

Name Fawad Khan

Sec/ID 7875/13

Subject C++

Teacher Engr Syed Ashraf Ali

Summer

# PROGRAM FOR GRADING SYSTEM

```

#include <iostream>
using namespace std;
int main ()
{ int marks;
cout << "\n Program to find grad —
— " << endl;
cout << " \n Enter marks ";
cin >> marks;
if (marks >= 88 && mark <= 100)
cout << " your Grad is A. ";
else if (marks >= 81 && mark < 88)
cout << " your Grad is B+. ";
else if (mark >= 74 && mark < 81)
cout << " your Grad is B. ";
else if (marks >= 67 && mark < 74)
cout << " your Grad is C+. ";
else if (mark >= 60 && mark < 67)
cout << " your Grad is C. ";
else if (marks >= 50 && mark < 59)
cout << " your Grad is F. ";
else cout << " invalid Marks. "; return 0; }

```

Q No: 1

Part 'b'

Differentiate Between "IF" And  
"IF Else" statement

1 IF Statement

Sometimes we want to selectively execute a block of code

2 When expression is true the block of code is executed.

3 when expression is false the block of code is skipped.

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

5 Never put a semi-colon directly after the Boolean expression in and if-statement

IF-Else Statement

1 sometimes we need to handle two alternatives in our code

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

if either block of code is only one line long the { } brackets can be omitted

We can visualize the program if-else statement decision process using a flow chart

Q No 2

Part A

```
#include <iostream>
```

```
using namespace std;
```

```
int main (void)
```

```
{
```

```
    char selection
```

```
    cout << "\n main";
```

```
    cout << "\n =====";
```

```
    cout << "\n A Append";
```

```
    cout << "\n M Modify";
```

```
    cout << "\n D Delete";
```

```
    cout << "\n X Exit";
```

```
    cout << "\n Enter selection";
```

```
    // read the input
```

```
    cin >> selection;
```

```
    switch (selection)
```

```
    {
```

```
        case 'A' :
```

```
        case 'a' : { cout << "\n To append a read \n"; }
```

```
        break;
```

```
Case(M);
```

```
Case(M): { cout << "\n To modify a record ";  
break ;
```

```
// Other than A, M, D and X ---
```

```
default; cout << "\n invalid selection";
```

```
// no break in the default case
```

```
}
```

```
cout << "\n";
```

```
return 0;
```

```
}
```

Q 2

Part (b)

## Difference between "Nested IF Else" And "Switch" statement

	Nested IF-Else statement	Switch Statement
i	It becomes complicated for multiple selections	It is easy to understand for multiple selections
ii	It uses an independent expression for each case	It uses a single expression for all cases, but each case must have a constant value of integer or character type.
iii	The test condition can be given in a special range of value. If the given condition matches then the statements under it will be executed.	Only a single expression is given in the switch statement which returns a single value. The test condition cannot be given in a specified range. It is drawback.

iv) It is difficult to edit the If-else statement if the nested if-else statement is used

It is easy to edit switch cases as they are recognized easily.

## Q3 Part 2)

### Relational Operator:-

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 (e.g.,  $5=5$ ) and inequalities (e.g.,  $4 < 3$ ).

### Relational Expression:-

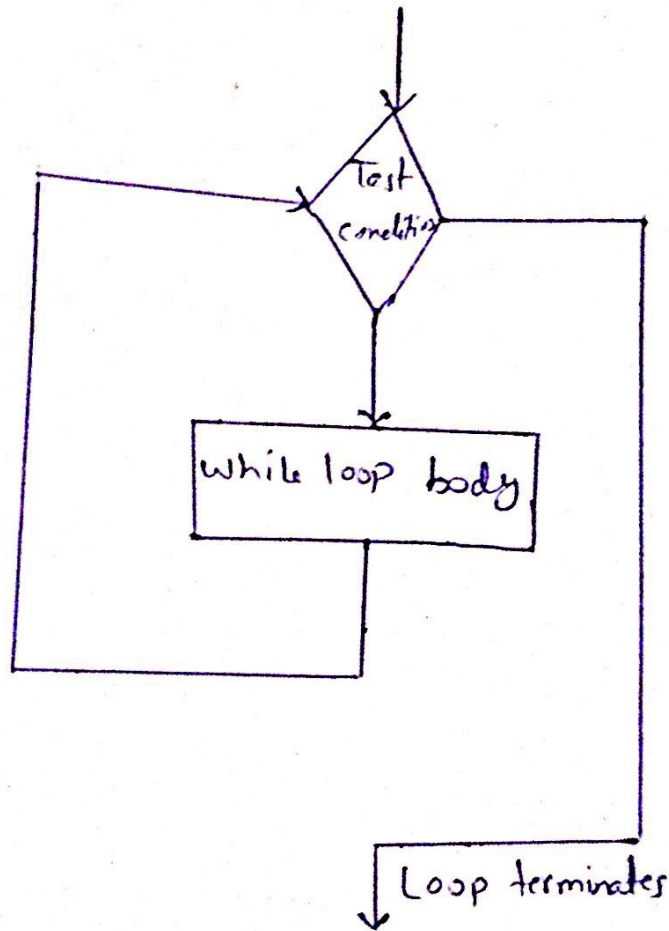
A relational expression consists of two arithmetic expressions or two character expressions separated by a relational operator. A relational operator tests for a relationship between the two expressions. The value of the relational expression is either True or False. Depending on whether the stated relationship holds.



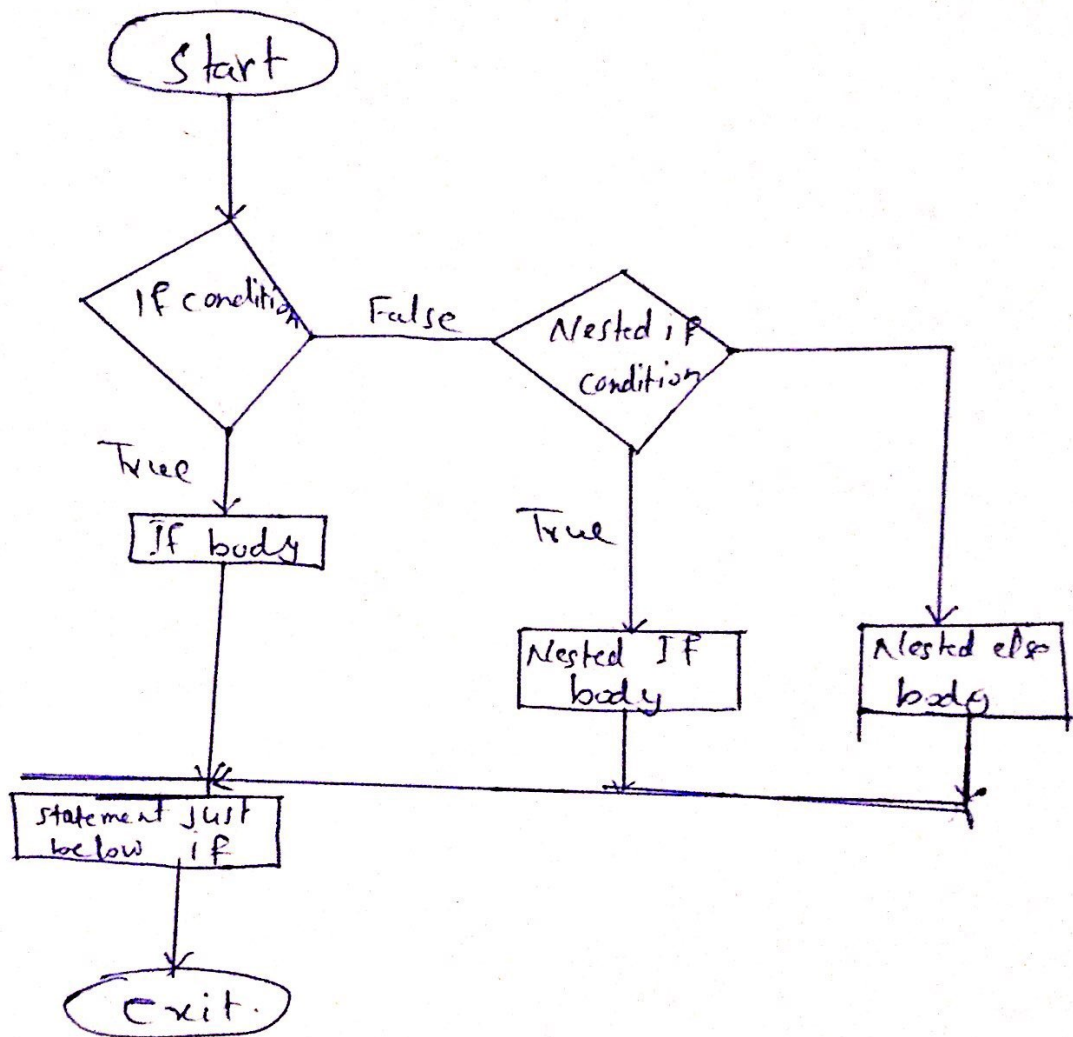
Q 3

Part "b"

Flow chart of while Loop



# Flowchart of Nested IF Statement



Q No 4

(a)

```
#include <iostream.h>
```

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

```
main()
```

```
{
```

```
clrscr();
```

```
float radius, height, volume;
```

```
cout << "Enter the radius = ";
```

```
cin >> radius;
```

```
cout << "Enter the heigh = ";
```

```
cin >> height;
```

```
Volume = 3.14 * r * r * h;
```

```
cout << "Volume of the cylinder = " << Volume;
```

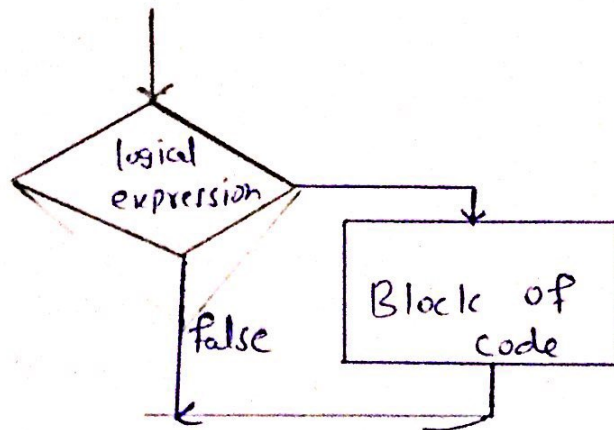
```
getch;
```

```
}
```

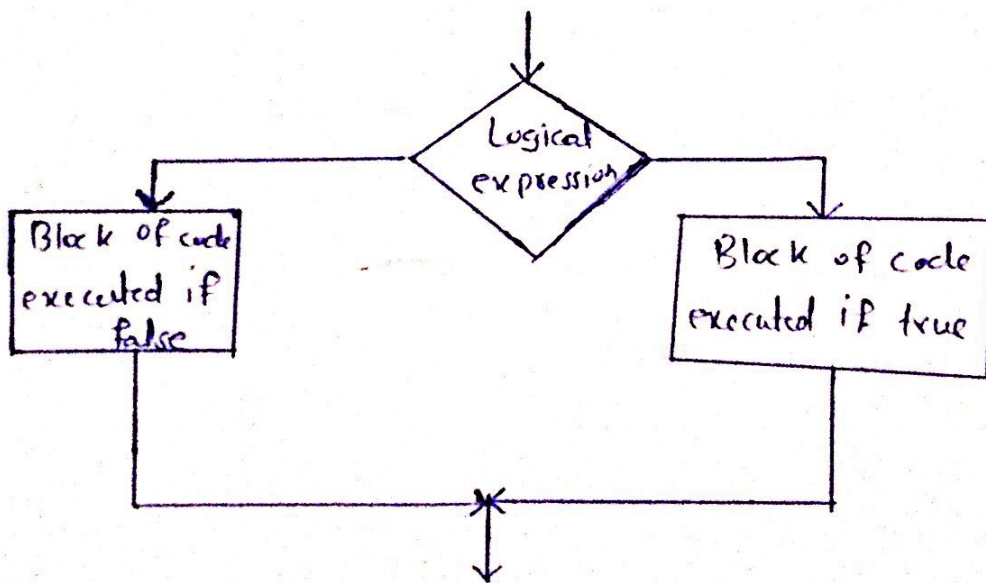
Q4

Part "b"

# Flow Chart For "IF Statement"



# Flow chart For "IF-Else" Statement



Q&A

Part A

## Sequential Statement:-

Sequential Statement are assignment statements that assign value to variables and signals. Flow control statements that conditionally execute statements (if and case)

To Familiarize Yourself with Sequential Statements, consider the following:-

- 1 Assignment Statement
- 2 Variable Assignment Statement
- 3 Signal Assignment Statement
- 4 if statement
- 5 case statement
- 6 Loop statement
- 7 Next statement
- 8 Exit Statement.

# Arithmetic Operation by using Switch statement

```
#include <iostream.h>
```

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

```
main()
```

```
{
```

```
clrscr();
```

```
int a, b;
```

```
char operation
```

```
cout << "Enter the first number, operator and  
then second number = \n";
```

```
cin >> a >> operation >> b;
```

```
switch (operation)
```

```
{
```

```
case '+':
```

```
cout << "Addition = " << (a+b);
```

```
break;
```

```
case '-':
```

```
cout << "subtraction = " << (a-b);
```

```
break;
```

```
case 'x':
```

```
cout << "multiplication = " << (a*b);
```

```
break;
```

case '1':

cout << "Division = " << (a/b);

default:

cout << "Invalid input";

}

getch();

}