

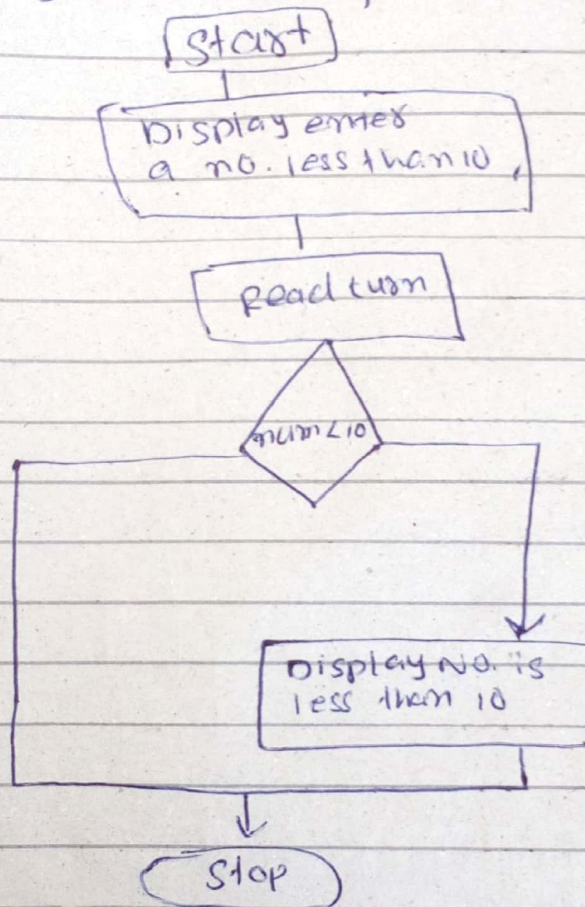
Name Younas Khan
ID 12422
Subject programming fundamental.

Q No. 1: (a) If Statement

If (this condition is true)
execute this statement

- Single-entry / single exit
- Nonzero is true, zero is false.

If statement flow chart-1



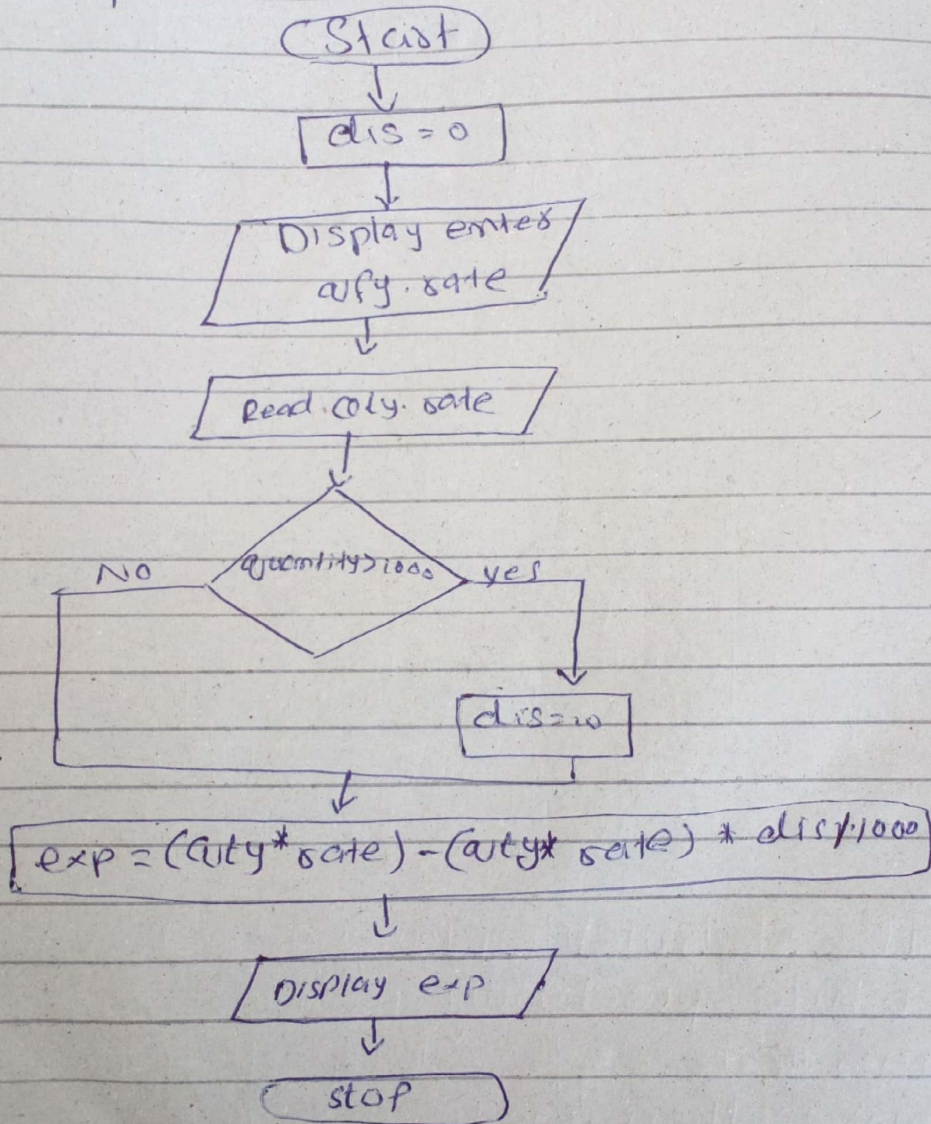
If Statement Flowchart-2

- While purchasing items, a discount of 10% is offered if the quantity is more than 1000. If the quantity and piece per item is input through a keyboard, write a program to calculate and display the total expense.

(2)

Input : quantity, price
processing : calculating discount
and expense

Output : display discount
If Statement Flowchart.



(3)

Quick test

- `if (3 + 2 % 5)`
`cout << "Print : This works";`
- `if (a = 10)`
`cout << "Print : This works";`
- `if (-5)`
`cout << "Print : This works";`
- `if (2 + 2 % 5)`
`cout << "Print : This works";`

* Multiple statement within if

• More than one statement can be executed if the condition in if statement is true.

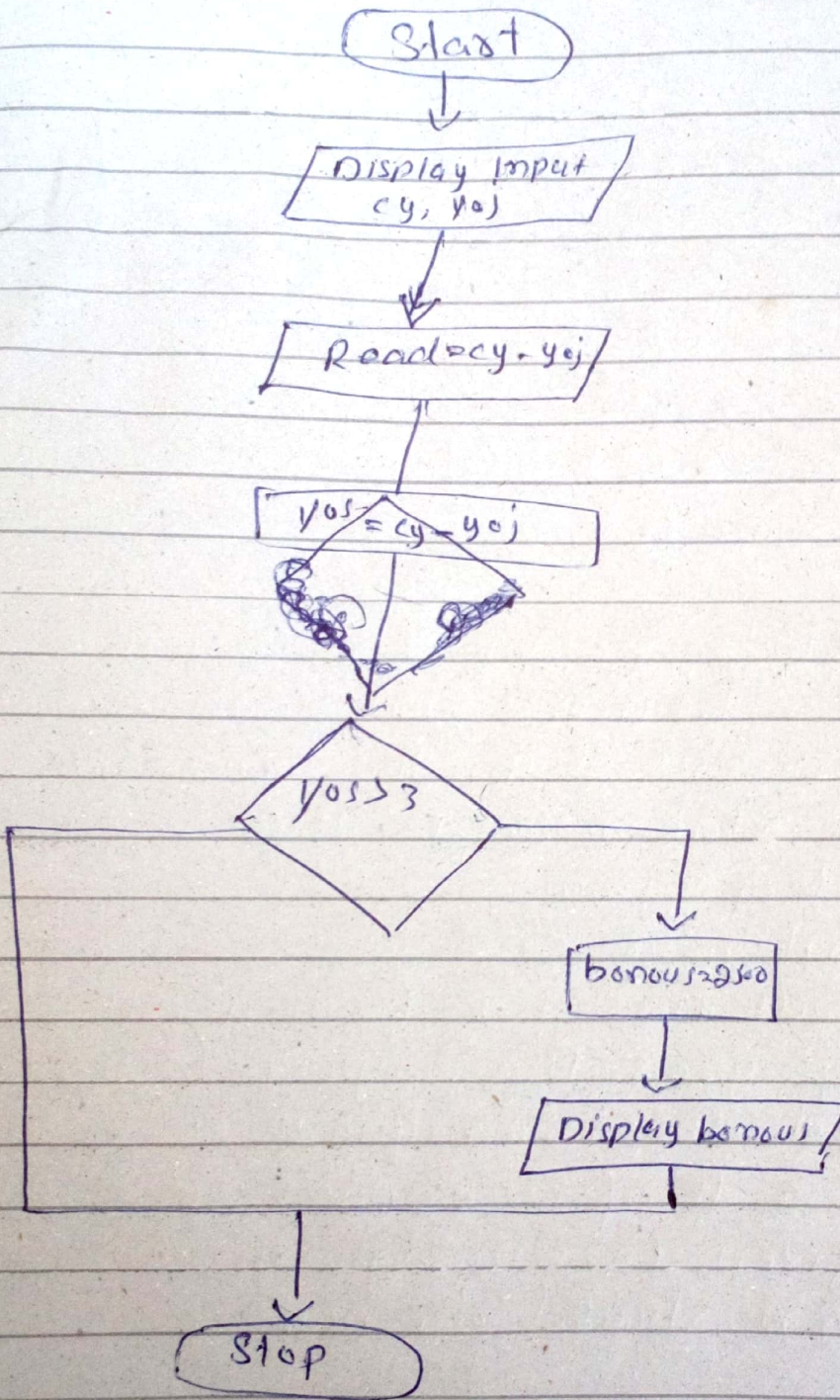
example: program:

current year and year of joining is input through keyboard. If the no of years the employed has served the organization is greater than 3 then a Bonus of Rs. 2500/- is given to employ. otherwise the program should do nothing.

Compound statement

next page

(4)



5

Q No. 1 (b)

```
#include <conio.h>
#include <iostream.h>
class largest
{
    int d;
    public:
    void getdata (void);
    void display_large (largest, largest);
};
void largest :: getdata (void)
{
    cout << "\n\nEnter value: - ";
    cin >> d;
}
void largest :: display_large (largest o1,
                                largest o2)
{
    if (o1.d > o2.d)
        cout << "\n\n object 1 contain largest value
                \n" << o1.d;
    else if (o2.d > o1.d)
        cout << "\n\n object 2 contain largest value
                \n" << o2.d;
    else
        cout << "\n\n both are equal";
}
}
```

next page

8

```
void main()
{
    largest o1, o2, o3;
    classes();
    o1. getdata();
    o2. getdata();
    o3. display - large (o1, o2);
    getch();
}
```

Display =

Enter value :- 19

Enter value :- 89

object 2 containt largest value 89

Q No 2: (A) logical operation

* && (logical AND)

- used to combine two condition

- true if both condition are true.

IF (gender == 1 && age >= 65)

senior ++;

* || (logical OR)

- true if either of condition is true

IF (semesterAvg >= 90 || final exam -

- >= 90)

cout << ("student grade is A");

7

- ! (Logical NOT, logical Negation)
- Returns true when its condition is false, & vice versa.

```
if (! (grade == 20))  
    cout << "hello world";
```

Alternative:

```
if (grade != 20)  
    cout << "hello world";
```

Logical operation

Operands		Results			
x	y	!x	!y	x && y	x y
0	0	1	1	0	0
0	non-zero	1	0	0	1
non-zero	0	0	1	0	1
non-zero	non-zero	0	0	1	1

Sample program

- To calculate the division
- input: marks of 2 different subject.

Q8

- Percentage above or equal to 60 - First division
- Percentage between 50 and 59 - Second division
- Percentage between 40 and 49 - Third division
- Percentage less than 40 - Fail

Sol :-

Nested if else
logical operation

* Using logical operations
A company insures its drivers
in the following cases:

- If the driver is married
- If the driver is unmarried,
male and above 30 years of
age
- If the driver is unmarried,
female & above 25 years of
age

```
if (ms == 'M')
    cout << "Driver is insured";
else
{
    if (sex == 'M')
    {
        if (age > 30)
```


9

```
cout << "driver is not insured";  
}  
else  
{  
    if (age > 25)  
        cout << "driver is insured";  
    else  
        cout << "driver is not insured";  
}  
}  
  
if ((ms == 'M' || (ms == 'U' && sex == 'M' && age > 30)) ||  
    (ms == 'U' && sex == 'F' && age > 25))  
    cout << "driver is insured";  
else  
    cout << "driver is not insured";
```

Gender	years of service	Qualification	Salary
Male	≥ 10	Post-Graduate	15000
	≥ 10	Graduate	10000
	< 10	Post-Graduate	10000
	< 10	Graduate	7000
Female	≥ 10	Post-Graduate	12000
	≥ 10	Graduate	9000
	< 10	Post-Graduate	10000
	< 10	Graduate	6000

(10)

Operators	Type
!	logical Not
* / %	Arithmetic and modulus
+ -	Arithmetic
< > <= >=	Relational
= = ! =	Relational
& &	logical AND
	logical OR
=	Assignment.

~~Operator~~

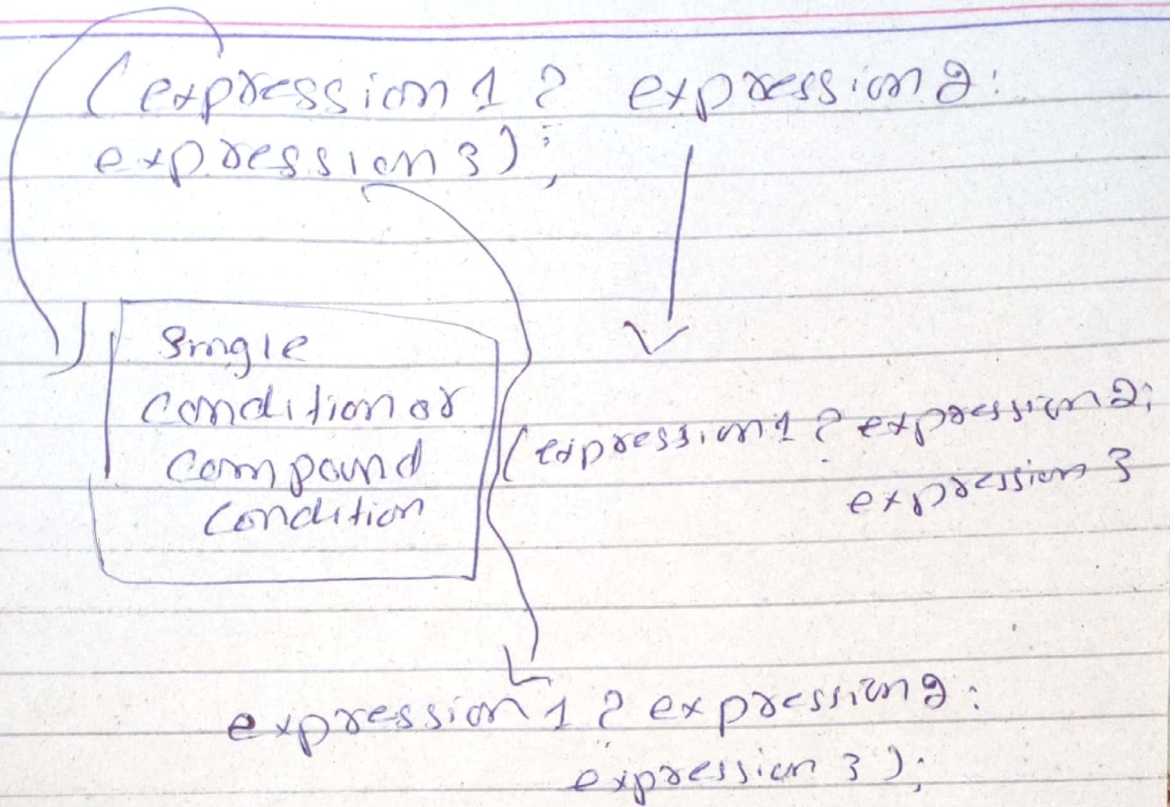
* Conditional operators
General form is,
expression 1 ? expression 2
expression 3 ;

Conditional operators ? and
are sometimes called ternary
operators

• If expression 1 is true, then
the value returned will be
expression 2; otherwise the
value of returned will be expression
3.

* Nested conditional
operator :-

11



QNO 2 B:

```
#include <conio.h>
#include <stdio.h>
void main()
{
    int temp;
    printf("input day's temperature:");
    scanf("%d", &temp);
    if (temp < 0)
        printf("freezing weather.\n");
    else if (temp < 40)
        printf("very hot weather.\n");
    else if (temp < 35 & 40)
        printf("Tolerable weather.\n");
```


(12)

```
else if (tmp < 30 & 35)
    printf("warm weather.\n");
else if (tmp < 30)
    printf("its cool.\n");
else
    printf("cool.\n");
}
```

QNO3:- (A)

loop structure is used to execute a statement or statement repeatedly.

Three major loops structures in C++.

- while loop
- for loop
- do while loop
- loop has a termination point → finite loop
- loop execution stops when the loop condition become false
- loop has a counter that count number of iterations of that loop.
- loop has statement or a set of statements, that are executed until the loop condition become false

(13)

* while loop: The syntax of a while loop in C++ is - while loop (condition) (statements); } These statements may be a single statement or a block of statements. The condition may be any expression and true is any non-zero value. The loop iterates while the condition is true.

* For loop: A loop is used for executing a block of statements repeatedly until a particular condition is satisfied. For example: when you are displaying numbers from 1 to 100 you may want set the value of a variable to 1 and display it 100 times, increasing its value by 1 on each loop iteration.

* do-while-loop: The do while loop. This loop will execute the code block once, before checking if the condition is true then it will repeat the loop as long as the condition is true.

14

Q No 3 (B):

```
#include <conio.h>
#include <stdio.h>
int main()
{
    int num;
    printf("Enter an
    integer number :");
    scanf("%d", &num);
    /* If number is divisible
    by 2 then number is even
    otherwise number is odd */
    if (num % 2 == 0)
        printf("%d is an Even number.",
        num);
    else
        printf("%d is an odd number.",
        num);
    return 0;
}
```

Output -

Enter an integer number : 123
123 is an odd number.

15

```
#include <conio.h>
```

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

```
int main ()
```

```
{
```

```
int num
```

```
printf ("enter an integer no  
no:");
```

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

```
if number is divisible  
by 2 then number is even  
otherwise number is odd
```

```
if (num % 2 == 0)
```

```
printf ("%d is an even no", num);
```

```
else
```

```
printf ("%d is an odd no.", num);
```

```
return 0;
```

```
}
```

```
getch
```

```
Enter an integer: 10
```

```
10 is even number.
```


(16)

QNO 4:- (a) The break statement:

These are situations where we want to jump out of a loop instantly, without waiting to get back to the conditional test.

- The keyword break allow us to do this
 - When break is encountered inside any loop, control automatically passes to the first statement after the loop.
- A break is usually associated with an if
 - The keyword break the control only from the loop in which it placed.

* The Continue Statement:

Continue statement allows to take the control to the beginning of the loop, bypassing the statement inside the loop, which have not yet been executed.

A continue is usually associated with an if.

(17)

```
#include <studio.h>
main()
{
    int i, j;
    for (i = 1 : i <= 2 : i++)
    {
        for (j = 1 : j <= 2 : j++)
        {
            if (i == j)
                continue;
            cout << " | n^n. i.j);
        }
        System(" pause");
    }
}
```

QNO4 (b) :-

```
#include <iostream.h>
#include <conio.h>
void main()
{
    clrscr();
    int i, n, sum = 0;
    cout << "1+2+3+.....+n";
    cout << " n Enter the value of n (0!)"
    cin >> n;
    for (i = 1; i <= n; ++i)
        sum += i;
    cout << " n sum = " << sum;
    getch();
}
```


(18)

output =)

1+2+3+...+n

Enter the value of n: 10

sum = 55

Q105:-

Q C++ character set

In C++ character set of all valid character that can be used in C++ program.

Character set is used to

Specify the character or symbols recognized by the language.

Character set is a set of all valid characters that can be used to form words, numbers and expression in source program.

Alphabets

A, B, ... Z

a, b, ... z

Digits

0, 1, 2, 3, 4, 5, 6, 7, 8, 9

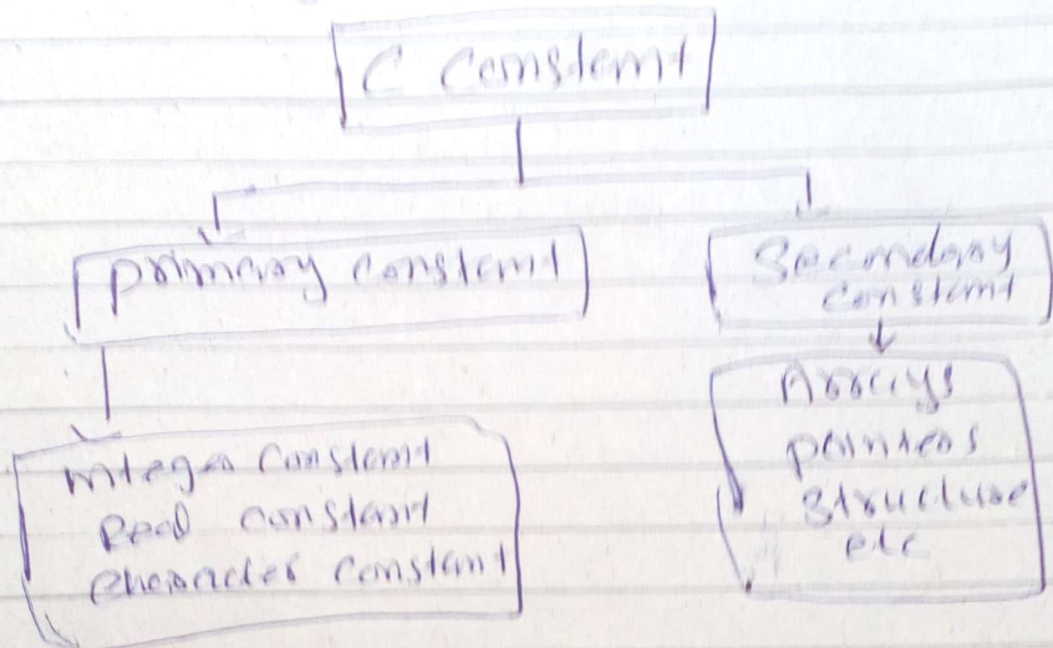
special symbols

~ ! # \$ % & * () _ = +

{ } [] ; : " ' < > ? , . / \

(19)

Ⓐ * Constant: constant is an entity that doesn't change.



* Integer constant:

- must have at least one digit
- It must not have a decimal point
- It can be either positive or negative
- If there is no sign on integer constant is assumed to be positive.
- No commas or blanks are allowed within an integer constant.

Example: 135, -67, 3401, -5670.

* Real Constant

Also called floating point constant.

A real constant must have at least one digit.

It must have a decimal point. It could be either positive or negative.

Default sign is positive.

No commas or blanks are allowed within a real constant.

Examples:- +325.34, 426.0, -32.76
-48.5792.

* Character Constant:-

A character constant is a

- single alphabet
a single digit
or a single special symbol enclosed within single inverted commas. "

• Both the inverted commas should point to left.

The maximum length can be 1 character.

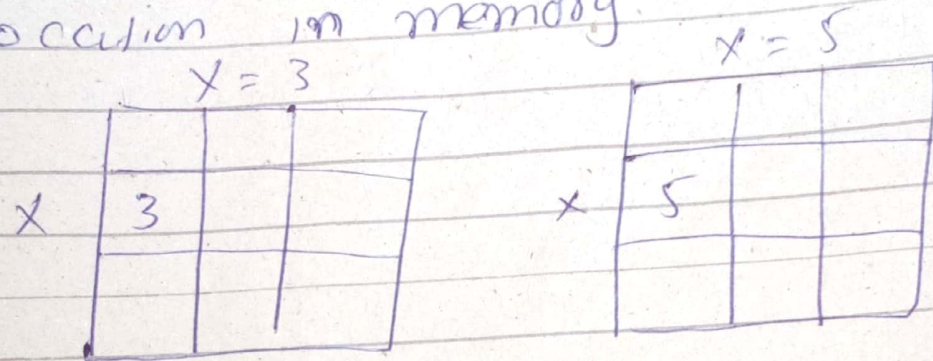
* Example:- "A", "1", "5", "="

(21)

(C) Variables :-

An entity that may change during program execution.

These are names given to location in memory.



Series of characters (letters, digits, underscore) must begin with a letter or underscore.
case sensitive
meaningful naming scheme.

No commas or blanks are allowed within a variable name.

No special symbols are used in name.

• Example: - Integer 1, sum first num.

• Invalid variable names

- #sum, 12x, first name.

a) Key words :-

These are reserved word
Compiler knows their meaning
cannot be used as variable name
cannot be change

auto	double	int	struct
break	else	long	switch
case	enum	register	typedef
char	extern	short	union
const	float	signed	unsigned
continue	for	sizeof	void
default	goto	sizeof	volatile
do	if	static	while

Programs consists of instructions
as statement.

Statement must be in order
each statement must end with a ;
Blanks for readability or clarity
case sensitive

② Relational operators:-

A relational operator is a programming language construct or operator that tests or defines some kind of relation between two entities. These include numerical equality.

eg: $5 = 5$ and inequalities eg: $4 \geq 3$

Standard algebraic Relational operator	C++ equality	example	meaning
$>$	$>$	$x > y$	x is greater than y
$<$	$<$	$x < y$	x is less than y
\geq	\geq	$x \geq y$	x is greater than or equal to y
\leq	\leq	$x \leq y$	
Equality operators			
$=$	$==$	$x == y$	
\neq	$!=$	$x != y$	

meaning

x is greater than y

x is less than y

x is greater than or equal to y

x is less than or equal to y

x is equal to y

x is not equal to y.