

NAME:-

NAVEED AHMAD

I.D. :-

7880

SUBJECT:-

C++

EXAM:-

SUMMER - FINAL

QUESTION- 1-a

①

PROGRAM FOR GRADING SYSTEM.

```
#include <iostream>
using namespace std;
int main()
{
    int marks;
    cout << "----- Program To Find Grad -----" << endl;
    cout << "\n Enter Marks ";
    cin >> marks;
```

```
if (marks >= 88 && marks <= 100)
    cout << "Your Grad is A.";
else if (marks >= 81 && marks < 88)
    cout << "Your grade is B+.";
else if (marks >= 74 && marks < 81)
    cout << "Your Grad is B.";
else if (marks >= 67 && marks < 74)
    cout << "Your Grad is C+.";
else if (marks >= 60 && marks < 67)
    cout << "Your Grad is C.";
else if (marks >= 50 && marks < 60)
    cout << "Your Grad is D.";
else if (marks >= 0 && marks < 50)
    cout << "Your Grade is F.";
else cout << "invalid Marks."; return 0; }
```

QUESTION- 1-b

2

IF STATEMENT:-

If statement is the most simple decision making statement. It is used to decide whether a certain statement or block of statements will be executed or not i.e. if a certain condition is true then a block of statement is executed otherwise not.

Syntax

```
if (condition
```

```
{
```

```
    // statement to execute if
```

```
    // condition is true
```

```
}
```

(3)

IF ELSE STATEMENT:-

The if statement alone tells us that if a condition is true it will execute a block of statement and if the condition is false it won't. but what if we want to do something else if the condition is false. Here comes the else statement. We can use the else statement with if statement to execute a block of code when the condition is false.

Syntax

```
if (condition
{
    // Executes this block if
    // condition is true
}
else
{
    // Executes this block if
    // condition is false
}
```

(4)

QUESTION- 2-a

```
include < iostream >
```

```
using namespace std;
```

```
int main(void
```

```
{
```

```
char selection;
```

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

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

```
cout << "\n M - modify";
```

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

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

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

```
cin >> selection;
```

```
switch (selection)
```

```
{
```

```
{
```

case 'A':

5

```
case 'a': { cout << "\n To appen a record\n"; }
```

```
break;
```

```
case 'M':
```

```
case 'm': { cout << "\n To modify a record"; }
```

```
break;
```

```
case 'D':
```

```
case 'd': { cout << "\n To delete a record"; }
```

```
break;
```

```
case 'X'
```

```
case 'x': { cout << "\n To exit the menu"; }
```

```
break;
```

```
// other than A, M, D, and X
```

```
default: cout << "\n invalid Selection";
```

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

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

```
return 0;
```

```
}
```

QUESTION - 2 - "b"

DIFFERENTIATE BETWEEN.

Nested if Else Statement

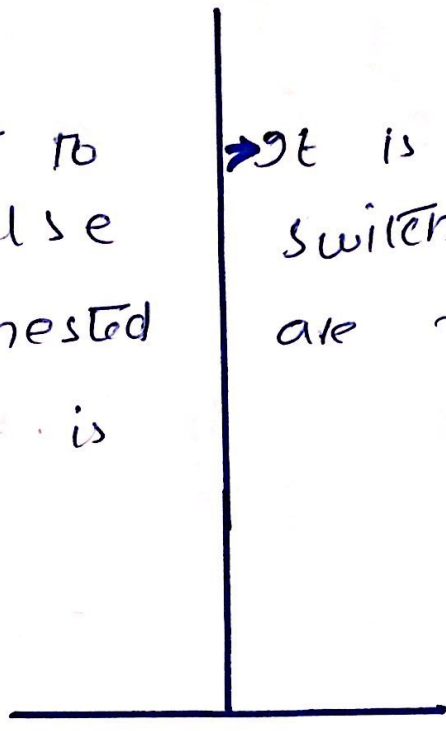
- It becomes complicated for multiple selection
- if uses an independent expression for each case
- The test condition can be given in a special range of value, if the given condition matches then the statement under it will be executed.

Switch Statement

- 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 integers or character type.
- Only a single expression is given in the switch statement which returns a single value. The test condition can not be given in a specific range it is drawback.

→ It is difficult to edit the if else statement if nested if else statement is used

→ It is easy to edit switch case as they are recognized easily



[Faint, illegible handwritten notes in red ink]

QUESTION- 3-a

RATIONAL OPERATOR:

A relational operator compare two values.

Values can be any built in C++ data type

The comparison involves such relationship as equal to less than or greater than

The result of comparison is either true or false.

RATIONAL EXPRESSION:

9

Evaluate to true integer value 1 (true) or 0 false.

All of these operators are called binary operators because they take two expressions as operands.

Rational operator ($==$) is used to compare 2 value whether are equal or not.

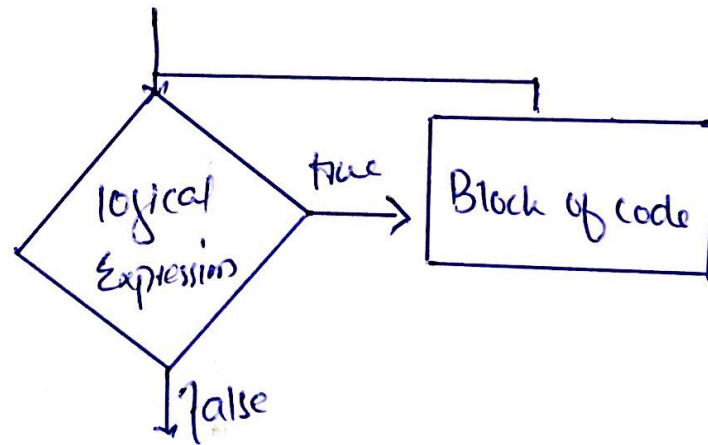
If both values are equal, output displayed as "value are equal" Else output is displayed as "value are not equal"

(10)

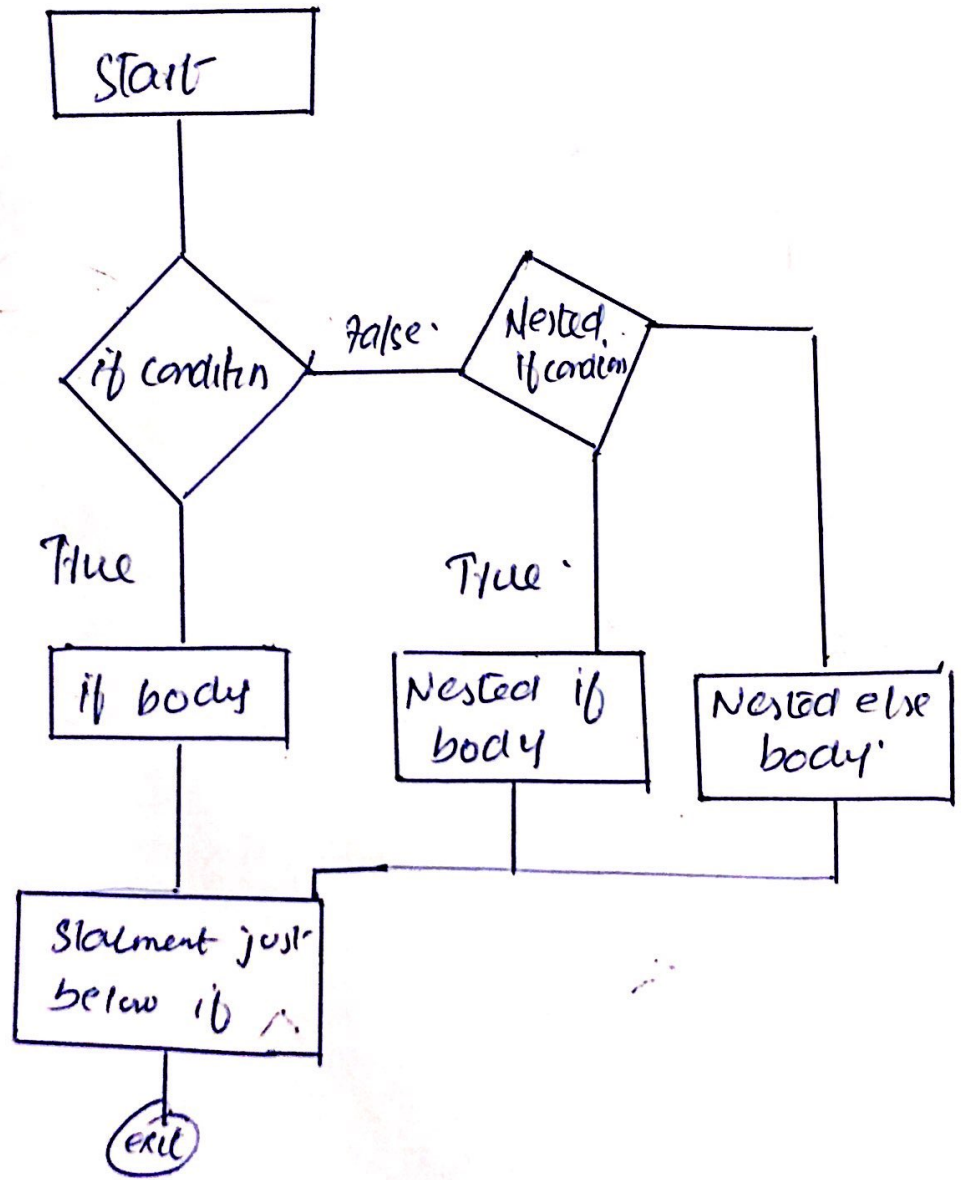
QUESTION- 3- b

FLOW CHARTS.

WHILE LOOP.



IF NESTED STATEMENT:



QUESTION - 4. a.

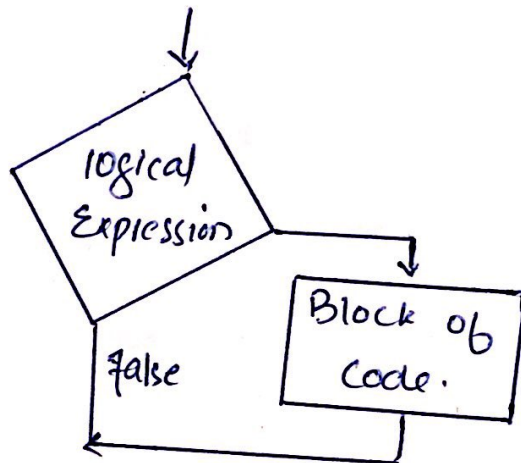
(12)

```
#include <iostream.h>
#include <conio.h>
main()
{
  clrscr();
  float radius, height, volume;
  cout << "Enter radius = ";
  cin >> radius;
  cout << "Enter the height = ";
  cin >> height;
  volume = 3.14 * r * r * h;
  cout << "volume of the cylinder = " << volume;
  getch();
}
```

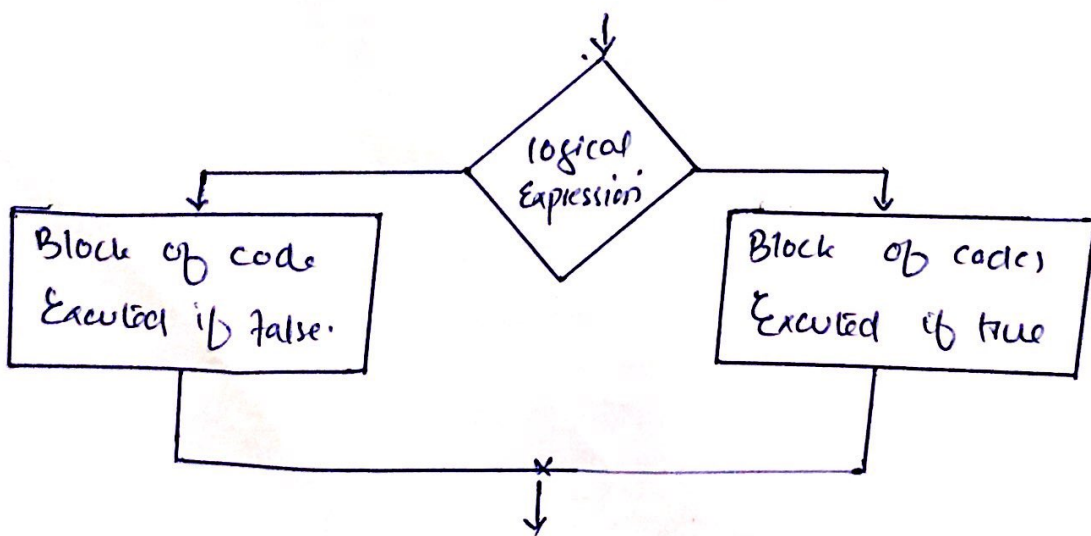
QUESTION- ~~Q.10~~ 4-b

13

FLOW CHART FOR "IF STATEMENT."



FLOW CHART FOR "IF ELSE STATEMENT"



QUESTION- 45a

SEQUENTIAL STATEMENT:

Sequential statements are assignment statement that assign values to variables and signals flow control systems that conditionally executes statement (if and case)

To familiarise yourself with sequential statement consider the following

Assignment statement

Variable assignment statement

Signal assignment statement

if statement

case statement

Loop statement

Next statement

Exit statement.

Q5-b

(10) (15)

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

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

```
main ( )
```

```
{
```

```
clrscr ( );
```

```
int a, b ;
```

```
char operation
```

```
cout << "Enter the first number, operator and
```

```
then second number = "endl;
```

```
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 '\';
```

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

```
default;
```

```
cout << "invalid input";
```

```
}
```

```
getch();
```

```
}
```