18.      Print "enter of array after merge"
19.      for(i=0; i<10; i++)
20       Print c[i]
21.      end
```

Date .....
Page .....

Ques:- write an algorithm to create an array of 10 element display maximux value.

Step 1. a[10], max = 0, i
2. Print "Enter 10 element in array"
3. for(i=0; i<10; i++)
4. input a[i]
5. for(i=0; i<10', i++)
6. if(max<a[i])
7. max = a[i]
8. end if
9. end for
10. Print max.

Ques:- write an algorithm to insert the marks of 100 students display minimum marks.

Step 1. a[100], min = 0, i;
2. Print "Enter 100 students marks in array".
3. for [i=0; i<100; i++)
4. Input a[i]
5. for(i=0; i<100; i++)
6. if (min > a[i])
7. min = a[i]
8. end if
9. end for
10. Print "min"

Date .....
Page .....

Qu:- write an algorithm create an array of 10 students and perform deletion on the basis of value input by user.

```
Step 1.  a[10], del, i, pos
2.  Print "Enter 10 element in array"
3.  for (i=0; i<10; i++)
4.  input a[i]
5.  Print "Enter deleted data".
6.  input del
7.  for (i=0; i<10; i++)
8.  if (a[i] == del)
9.  Pos = i
10. break;
11. end if
12. end for
13. for (i= pos; i<10; i++)
14. a[i] = a[i+1]
15. end for
16. for (i=0; i<9; i++)
17. Print a[i]
18. end for
19. end
```

Date .....
Page .....

Ques:- write an algorithm create an array of 10 element insert a value in any specified potion in array.

Step 1. a[10], pos, val, i
2. Print "Enter 9 element in array"
3. for [i=0; i<9; i++)
4. input a[i]
5. end if
6. Print "Enter insertion position".
7. input pos
8. Print "Enter value"
9. input value
10. for(i=9; i<-pos-1; i--)
11. a[i] = a[i-1]
12. end for
13. a[pos-1] = value
14. for (i=0; i<10; i++)
15. Print a[i]
16. end if
17 end

Date .....
Page .....

- Sorting :- —

```
Step-1.  a[10], i, j, temp = 0
    2.   Print "Enter 10 element"
    3.   for (i=0; i<10; i++)
    4.   input a[i]
    5.   for (i=0; i<10; i++)
    6.     for (j=0; j<10-i; j++)
    7.       if (a[i] > a[i+1]
    8.           temp = a[i]
    9.           a[i] = a[i+1]
    10.      a[i+1] = temp
    11.    endif
    12.  end for
    13.  end for
    14.  Print "element after shorting"
    15.  for (i=0; i<10; i++)
    16.  Print a[i]
    17.
```

Date .....
Page .....

- **Expression :—**

It is a combination of operator and operand. An operator may be relational, logical or Arithemetic operator. An operand may be varible or constant. for example :- $x+y+2$ is an expression in which $x, y, 2$ are operand and + is an operator.

- **Statement :—**

An statement is any single line written in algorithm which perform some operation of algorithm. Example - Print "abc" is an statement which print a message abc on the screen.

- **Control Structure :—**

The execution of any statement is called control. we can move control accourding to our need at any position of algorithm. There are following way to move control in an algorithm.

i) Sequence :— It is default control structure in which control move from top to buttom. It is not control by user.

Date .....
Page .....

ii) selection :—

In this control structure we can move control at any place of an algorithm based on some condition. Condition either true or false. we can move control at any place if condition is true otherwise move to another place depend on code written in algorithm

for example: —

step 1. Print "abc"
2. a=10
3. if (a>10)
4. Print "pst"
5. else
6. Print "pqr"
7. end if
8. end

iii) Iteration : —

In this control structure control move in cycle. it required when some statement execute for ~~mult~~ multiple times. There are following types of Iteration statement:

Date .....
Page .....

- Subroatine & function :—

```
for (i=0; i<10; i++)
print "abc"
```

```
i = 0
while (i>10)
do
print "abc"
i = i+1
end while.
```

- Nested loop ; — when a loop enclose in another loop called nested loop

Example :—

```
for (i=0; i<10; i++)
  for (j=0; j<10; j++)
  print "i"
  end
```

ns:- write an algorithm to print following digram.

```
1 1 1 1 1
1 1 1 1 1
1 1 1 1 1
```

```
Step-1.   let x=1
2.        for (row=0; row<3; row++)
3.        for (col=0; col<5; col++)
4.        print x
5         end for
```

Date .....
Page .....

6. Print "\n"
7. end for

Ques:- write an algorithm to print following digram

```
1  2  3  4
5  6  7  8
9  10 11 12
```

Step 1. Let x = 1
2. for (row = 0; row < 3; row++)
3. for (col = 0; col < 4; col++)
4. Print x
5. x = x + 1
6. end for
7. Print "\n"
8. end for

Ques:- write an algorithm to print following digram.

```
1 2  3 4
1 2 3 4
1 2 3 4
```

Step - 1. for (row = 0; row < 3; row++)
2. x = 1
3. for (col = 0; col < 4; col++)
4. Print x
5. x = x + 1
6. end for
7. Print \n
8. End for

Date .....
Page .....

Ques:- write an algorithm to display table from 1 to 10.

```
step-1  for (row= 1; row <11; row++)
   2    for( col= 1; col <11; col++)
   3    Table = row * col
   4    Print " Table"
   5    end for
   6    Print "\n"
   7    End for
```

Ques:- write an algorithm to display all prime number between 1 to 100.

```
Step1.  for(num=2; num<100; num++)
   2.   count = 0
   3.   for( i=1; i<=num; i++)
   4.   if (num % 2 == 0)
   5.    count = count + 1
   6.    end if
   7.   end for
   8.   if (count == 2)
   9.   Print num\n
  10.   end if
        end for.
```

Date .....
Page .....

Ques:- write an algorithm to display all armstrong number between 1 to 1000.

1. let dig, x, sum
2. for(num=1; num<1000; num++)
3. sum=0
4. for(x=num; x>0; x=x/10)
5. dig = x%10
6. sum = sun + dig * dig * dig
7. end for
8. if(sum==num)
9. Print num \n
10. end if
11. end for

Ques:- write an algorithm to display following digram.

Ans
```
1 2 3 4
1 2 3
1 2
1
```

Step 1. for (row=4; row>0; row--)
2. for(column=1; column<=row; column++)
3. Print column
4. end for
5. Print \n
end for.

Date .....
Page .....

Q)

```
   1
  1 2
 1 2 3
1 2 3 4
```

```
1. for (row= 4; row>0; row--)
2.   x=1
3.   for (col= 1; col<=4; col++)
4.     if (col>=row)
5.       Print x
6.     x=x+1
7.     else
8.       Print " "
9.     end if
10.  Print \n
11.  end for.
```

Q)

```
1
1 2
1 2 3
1 2 3 4
```

```
1. for (row=1; row<4; row++)
2.   for(col=1; col<=row; col++)
3.     Print col
4.   end for
5.   Print \n
6.   end for.
```

Date .....
Page .....

⊖⇒

```
    *
  * *
* * *
* * * *
```

```
1. for (row = 4 ; row > 0 ; row - - )
2. for ( col = 1 ; col <= 4 ; col++ )
3.   if (col >= row)
4.     Print *
5.   else
6.     Print " "
7.   end if
8.   end Print \n
9.  end for
10. end for
```

or,

```
1. for ( row = 1 ; row <= 4 ; row++ )
2. for ( col = 1 ; col <= 4 ; col++ )
3. if ( col >= 5 - row)
4. Print *
5. else
6. Print " "
7 end if
8. end for
9 Print \n
10. end for
```

Date .....
Page .....

Q) 4 3 2 1
3 2 1
2 1
1

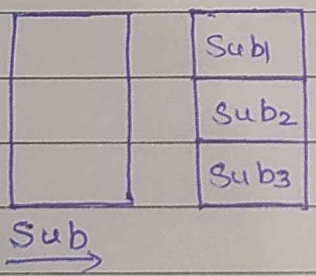
Step 1. S = 4
2. for(row = 1; row <= 4; row++)
3. x = 4
4. for (col = 1; col < 4; col++)
5. if (col >= row)
6. Print x
7. x = x - 1
8. else
9. print " "
10. end if
11 S = 4 - row
12. Print "\n"
13. end for.

Q) * * * *
* * *
* *
*

Date .....
Page .....

- Subroutine and function :—

It is a set of instruction which perform some specific task of any algorithm. It is helpful in writing any algorithm. we use it at anyplace of an algorithm where we required to perform some task which is responsible by that subrotine. It is known as function, procedured.



Advantage :—

→ Break a big program into different part
→ easy in writing
→ easy in uderstanding
→ fast error finding
→ minimum repitation.

Date .....
Page .....

Ques:- write a function in c which accept two number and return some of input number.

```
#include<stdio.h>
#include<conio.h>
void main()
{
int abc(int x, int y);
int a, b, c
Printf("enter two number");
Scanf("%d%d", &a&b);
c= abc(a, b);
}
Printf("%d", c);
int abc(int x, int y)
{
int r;
 r= x+y;
return r;
}
```

Algorithm
Step 1. abc(p, q)
2. Print "enter two number".
3. input a, b
4. s= abc(a, b) / "goto step 7"
5. Print S
6. end
7. abc(a, b)
8. S= a+b
9. return S
10. goto step 4

Date .....
Page .....

Ques:- write an algorithm which accept redius of a circul and return area of circul.

```
Step 1.  Area(r)
    2    Print "enter the radius"
    3    input a
    4    S= Area(a) / goto Step 7
    5    Print S
    6    end
    7    Area (a)
    8    r = (22 x a x a) / 7
    9    return r
   10.   goto step 4
```

Ques:- write an function which accept the number and return reverse of input number.

```
1.   Reverse (a)
2.   Print "enter a number"
3.   input x
4.   Rev = Reverse(x) / goto step7
5.   Print " Reverse of number =", Rev
6.   end
7.   Reverse (x)
8.   let m=0
9.   while (x>0)
10.    do
11.    dig = x % 10
12.    m = m*10 + dig
13.    x=x/10
```

Date .....
Page .....

14. end while
15. return m
16. end reverse/ no to step4.

Ques:- write an algorithm to enter a number and check that number is even or odd using function.

Step 1. check number(a)
2. Print "enter a number"
3. input x
4. s = check num(x) / no to 10
5. if (s==0)
6. Print "number is even
7. else
8. Print "number is odd"
9. end if
10. check num(a)
11. let s
12. s = a%2
13 return s
14. end check num
15. end

Ques:- write an algorithm to enter two number and display greather number using function

step 1. check num (a, b)
2. Print "enter two number"
3. input (x, y)

Date .....
Page .....

```
4. r= check num (x, y) / 40 to step 10
5. if (r==0)
6. Print "a is greather"
7. else
8. Print "b is greather"
9. end if
10. check num (x, y)
11  if (x>y)
12. return 0
13. else
14. return 1
15  end if
16. end check num.
```

- Conventional programming technice —

Programming technique change regularly since the invention of computer when computers where first invented programming had been done by binary machine instruction. As long as program where just a few hundred instructions long then a programmer face many problem to write the program. Assembly language was invented so that a programmer easily deal with complex program using symbolic representation of the machine instruction. As the program continue to pro high level language where introduced that give the programmers more tools which to handle

Date .....
Page .....

complexity. The first high level language was fortran. It was very impressive, it provide the facility to understand the program clearly. the birth of structural programming took place in 1960.

The structural programming :—

The structural programming is the method into which the long program can be break into different part, each part compile sperately and then link the all different part get the result. structure design can not reduce the over all compexity of a program, but breaking it into smaller part means that the amount of details that the programmer has to understand for one part is much less then that it dealing with the system as one.

The language such as c and pascal provide structure programming. The job of maintaning the program become easier and also reauire less time to understand the program changes can be made more quickly then if the program consisted of one long piece of code. Any program produced can be made from a combination of sequence, selection and itration of control structure.

Date .....
Page .....

functional programming :—

A function programming is based on the consept of function use in mathmatics. A function in mathmatics return same value. if we pass same value to the function for several time. Now the return value of function depends on the argument pass to the function using this concept the programmer write a code to perform this function in program. Example :- print is a function use in programming in c language to print any massage on the screen and display the value present in variable.

Ex- Print f ("Swati")

object oriented programming :— (OOPS)

Any programming based on object called object oriented programming.. for example - c++, Java

i> object - Any thing present in our world is called object. each object having some attribute and method for example - pen is an object which having attributes like colour, price, company and its method is writting.

Date .....
Page .....

ii) class :- A group of object having same attributes and method called class. for example - parrot, sparro and object in class birds.

iii) encapsulation :- Integration of data and operation in a class is called encapsulation it can be perform using keywords private, public, protected

iv) Abstraction :- it is an important feature of oops. it is used to display only necessary and assential feature of an object to the out side world. it hide the internal structure of the object for example- A user using a switch board to on/off a fan without knowing the wairing of switch board.

v) Inheritance :- it is used to aquire the data and operation of any exiting object. for example- at A and B are two object in which be aquire the data represent in object A.

vi) polymorphism - it is the process of object oriented programming in which any method, operator, perform multiple teske based on value pass to the operator or method. it is also called overloading.

Date .....
Page .....

Ex - + perform polymorphism if we provide numerical value it perform addition. if we provide character value then it concadinate both character.

4+5 = 9

"4"+"5" = 45