Name	Khalid Khan
ID	7936
Section	B
Sub	CP
Submitted to	Engr-Ashraf Ali

M T W T F S

Date:

Q = 1:-

Part = A:-

Grade Statment program:-

```
#include <iostream>
using namespace std;
int main( );
Cout<<"program to find grades"<<endl;
cout <<"\nEnter marks:";
cin >>marks;
if (marks>=90 && marks<=100)
cout<<"your grade is A";
else if(marks>=80 && marks<=90)
cout <<"your grade is A";
else if (marks>=70 && maoks<=80)
cout <<"your grade is B";
else if (marks>=60 && marks<=70)
cout <<"your grade is c";
else if (marks>=50 && marks<=60)
cout <<"your grade is D";
else if (marks>=0 marks<50)
Cout <<"your grad is F";
else cocet <<"invalid marks";
return 0; }
}
```

Date: M T W T F S

Part = B :-

Different :-

"If Statement" :-

Sometime we want to Selectively execute a block of code.

⇒ The C++ Syntax of the if Statment is;

```
if (logical expression)
{
// Block of code to excute
if expression is true
}
```

⇒ When expression is true, the block of code is executed.

⇒ When expression is flase, the block of code is Skipped.

⇒ The block of code should be indented 3-4 spaces to aid program readability.

⇒ If the block of code is only one line long the {} braccets can be omitted.

⇒ The empty statment b/w) & ; will be selectively executed based on the logical exp value.

The block of code directly below if statment will always be excuted, which is probably not what you intedad.

Date: M T W T F S

if-Else Statment:-

Sometimes we need to handle two alternatives in our code.

⇒ The C++ syntax of the if-else statment is;

```
if (logical expression)
{
// Block of code of excute if expression is true.
}
else
{
// Block of code of excute if expression is false.
}
```

⇒ Type the "if line" and the "else line" and the {} brackets so they are vertically aligned with each other

⇒ Do not put a semi-colon after the "if line" or the "else line" or you will get very strange run time errors.

⇒ The two blocks of code should be indented 3-4 spaces to aid Program readability.

M T W T F S

Date:

Q = 2 :-

Part = A :-

Switch statment :-

```
#include<iostream.h>
# include<conio.h>
main ( )
{
clacr ( );
int [redacted] clscr.
Cout <<" program to input dates
<endl;
Cout : Program to input.
```

M T W T F S

Date:

Q = 21-

Part = B:-

Difference b/w "Nested if-else" & "switch" Statement:-

Nasted if-else :-

⇒ it becomes complicated for multiple Selections.

⇒ if uses an independent expression for each side.

⇒ The test condition can be given is a speeral range of value. if the given condition matches then the statments under it will be excuted.

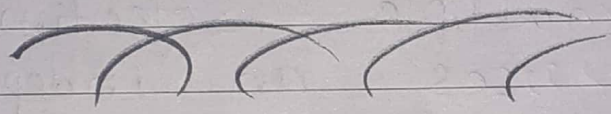
Switch Statments:-

⇒ it is easy to understand for multiple selection.

⇒ it uses a single expression for all cases, but each case must have a constant value of integer type or character type.

Date: M T W T F S

⇒ only a single expression is given in the switch statment which returns a single value - the ~~test~~ test condition cannot be given in a specified range. it is drawback.



M T W T F S

Date:

Q = 3 L

A :-

Relational Expression :-

⇒ Relational expression evalute to the integer value 1 (true) or 0 (false).

⇒ All of these operations are called binary operations because they take two expression as operands.

⇒ Relational operator (==) is used to campare 2 value whether they are equal are not.

⇒ if both value are equal output is displayed as "value are equal" Else, output is displaged as "Values are not equal".

⇒ Double equal sign should be used to compare 2 values, we should not single equal sign (=).

Date: M T W T F S

Relational operator :-

⇒ A Relational operater compares two values.

⇒ value can be any built in C++ data type.

⇒ The comperison involves such relationship as equal to, less then and greater then.

⇒ The result of the comparison is either true or flase.

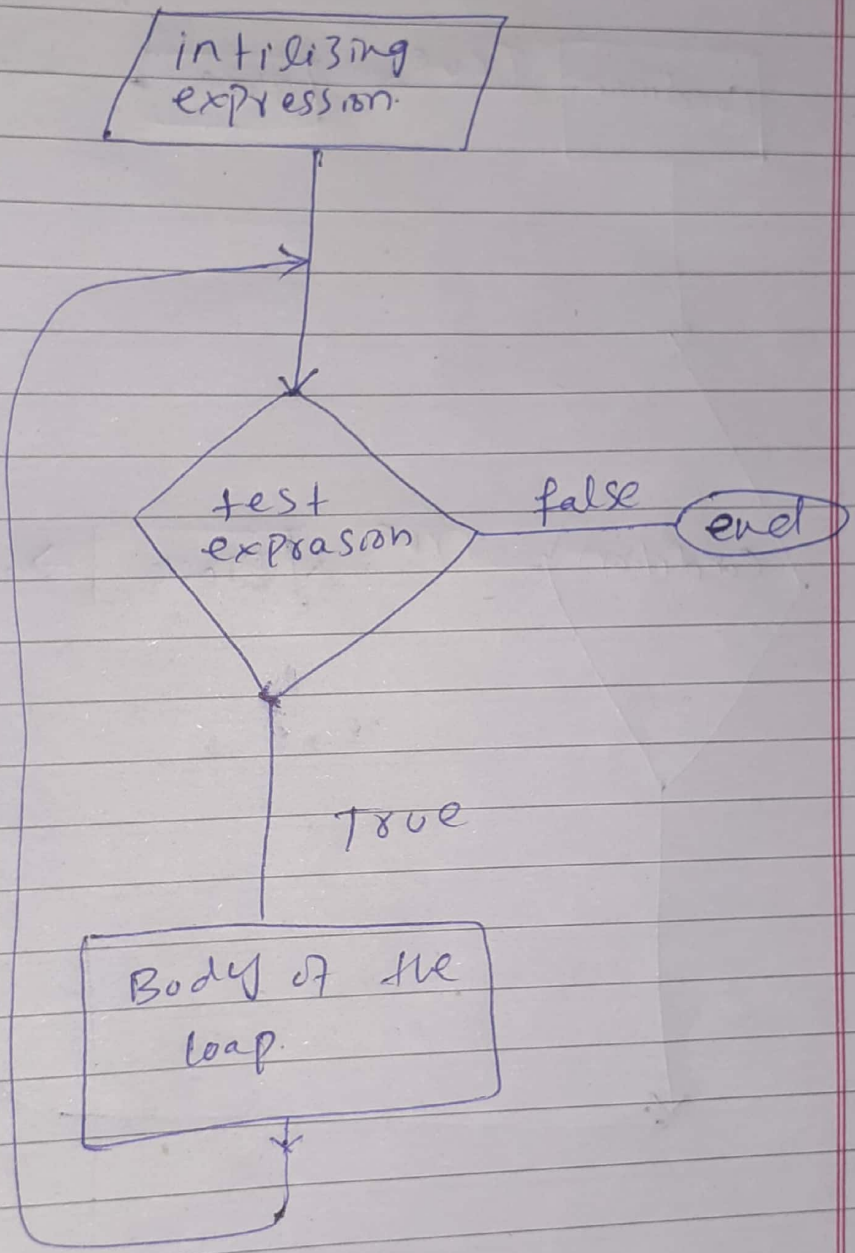
⇒ < less then
⇒ > greater then
⇒ <= less then or equal to
⇒ >= greater then or equal to
⇒ == is equal to
⇒ != is not equal.

M T W T F S

Date:

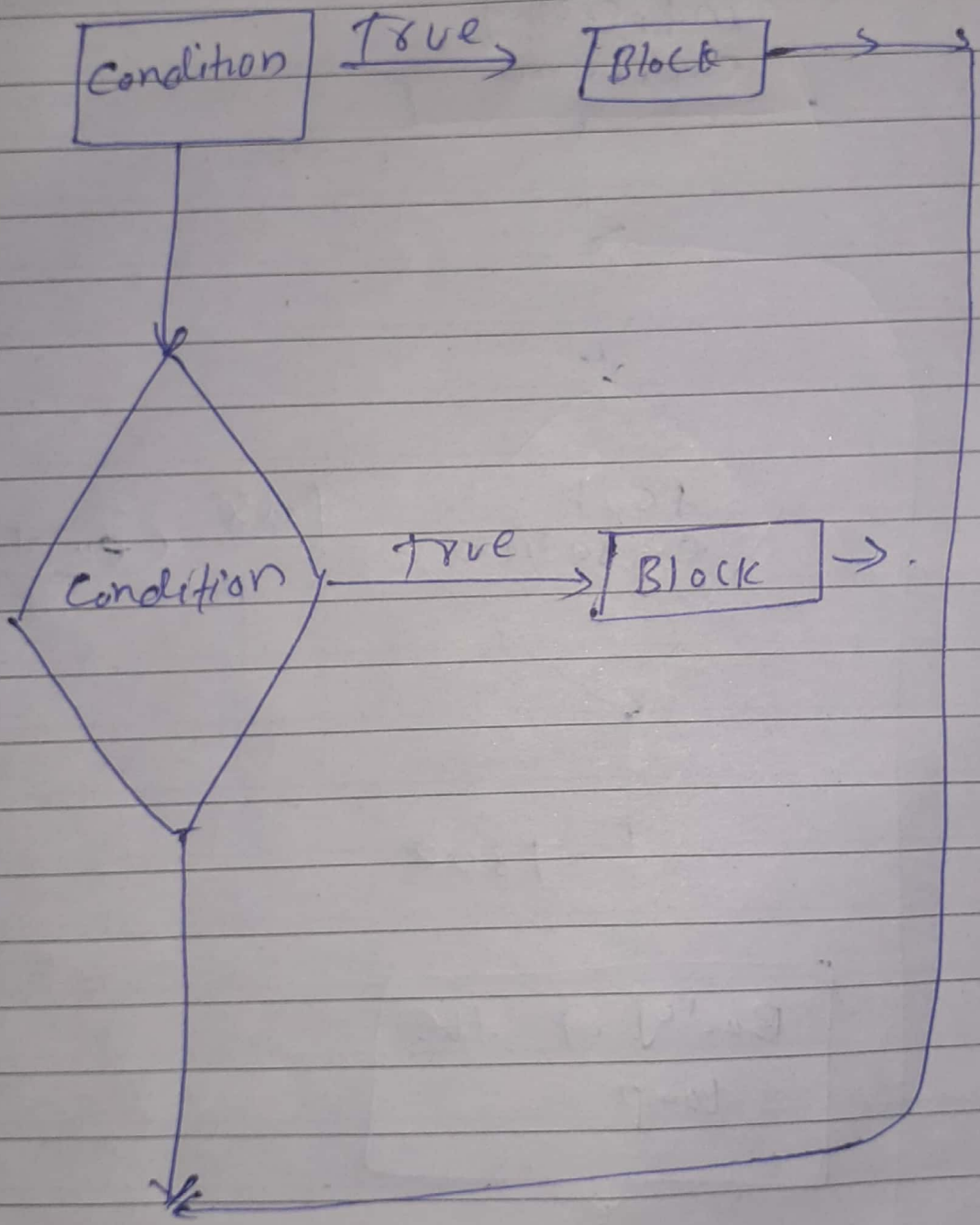
B:-

while loop chart:-



Date: M T W T F S

flow chart of Nested if else :-



M T W T F S Date:

Q = 4 :-

A :-

```
#include <iostream>
using namespace std;
int main ()
{
int rad1, hgt;
float volcy;
cout<<"\n\n calculate the volume
of a cylinder:\n";
cout<<"\n\n calculate the volume
of a cylinder:\n";
cout<<"input the radius of the
cylinder:";
cin >> rad1;
cout<<"input the height of
the cylinder:";
cin >>hgt;
volcy = (3.14 x rad1 x rad1 x hgh);
cout<<"the volume of a cylinder
is :"<< volcy <<endl;
cout<<endl;
```

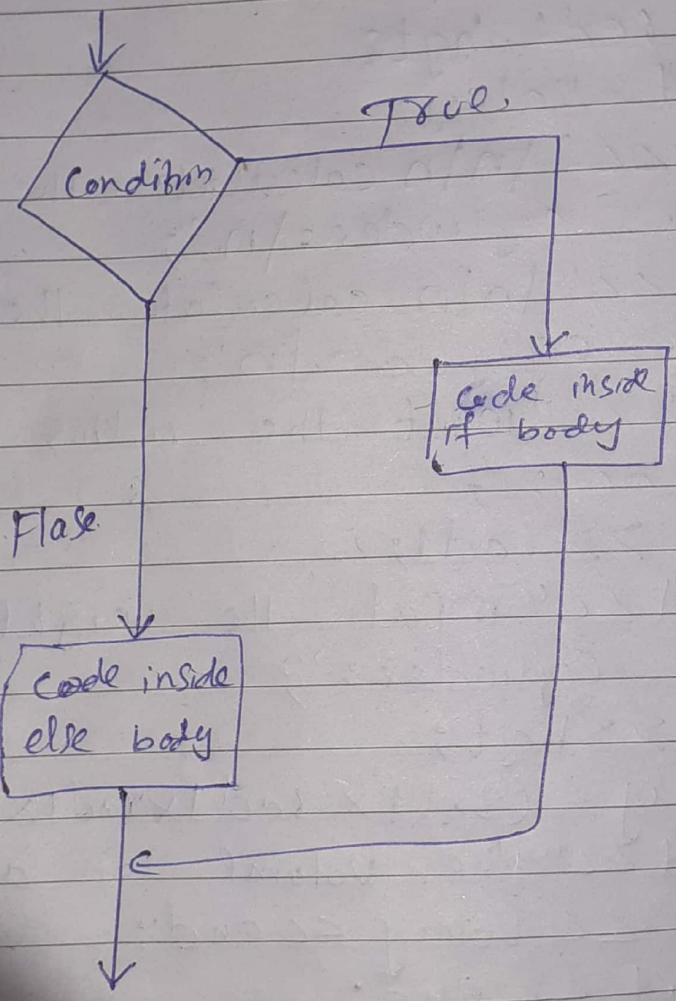
M T W T F S

Date:

Q=4:-

B

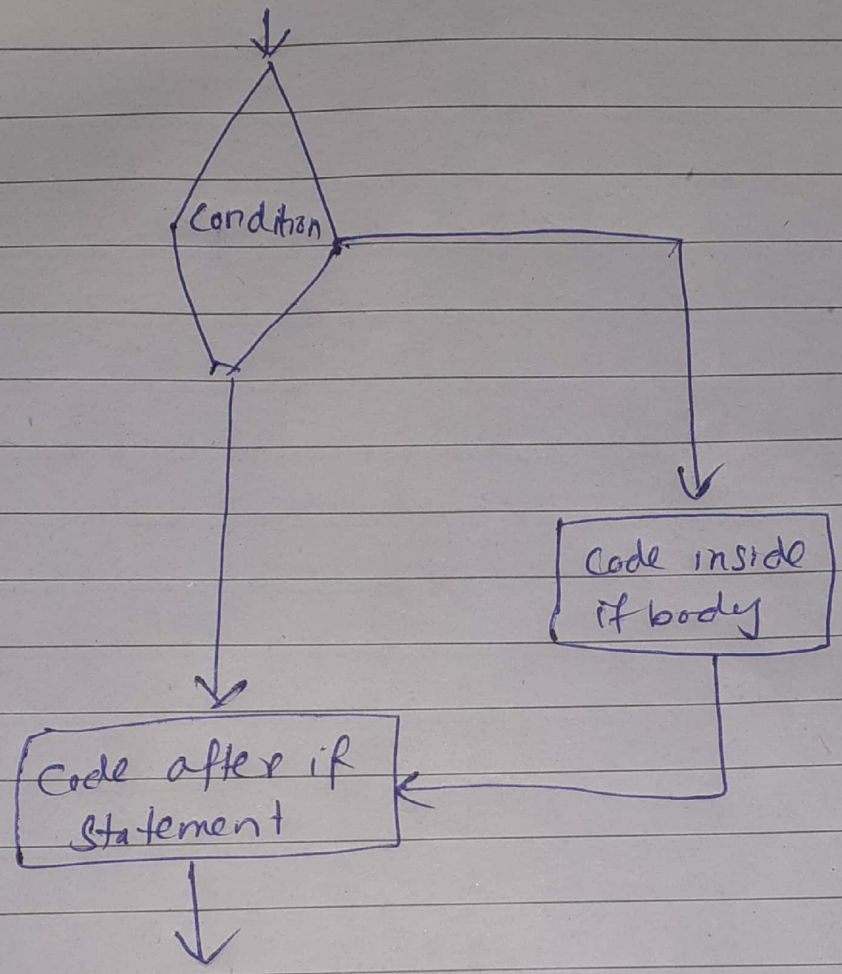
flow chart of if else statment :-



M T W T F S

Date:

if Statment :-



M T W T F S Date:

Q=5:-

A

Sequential statment:-

Sequential statment define algorithms for the execution within a process or a subprogram. they belong to the conventional nation of Sequential flow, control conditionals, and iterations such as high level programming languages such as pascal, C, or Ada. They excude in order in which they appear in the process-

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
#include <math.h>
void main()
 {
 float farenhite, celsius;
 printf("enter degree in far\n");
 scanf("%f", &farenhite);
 celsius= (float) 5/9 * (frenhite-32);
 Print("result=%f\n", celsius);
 getch();
```

Date: M T W T F S

Execution begins from the main, we enter a ferenhite degree value. control moves to the formula which will convert ferenhite into celcius and the equivlent celsius value is output.

Processes:-

A processes is a sequence of statments that are the excuted in the specified order. The process declaration delmits a sequantial domain of the architecture in which the declaration appers. processes are used for behavioral description.

Notes:-

In a processes it is not allowed to declare singles, only statments and variable may be declared.

M T W T F S

Date:

Q = 5 :-

B :-

```
#include <stdio.h>
#include <math.h>
int main ( )
  {
double first_number, second_number;
double addition, substraction, multiplication,
division, modulus;
PrintF ("Enter the first numbers to
perform Arthimetic operation");
ScanF ("%If", & first_number);
PrintF ("Enter the second number to
perform Arthemetic operation ";
Scanf ("%If", & Second_number);
addition - first_number + Second number;
Substraction = first-number - Second num;
Multiplication = fiost number x sec_num;
division - first number / Second number;
modulus = fnod (first_number, Second number);
Printf {"\n\sum of two nembers are =
\t\t%1f\n", addition);
Printf {"Differences of two number are=
\t\t%1f\n", addition);
Printf {" multiplication of two number are =
\t\t%1f\n", multiplication;
Printf {"Quotients of two number are =
\t\t%1f\n", division;
```

STAR BOOK AGENCY

Date: | | | | | | |

M | T | W | T | F | S

```
Printf("modulus of two numbers are=
\t\t%d\n", modulus);
return 0;
```