

# IUBAT

## International University of Business Agriculture and Technology

### **Lab Report - Final**

**Course Title** : Programming in C++.

**Course Code** : CSC-284

#### **Submitted To: -**

**ASM Shakil Ahmed,**

Lecturer,

Department of CSE.

#### **Submitted By:-**

**Name : Md. Alamin**

**ID : 21303134**

**Semester : Summer-2022**

**Program : BCSE**

**Section : C**

**Date of Submission: 21-08-2022**

1.

```
1 #include <iostream>
2 using namespace std;
3 class Contacts{
4     protected:
5         string name;
6         string personalNumber;
7         string officialNumber;
8
9     public:
10        Contacts() {
11            name = "Unknown";
12            personalNumber = "";
13            officialNumber = "";
14        }
15        void display() {
16            cout << name << endl;
17            cout << "Personal: " << personalNumber << endl;
18            if(!officialNumber.empty()){
19                cout << "Office: " << officialNumber << endl;
20            }
21            cout << endl;
22        }
23    }
24
25 };
26 class MyContacts : public Contacts{
27     public:
28        MyContacts(string name, string pn){
29            this->name = name;
30            personalNumber = pn;
31        }
32        MyContacts(string name, string pn, string on){
33            this->name = name;
34            personalNumber = pn;
35            officialNumber = on;
36        }
37    };
38
39 class FriendContacts : public Contacts{
40     public:
41        FriendContacts(string name, string pn){
42            this->name = name;
43            personalNumber = pn;
44        }
45        FriendContacts(string name, string pn, string on){
46            this->name = name;
47            personalNumber = pn;
48            officialNumber = on;
49        }
50    };
51 int main(){
52    FriendContacts fc("Alamin", "01822679672");
53    fc.display();
54    MyContacts mc("Shakib", "01912345678", "01712345678");
55    mc.display();
56
57
58 }
```

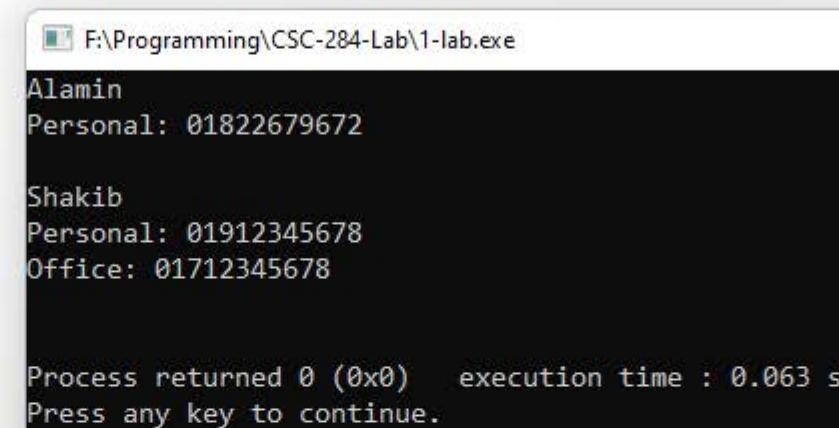
```
    name = "Unknown";
    personalNumber = "";
    officialNumber = "";
}
void display(){
    cout << name << endl;
    cout << "Personal: " << personalNumber << endl;
    if(!officialNumber.empty()){
        cout << "Office: " << officialNumber << endl;
    }
    cout << endl;
}

};

class MyContacts : public Contacts{
public:
    MyContacts(string name, string pn){
        this->name = name;
        personalNumber = pn;
    }
    MyContacts(string name, string pn, string on){
        this->name = name;
        personalNumber = pn;
        officialNumber = on;
    }
};

class FriendContacts : public Contacts{
public:
    FriendContacts(string name, string pn){
        this->name = name;
        personalNumber = pn;
    }
    FriendContacts(string name, string pn, string on){
        this->name = name;
        personalNumber = pn;
        officialNumber = on;
    }
};

int main(){
    FriendContacts fc("Alamin", "01822679672");
    fc.display();
    MyContacts mc("Shakib", "01912345678", "01712345678");
    mc.display();
}
```



```
F:\Programming\CSC-284-Lab\1-lab.exe
Alamin
Personal: 01822679672

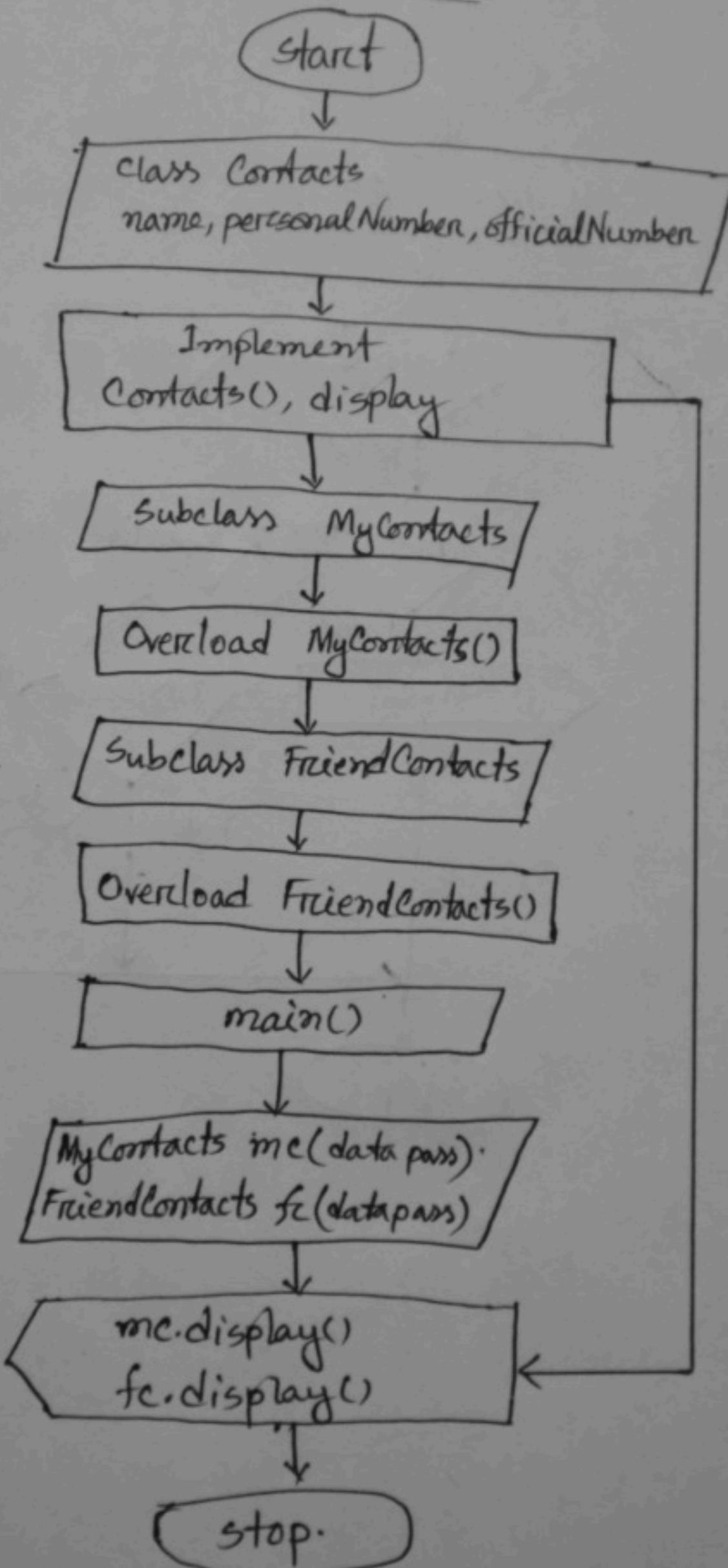
Shakib
Personal: 01912345678
Office: 01712345678

Process returned 0 (0x0)   execution time : 0.063 s
Press any key to continue.
```

## Algorithm for problem - 1:

- ① Start
- ② Create a class 'contacts' with the following members:  
name, personalNumber, officialNumber as member data.  
display as member function and contacts as an special member function.
- ③ Implement all the member function with their respective codes.
- ④ Create a sub-class 'MyContacts' of 'contacts'
5. Overload MyContacts as construction to set name, personalNumber and officialNumber.
6. Create another sub-class of 'FriendContacts' of 'contacts'.
7. Overload 'FriendContacts' as construction to set name, personalNumber, and official Number.
8. Create main function.
9. Create one object each (MyContacts and friendContacts) of these classes.
10. call the object member function.
11. Stop.

## Flowchart for Algorithm-1:



## 2.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     float an;
5     cout << "Enter a number: ";
6     cin >> an;
7
8     if(an==0) {
9         cout << "zero";
10        cout << " and small." << endl;
11    }else if(an<0){
12        cout << "negative";
13        cout << " and small." << endl;
14    }else if(an>0){
15        cout << "positive";
16        if(an>1000000){
17            cout << " and large." << endl;
18        }
19    }
20}
21
22 }
```

2-Lab.cpp X

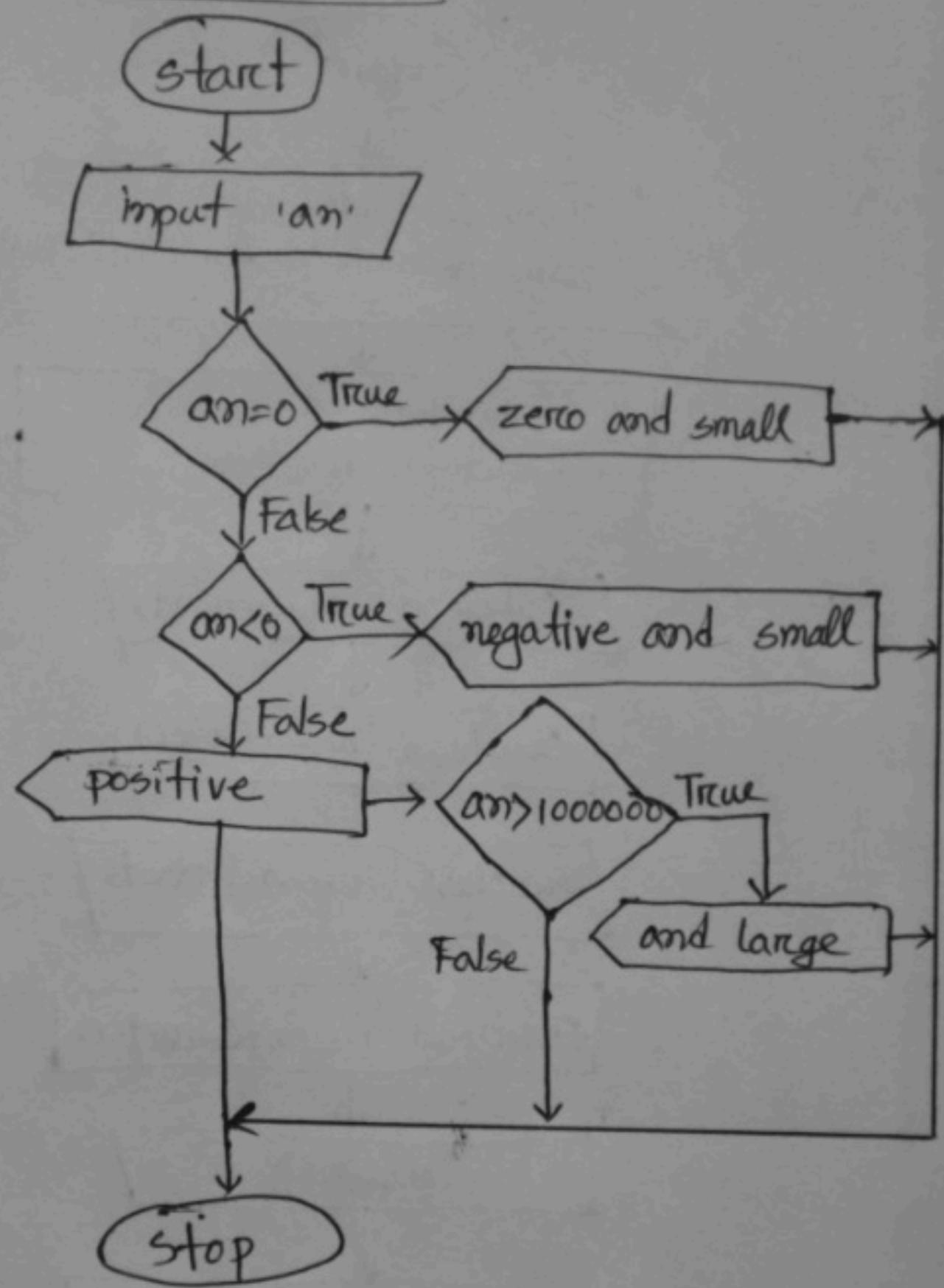
```
/**  
 * @file 2-Lab.cpp  
 * Write a C++ program that reads a floating-point number from the user.  
 * Prints "zero"; otherwise, print "positive" or "negative". If the absolute  
 * value of the number is less than one or "large." if the absolute value  
 * is greater than 10,000,000.  
 * @author Md. Alamin (alamin5g@yahoo.com)  
 * I would love be a software engineer at Google.  
 * @version 0.1  
 * @date 2022-08-03  
 *  
 */  
  
#include <iostream>  
using namespace std;  
  
int main(){  
    float an;  
    cout << "Enter a number: ";  
    cin >> an;  
  
    if(an==0){  
        cout << "zero";  
        cout << " and small." << endl;  
    }else if(an<0){  
        cout << "negative";  
        cout << " and small." << endl;  
    }else if(an>0){  
        cout << "positive";  
        if(an>1000000){  
            cout << " and large." << endl;  
        }  
    }  
}
```

F:\Programming\CSC-284-Lab\2-Lab.exe  
Enter a number: -5  
negative and small.  
Process returned 0 (0x0) execution time : 13.928 s  
Press any key to continue.

## Algorithm for lab-2

- ① Start.
- ② Declare a variable 'an' float type  
and take input.
- ③ if  $an = 0$   
    print 'zero and small'.
- ④ else if  $an < 0$   
    print 'negative and small'.  
else  
    print 'positive'  
    if ( $an > 1000000$ )  
        print 'and large'
4. Stop.

## Flowchart for Algorithm-2



### 3.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     string sentence;
5     getline(cin, sentence);
6     int vowel, consonant, len;
7     vowel = 0, consonant = 0;
8     len = sentence.size();
9     for(int i; i<len; i++){
10         if(sentence.at(i) == 'A' || sentence.at(i) == 'E' || sentence.at(i) == 'I' ||
11             sentence.at(i) == 'O' || sentence.at(i) == 'U'){
12             vowel++;
13         }else if(sentence.at(i) == 'a' || sentence.at(i) == 'e' || sentence.at(i) == 'i' ||
14             sentence.at(i) == 'o' || sentence.at(i) == 'u'){
15             vowel++;
16         }else if((sentence.at(i) >= 'A' && sentence.at(i) <= 'Z') || (sentence.at(i) >=
17             'a' && sentence.at(i) <= 'z')){
18             consonant++;
19         }
20     cout << "Vowels = " << vowel << endl;
21     cout << "Consonant = " << consonant << endl;
22
23     return 0;
24 }
```

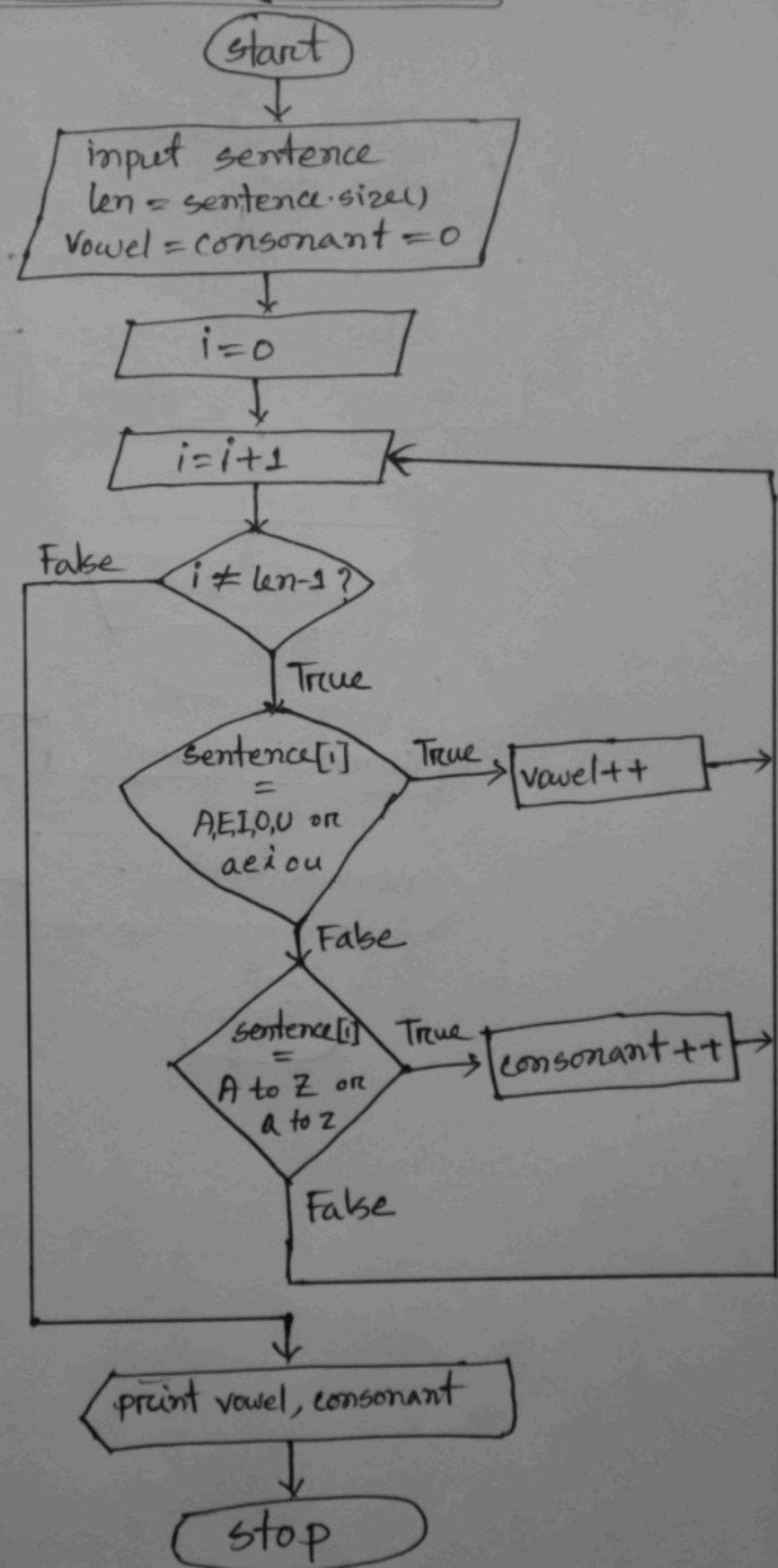
```
* Following is a statement:  
C++ is an OOP language  
In this sentence, there are  
Write a C++ program that will  
count the number of vowels and  
consonants.  
* @author Md. Alamin (alamin@louisville.edu)  
* I would love be a software engineer.  
* @version 0.1  
* @date 2022-08-02  
*  
* @copyright Copyright (c)  
*  
*/
```

```
F:\Programming\CSC-284-Lab\3-lab.exe  
C++ is an OOP Language  
Vowels = 8  
Consonant = 8  
Process returned 0 (0x0) execution time : 14.314 s  
Press any key to continue.
```

## Algorithm for problem -3

1. Start.
2. Declare, 1 string variable - sentence  
3 int variable len, vowel, consonant.  
take input 'sentence' variable.  
set len = ~~go~~ sentence.size();
3. set i=0;
4. set i=i+1;
5. if sentence.at(i) = A or E or I or O or U  
or a or e or i or o or u  
set vowel = vowel + 1.  
else if sentence.at(i) = A and <= Z or >= a  
and <= Z.  
set consonant = consonant + 1.
6. if i ≠ len-1 repeat step-4 and 5.
7. print vowel and consonants.
8. Stop.

## Flowchart for Algorithm-3:



# 4.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 class DhakaBank
4 {
5     string name;
6     int accountNo;
7     string month;
8     double deposit;
9     double balance;
10    double withdraw;
11    string email;
12
13
14 public:
15     DhakaBank()
16     {
17         name = "Md. Alamin";
18         accountNo = 43629;
19         balance = 500;
20         email = "alamin5g@yahoo.com";
21     }
22     void deposits()
23     {
24         cout << "Enter deposit Date: (ex: 31-July-2022): " << endl;
25         cin >> month;
26         cout << "Enter Amount: ";
27         cin >> deposit;
28         balance += deposit;
29     }
30     void withdrawn()
31     {
32         cout << "Enter withdraw Date: (ex: 31-July-2022): " << endl;
33         cin >> month;
34         cout << "Enter Amount: ";
35         cin >> withdraw;
36         if (withdraw > balance)
37         {
38             cout << "Insufficient Balance!" << endl;
39         }
40         else
41         {
42             cout << "You have withdrawn " << withdraw << " taka." << endl;
43             balance -= withdraw;
44         }
45     }
46
47
48     void sendInfo()
49     {
50         cout << endl;
51         cout << "~~~~~Information of your Account~~~~~\n" << endl;
52         cout << setw(10) << name << endl;
53
54         cout << "Account No- " << accountNo << endl;
55         cout << setw(10) << email << endl;
56         cout << setw(10) << "Remaining Balance: ";
57         cout << balance << endl;
58     }
59 };
60 int main()
61 {
62     DhakaBank alamin;
63     alamin.deposits();
64     alamin.withdrawn();
65     alamin.sendInfo();
66
67     return 0;
68 }
```

```
4-Lab.cpp x
}
void withdrawn()
{
    cout << "Enter deposit Date: (ex: 31-July-2022):";
    cin >> month;
    cout << "Enter withdraw Date: (ex: 31-July-2022):";
    cin >> withdraw;
    if (withdraw > balance)
    {
        cout << "Insufficient Balance!";
    }
    else
    {
        cout << "Your Account No- 43629";
        cout << "Email: alamin5g@yahoo.com";
        cout << "Remaining Balance: 2000";
    }
}

void sendInfo()
{
    cout << endl;
    cout << "_____";
    cout << setw(10) << "Account No: ";
    cout << setw(10) << "Name: ";
    cout << setw(10) << "Balance: ";
    cout << balance;
}

int main()
{
    DhakaBank alamin;
    alamin.deposits();
    alamin.withdrawn();
    alamin.sendInfo();

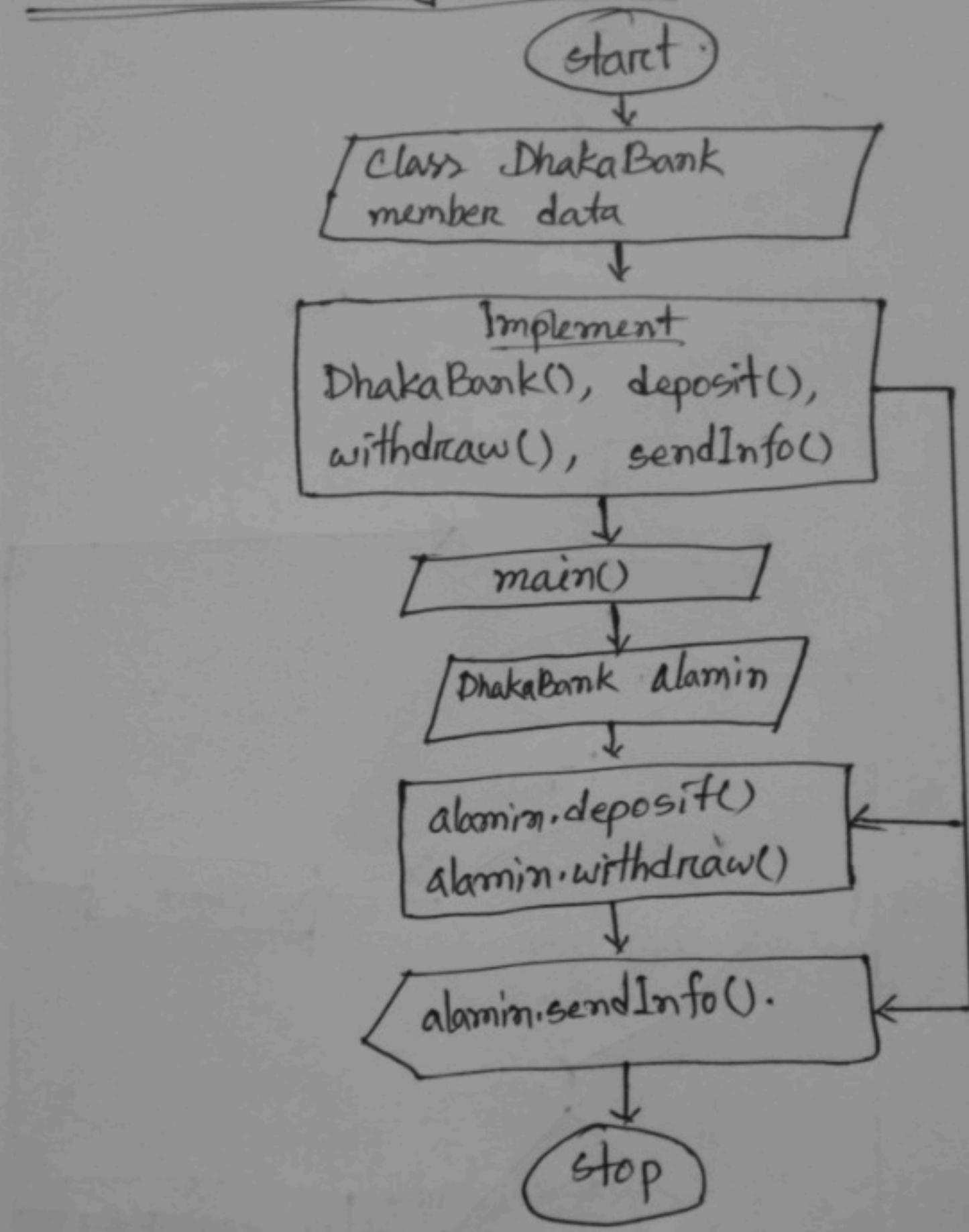
    return 0;
}
```

```
F:\Programming\CSC-284-Lab\4-Lab.exe
Enter deposit Date: (ex: 31-July-2022):
05-Aug-2022
Enter Amount: 1500
Enter withdraw Date: (ex: 31-July-2022):
09-Aug-2022
Enter Amount: 3000
Insufficient Balance!
____Information of your Account_____
Md. Alamin
Account No- 43629
Email: alamin5g@yahoo.com
Remaining Balance: 2000
Process returned 0 (0x0) execution time : 25.637 s
Press any key to continue.
```

## Program Algorithm for problem-4

- ① Start
- ② Create a class 'DhakaBank' with the following members:  
name, accountNo, month, deposit, balance, email, withdraw as member data.  
DhakaBank (constructor), deposit, withdraw, sendInfo as member function.
- ③ Implement all the member functions with their respective code.
- ④ Create main() function.
- ⑤ Create an object 'Alamin' of DhakaBank.
6. Call the deposit method and withdraw method through the object 'Alamin' of DhakaBank class.
7. Show the info by calling sendInfo().
8. Stop.

## Flowchart for Algorithm-4:



## 5.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     double engMarkJSC, engMarkSSC, engMarkHSC;
5     cout << "Enter JSC English Marks: ";
6     cin >> engMarkJSC;
7     cout << "Enter SSC English Marks: ";
8     cin >> engMarkSSC;
9     engMarkJSC = (engMarkJSC*25)/100;
10    engMarkSSC = (engMarkSSC*75)/100;
11    engMarkHSC = engMarkJSC + engMarkSSC;
12    cout << "Got " << engMarkHSC << " in HSC. ";
13    if(engMarkHSC>=80 && engMarkHSC<=100){
14        cout << "Grade A";
15    }else if(engMarkHSC>=70 && engMarkHSC<=79) {
16        cout << "Grade B";
17    }else if(engMarkHSC>=60 && engMarkHSC<=69) {
18        cout << "Grade C";
19    }else if(engMarkHSC>=50 && engMarkHSC<=59) {
20        cout << "Grade D";
21    }else if(engMarkHSC<50) {
22        cout << "Fail";
23    }
24    cout << endl;
25
26
27    return 0;
28 }
```

-Lab.cpp x

```
* @author Md. Alamin (alamin5g@yahoo.com)
* I would love be a software engineer
* @version 0.1
* @date 2022-08-02
*
* @copyright Copyright (c) 2022
*
*/
#include <iostream>
using namespace std;
int main(){
    double engMarkJSC, engMarkSSC, engMarkHSC;
    cout << "Enter JSC English Marks: ";
    cin >> engMarkJSC;
    cout << "Enter SSC English Marks: ";
    cin >> engMarkSSC;
    engMarkJSC = (engMarkJSC*25)/100;
    engMarkSSC = (engMarkSSC*75)/100;
    engMarkHSC = engMarkJSC + engMarkSSC;
    cout << "Got " << engMarkHSC << endl;
    if(engMarkHSC>=80 && engMarkHSC<=100) {
        cout << "Grade A";
    }else if(engMarkHSC>=70 && engMarkHSC<=79) {
        cout << "Grade B";
    }else if(engMarkHSC>=60 && engMarkHSC<=59) {
        cout << "Grade C";
    }else if(engMarkHSC>=50 && engMarkHSC<=49) {
        cout << "Grade D";
    }else if(engMarkHSC<50) {
        cout << "Fail";
    }
    cout << endl;

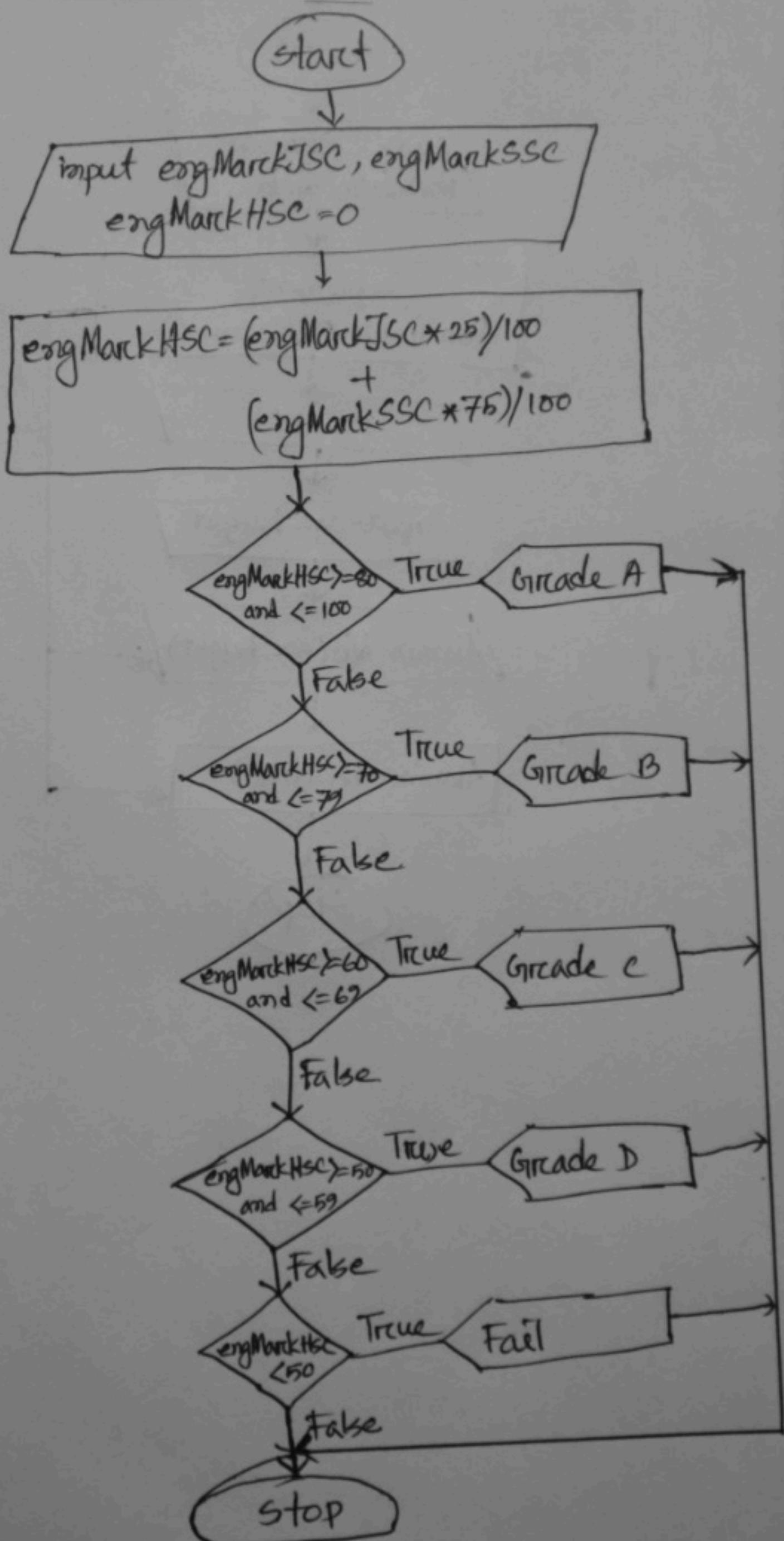
    return 0;
}
```

F:\Programming\CSC-284-Lab\5-Lab.exe  
Enter JSC English Marks: 100  
Enter SSC English Marks: 75  
Got 81.25 in HSC. Grade A  
Process returned 0 (0x0) execution time : 6.579 s  
Press any key to continue.

## Algorithm for problem-5:

- ① Start.
- ② Declare 3 double type variable: engMarkJSC, engMarkSSC, engMarkHSC. Input engMarkJSC and engMarkSSC.
- ③ get 25% of engMarkJSC and 75% from engMarkSSC.  
set engMarkHSC = 25% engMarkJSC ~~+ 75% engMarkSSC~~.
- ④ If  $\text{engMarkHSC} \geq 80$  and less than equal 100  
print 'Grade A'.  
else if  $\text{engMarkHSC} \geq 70$  and less than equal 79.  
print 'Grade B'.  
else if  $\text{engMarkHSC} \geq 60$  and less than equal 69  
print 'Grade C'.  
else if  $\text{engMarkHSC} \geq 50$  and less than equal 59  
print 'Grade D'.  
else if  $\text{engMarkHSC} < 50$   
print 'Fail'.
- ⑤ Stop.

## Flowchart for Algorithm-5:



# 6.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 class Alamin{
4     int age, height;
5     public:
6     Alamin(){
7         age = 0;
8         height = 0;
9     }
10    Alamin(int age, int height){
11        this->age = age;
12        this->height = height;
13    }
14    Alamin(Alamin &a){
15        cout << "Age = " << a.age << " and Height = " << a.height << endl;
16    }
17    ~Alamin(){
18        cout << "Memory cleaned!" << endl;
19    }
20 };
21 int main(){
22     int age, height;
23     cout << "Enter age and Height (cm): " << endl;
24     cin >> age >> height;
25     Alamin info(age, height);
26     Alamin show(info);
27
28     return 0;
29 }
```

```

* @version 0.1
* @date 2022-08-02
*
* @copyright Copyright (c) 2022
*
*/
#include <bits/stdc++.h>
using namespace std;

class Alamin{
    int age, height;
public:
    Alamin(){
        age = 0;
        height = 0;
    }
    Alamin(int age, int height){
        this->age = age;
        this->height = height;
    }
    Alamin(Alamin &a){
        cout << "Age = " << a.age <<
    }

    ~Alamin(){
        cout << "Memory cleaned!" << endl;
    }
};

int main(){
    int age, height;
    cout << "Enter age and Height (cm): " << endl;
    cin >> age >> height;
    Alamin info(age, height);
    Alamin show(info);

    return 0;
}

```

F:\Programming\CSC-284-Lab\6-Lab.exe

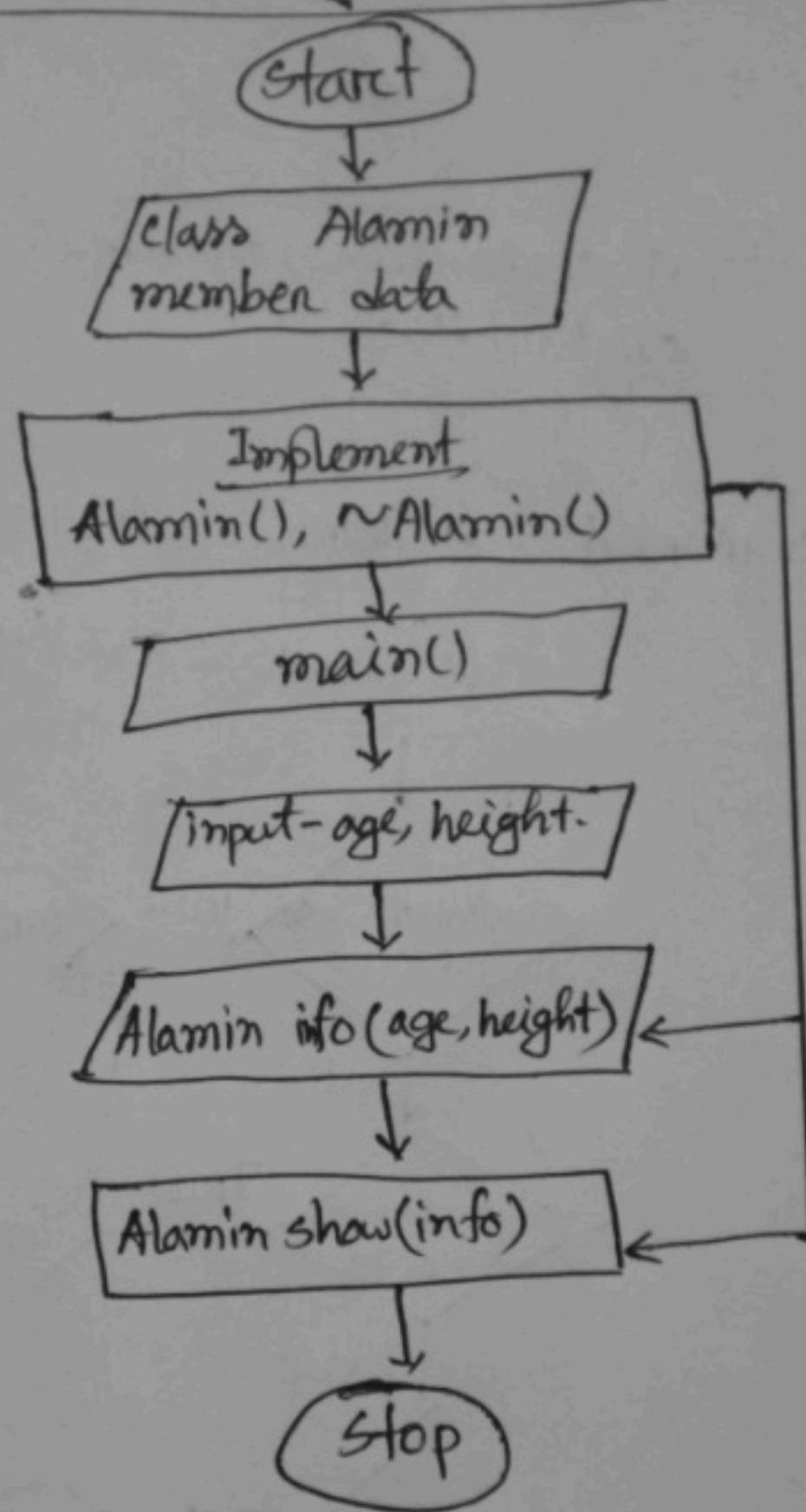
Enter age and Height (cm):  
25  
170  
Age = 25 and Height = 170  
Memory cleaned!  
Memory cleaned!

Process returned 0 (0x0) execution time : 56.094 s  
Press any key to continue.

## Algorithm for problem - 6 :

- ① Start
- ② Create a class 'Alamin' with the following members:
  - age, height as member data.
  - 3 constructor (special function)
  - 1 destructor.
- ③ Implement all the member function (constructor and destructor) with their respective code.
- ④ Create main function.
- ⑤ Declare 2 variable: age and height (int data)  
take input of age and height.
- ⑥ Create an object 'info' of Alamin class and pass age and height inside the parameter.
7. Create another object 'show' of Alamin and pass 'info' object inside the parameter of 'show' object.
8. Stop.

## Flowchart for Algorithm-6:



7.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int arr[] = {2, 7, 4, -5, 11, 5, 20};
5     int specifiedNumber=15;
6     for(int i=0; i<7; i++){
7         for(int j=i+1; j<7; j++){
8             if(arr[i]+arr[j]==specifiedNumber) {
9                 cout << arr[i] << " + " << arr[j] << " = " << arr[i]+ arr[j] << endl;
10            }
11        }
12    }
13 }
14 return 0;
15 }
```

7-Lab.cpp x

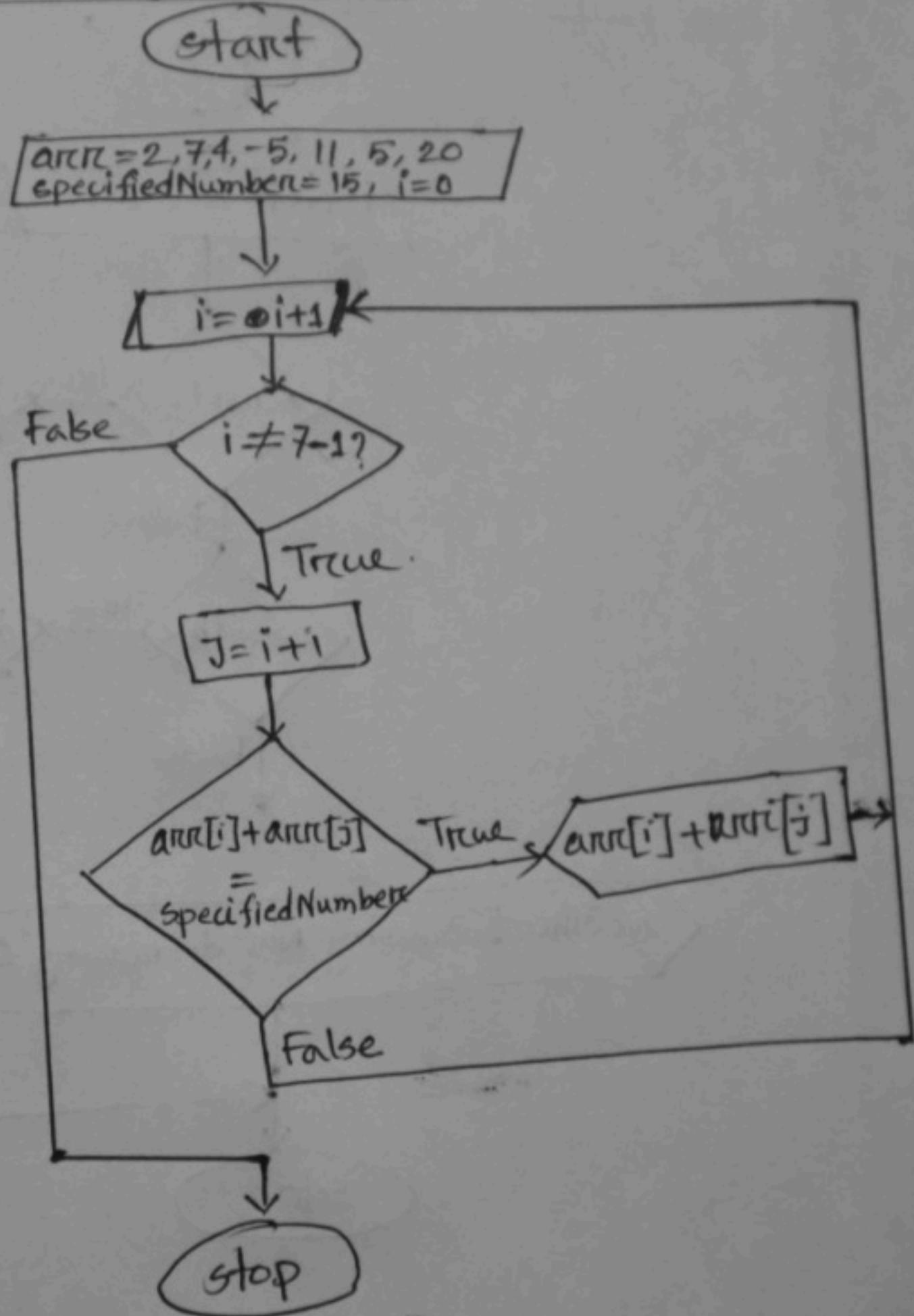
```
1  /**
2   * @file 7-Lab.cpp
3   * Write a C++ program to
4   * in an array whose sum
5   * @author Md. Alamin (a
6   * I would love be a sof
7   * @version 0.1
8   * @date 2022-08-03
9
10  */
11
12 #include <iostream>
13 using namespace std;
14 int main() {
15     int arr[] = {2, 7, 4, 11, 5, 20, 15};
16     int specifiedNumber=15;
17     for(int i=0; i<7; i++) {
18         for(int j=i+1; j<7; j++) {
19             if(arr[i]+arr[j]==specifiedNumber)
20                 cout << arr[i] << " + "
21                               << arr[j] << " = "
22                               << specifiedNumber << endl;
23     }
24 }
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 }
```

```
F:\Programming\CSC-284-Lab\7-Lab.exe
4 + 11 = 15
-5 + 20 = 15
Process returned 0 (0x0) execution time : 0.063 s
Press any key to continue.
```

## Algorithm for problem-7:

1. Start
2. Declare one array of int, 'arr' and set the value  $arr = 2, 7, 4, -5, 11, 5, 20$ .  
declare another (int) variable - 'specified Number' and set the value - 15.
3. set  $i=0$ :
4.  ~~$i=i+1$~~
5. set  $j=i+1$
6. if  $arr[i] + arr[j] = \text{specified Number}$   
print ' $arr[i] + arr[j]$ '.
7.  $i=i+1$ .
7.  $i \neq 7-1$  step 5.
8. Stop.

## Flowchart for Algorithm: 7



# 8.

```
1 #include <iostream>
2 using namespace std;
3 int main(){
4     int a, b, c;
5     cout << "Enter 3 separate number: ";
6     cin >> a >> b >> c;
7     if(a < b && b < c){
8         cout << "Increasing Order!" << endl;
9     }else if(a > b && b > c){
10        cout << "Decreasing Order!" << endl;
11    }else {
12        cout << "Neither increasing nor decreasing order" << endl;
13    }
14
15
16    return 0;
17 }
```

8-Lab.cpp X

```
F:\Programming\CSC-284-Lab\8-Lab.cpp
 * @file 8-Lab.cpp
 * Write a program that accept
 * if the numbers are in increasing
 * and /"Neither increasing
 * @author Md. Alamin (alami
 * I would love be a software engineer
 * @version 0.1
 * @date 2022-08-03
 *
 * @copyright Copyright (c)
 *
 */
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    cout << "Enter 3 separate number: ";
    cin >> a >> b >> c;
    if(a < b && b < c){
        cout << "Increasing order
    }else if(a > b && b > c)
        cout << "Decreasing order
    }else {
        cout << "Neither increasing nor decreasing order
    }
}
return 0;
```

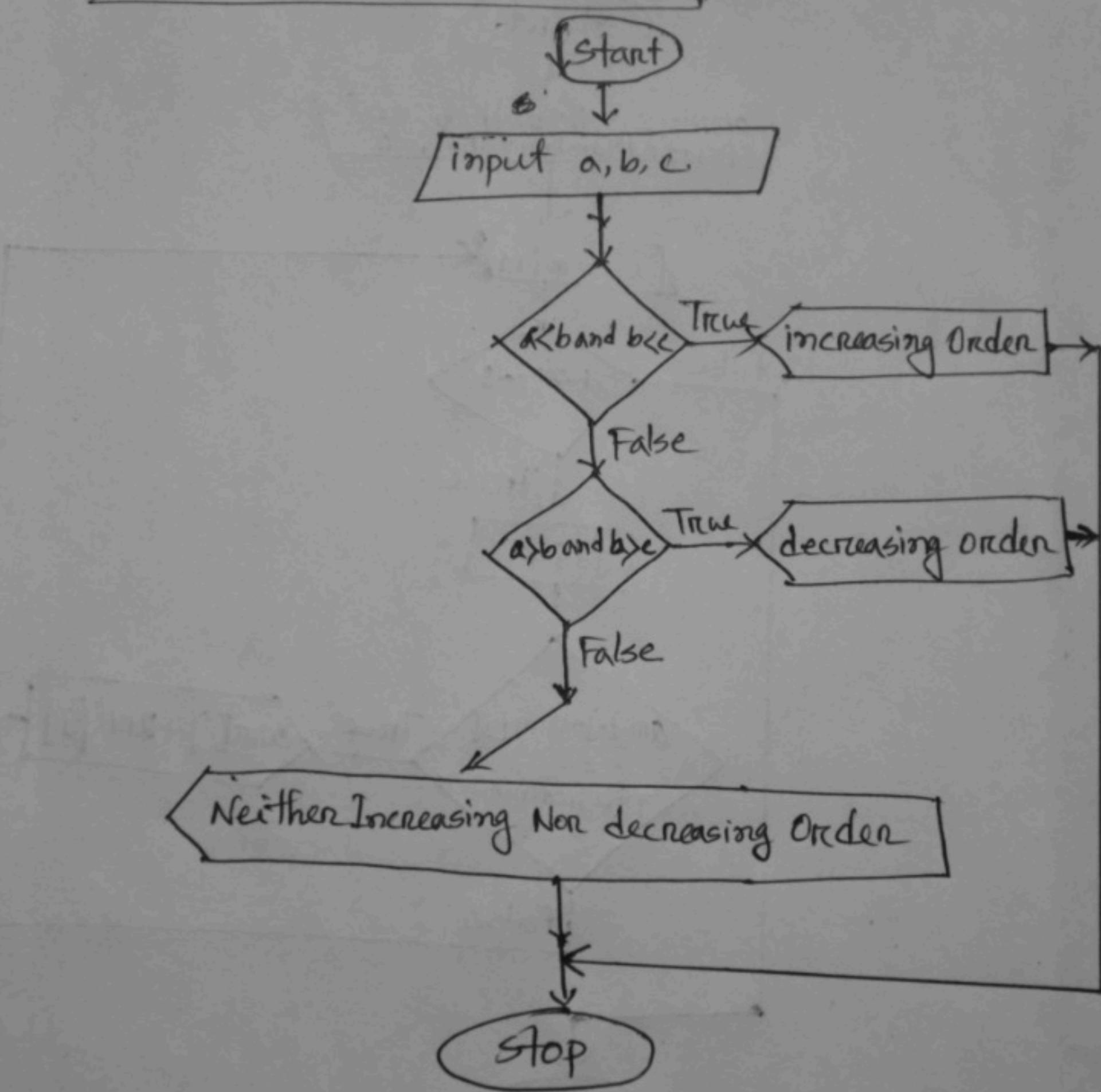
E:\Programming\CSC-284-Lab\8-Lab.exe

```
Enter 3 separate number: 50
70
60
Neither increasing nor decreasing order
Process returned 0 (0x0) execution time : 16.224 s
Press any key to continue.
```

## Algorithm for problem - 8:

1. Start
2. Declare 3 int variable a, b, c and take input.
3. if  $a < b$  and  $b < c$   
    print 'Increasing Order'  
else if  $a > b$  and  $b > c$   
    print 'Decreasing Order'.  
else  
    print 'Neither Increasing nor decreasing order'.
4. Stop.

## Flowchart for Algorithm-8:



# 9.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 class ABCIceCream{
4     public:
5         void getIceCream(){
6             cout << "All Ice Cream Flavored available." << endl;
7         }
8     };
9 };
10
11 class BranchInHouse : public ABCIceCream{
12     public:
13         void getIceCream(){
14             cout << "Vanilla Flavour Ice Cream" << endl;
15         }
16     };
17 class BranchInTown : public ABCIceCream{
18     public:
19         void getIceCream(){
20             cout << "Chocolate Flavored Ice Cream" << endl;
21         }
22     };
23 int main(){
24     BranchInHouse branch1;
25     branch1.getIceCream();
26
27     BranchInTown branch2;
28     branch2.getIceCream();
29
30     ABCIceCream mainBranch;
31     mainBranch.getIceCream();
32
33
34     return 0;
35 }
```

```
* @version 0.1
* @date 2022-08-04
*
* @copyright Copyright (c)
*
*/
#include <bits/stdc++.h>
using namespace std;
class ABCIceCream{
public:
    void getIceCream() {
        cout << "All Ice Cr
    }
};

class BranchInHouse : public
public:
    void getIceCream() {
        cout << "Vanilla Fl
    }
};

class BranchInTown : public
public:
    void getIceCream() {
        cout << "Chocolate
    }
};

int main() {
    BranchInHouse branch1;
    branch1.getIceCream();

    BranchInTown branch2;
    branch2.getIceCream();

    ABCIceCream mainBranch;
    mainBranch.getIceCream();

    return 0;
}
```

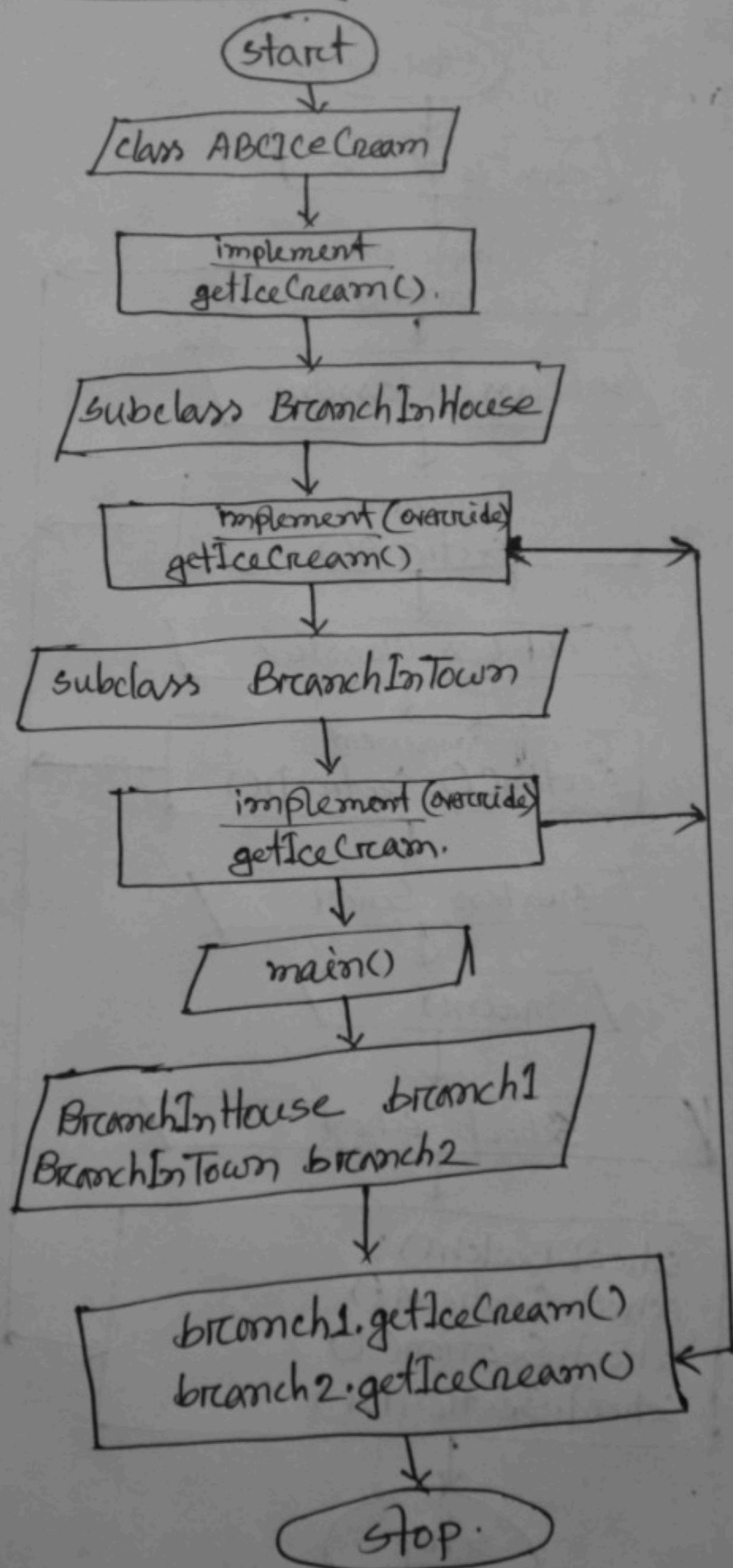
F:\Programming\CSC-284-Lab\9-Lab.exe  
Vanilla Flavour Ice Cream  
Chocolate Flavored Ice Cream  
All Ice Cream Flavored available.

Process returned 0 (0x0) execution time : 0.063 s  
Press any key to continue.

## Algorithm for problem-9:

- ① Start.
- ② Create a class 'ABCIceCream' with one member function - getIceCream.
- ③ Implement getIceCream function with its respective code.
- ④ Create a subclass 'BranchInHouse' of ABCIceCream.
- ⑤ Override getIceCream function of BranchInHouse with respective code.
- ⑥ Create another subclass 'BranchInTown' of ABCIceCream.
- ⑦ Override getIceCream function for BranchInTown with respective code.
- ⑧ Create a main function.
- ⑨ Create one object each of those two (BranchInHouse, BranchInTown) subclasses - branch-1 and branch-2.
- ⑩ Call getIceCream() function by branch-1 and branch2 object.
- ⑪ Stop.

## Flowchart for Algorithm-9:



# 10.

```
1 #include <iostream>
2 using namespace std;
3 class TestClass
4 {
5 public:
6     void Batch()
7     {
8         cout << "This is batch of TestClass" << endl;
9     }
10 };
11 class ClassOne : public virtual TestClass
12 {
13 public:
14     void SectionA()
15     {
16         cout << "This is SectionA of ClassOne" << endl;
17     }
18 };
19 class ClassTwo : public virtual TestClass
20 {
21 public:
22     void SectionC()
23     {
24         cout << "This is SectionC of ClassTwo" << endl;
25     }
26     void SectionD()
27     {
28         cout << "This is SectionD of ClassTwo" << endl;
29     }
30 };
31 class School : public ClassOne, public ClassTwo{
32 };
33 };
34 int main()
35 {
36     School school;
37     school.Batch(); // accessing TestClass
38     school.SectionA(); // accessing ClassOne
39     school.SectionC(); //accessing classTwo
40     school.SectionD(); //accessing classTwo
41
42     return 0;
43 }
44 }
```

```

{
public:
    void Batch()
    {
        cout << "This is batch of TestClass"
    }
};

class ClassOne : public virtual Test
{
public:
    void SectionA()
    {
        cout << "This is SectionA of ClassOne"
    }
};

class ClassTwo : public virtual Test
{
public:
    void SectionC()
    {
        cout << "This is SectionC of ClassTwo"
    }

    void SectionD()
    {
        cout << "This is SectionD of ClassTwo"
    }
};

class School : public ClassOne, public ClassTwo
{
};

int main()
{
    School school;
    school.Batch(); // accessing TestClass
    school.SectionA(); // accessing ClassOne
    school.SectionC(); //accessing classTwo
    school.SectionD(); //accessing classTwo

    return 0;
}

```

F:\Programming\CSC-284-Lab\10-Lab.exe

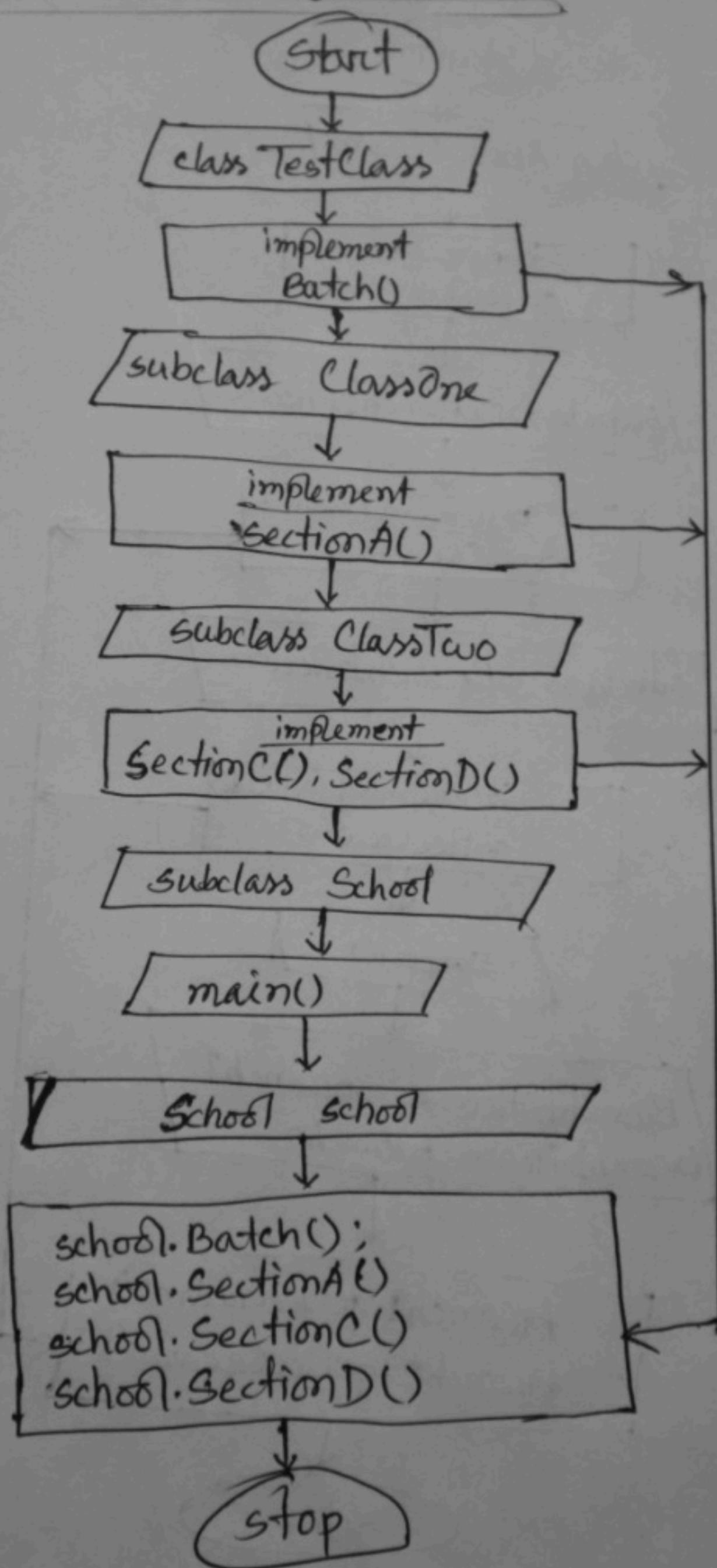
This is batch of TestClass  
 This is SectionA of ClassOne  
 This is SectionC of ClassTwo  
 This is SectionD of ClassTwo

Process returned 0 (0x0) execution time : 0.047 s  
 Press any key to continue.

## Algorithm for problem-10.

1. Start.
2. Create a class 'TestClass' with one member function - Batch.
3. Implement Batch function with its respective code.
4. Create a subclass 'ClassOne' of TestClass with one member function - section A.
5. Implement sectionA function with its respective code.
6. Create another subclass 'ClassTwo' of TestClass with 2 member function: SectionC and SectionD.
7. Implement SectionC and SectionD function with their respective code.
8. Create a & subclass of 'School' of ClassOne and ClassTwo.
9. Create main function.
10. Create an object 'school' of School' class
11. Call all the member function of the program by school object.
12. Stop.

## Flowchart for Algorithm-10



# 11.

```
1 #include <bits/stdc++.h>
2 using namespace std;
3 class Vehicles
4 {
5     protected:
6         int mileage, price;
7 };
8
9 class Car : protected Vehicles
10 {
11     protected:
12         int cost, warrenty, seat;
13         string fuelType;
14 };
15
16 class Bike : protected Vehicles
17 {
18     protected:
19         int cylinderNum, gearsNum, fuleTankSize;
20         string coolingType, wheelType;
21 };
22
23 class Audi : private Car
24 {
25     string model;
26 public:
27     void input(){
28         cout << "Enter Model Type: ";
29         getline(cin, model);
30         cout << "Enter Ownership Cost: ";
31         cin >> cost;
32         cout << "Enter warranty: ";
33         cin >> warrenty;
34         cout << "Enter Seating Capacity: ";
35         cin >> seat;
36         cout << "Enter Fuel Type: ";
37         cin >> fuelType;
38         cout << "Enter milage: ";
39         cin >> mileage;
40         cout << "Enter price: ";
41         cin >> price;
42     }
43     void showInfo(){
44         cout << "Model Type: " << model << endl;
45         cout << "Ownership Cost: " << cost << endl;
46         cout << "warranty: " << warrenty << endl;
47         cout << "Seating Capacity: " << seat << endl;
48         cout << "Fuel Type: " << fuelType << endl;
49         cout << "milage: " << mileage << endl;
50         cout << "price: " << price << endl;
51     }
52 };
53
54 class Ford : private Car
55 {
56     string model;
57 public:
58     void input(){
59         cout << "Enter Model Type: ";
60         getline(cin, model);
61         cout << "Enter Ownership Cost: ";
62         cin >> cost;
63         cout << "Enter warranty: ";
64         cin >> warrenty;
65         cout << "Enter Seating Capacity: ";
66         cin >> seat;
67         cout << "Enter Fuel Type: ";
68         cin >> fuelType;
69         cout << "Enter milage: ";
70         cin >> mileage;
71         cout << "Enter price: ";
72         cin >> price;
73     }
74     void showInfo(){
75         cout << "Model Type: " << model << endl;
76         cout << "Ownership Cost: " << cost << endl;
77         cout << "warranty: " << warrenty << endl;
```

```

78     cout << "Seating Capacity: " << seat << endl;
79     cout << "Fuel Type: " << fuelType << endl;
80     cout << "mileage: " << mileage << endl;
81     cout << "price: " << price << endl;
82   }
83 };
84
85 class Bajaj : private Bike
86 {
87   string makeType;
88 public:
89   void input(){
90     cout << "Enter make Type: ";
91     getline(cin, makeType);
92     cout << "Enter the Number of Cylinder: ";
93     cin >> cylinderNum;
94     cout << "Enter Gears Number: ";
95     cin >> gearsNum;
96     cout << "Enter fuel Tank Size Capacity: ";
97     cin >> fuleTankSize;
98     cout << "Enter Cooling Type: ";
99     cin >> coolingType;
100    cout << "Enter Wheel Type: ";
101    cin >> wheelType;
102    cout << "Enter milage: ";
103    cin >> mileage;
104    cout << "Enter price: ";
105    cin >> price;
106  }
107  void showInfo(){
108    cout << "make Type: " << makeType << endl;
109    cout << "the Number of Cylinder: " << cylinderNum << endl;
110    cout << "Gears Number: " << gearsNum << endl;
111    cout << "fuel Tank Size Capacity: " << fuleTankSize << endl;
112    cout << "Cooling Type: " << coolingType << endl;
113    cout << "Wheel Type: " << wheelType << endl;
114    cout << "milage: " << mileage << endl;
115    cout << "price: " << price << endl;
116  }
117};
118
119 class TVS : private Bike
120 {
121   string makeType;
122 public:
123   void input(){
124     cout << "Enter make Type: ";
125     getline(cin, makeType);
126     cout << "Enter the Number of Cylinder: ";
127     cin >> cylinderNum;
128     cout << "Enter Gears Number: ";
129     cin >> gearsNum;
130     cout << "Enter fuel Tank Size Capacity: ";
131     cin >> fuleTankSize;
132     cout << "Enter Cooling Type: ";
133     cin >> coolingType;
134     cout << "Enter Wheel Type: ";
135     cin >> wheelType;
136     cout << "Enter milage: ";
137     cin >> mileage;
138     cout << "Enter price: ";
139     cin >> price;
140  }
141  void showInfo(){
142    cout << "make Type: " << makeType << endl;
143    cout << "the Number of Cylinder: " << cylinderNum << endl;
144    cout << "Gears Number: " << gearsNum << endl;
145    cout << "fuel Tank Size Capacity: " << fuleTankSize << endl;
146    cout << "Cooling Type: " << coolingType << endl;
147    cout << "Wheel Type: " << wheelType << endl;
148    cout << "milage: " << mileage << endl;
149    cout << "price: " << price << endl;
150  }
151};
152
153 int main(){
154   Audi audi;

```

```
155     audi.input();
156     audi.showInfo();
157
158     Ford ford;
159     ford.input();
160     ford.showInfo();
161
162     Bajaj bajaj;
163     bajaj.input();
164     bajaj.showInfo();
165
166     TVS tvs;
167     tvs.input();
168     tvs.showInfo();
169
170 }
171
```

```

        cin >> fuelTankSize;
        cout << "Enter Cooling Type: ";
        cin >> coolingType;
        cout << "Enter Wheel Type: ";
        cin >> wheelType;
        cout << "Enter milage: ";
        cin >> mileage;
        cout << "Enter price: ";
        cin >> price;
        cout << endl;
    }

    void showInfo() {
        cout << "make Type: " << makeType;
        cout << "the Number of Gears: ";
        cout << "Gears Number: ";
        cout << "fuel Tank Size: ";
        cout << "Cooling Type: ";
        cout << "Wheel Type: ";
        cout << "milage: ";
        cout << "price: ";
    }
}

int main() {
    Audi audi;
    audi.input();
    audi.showInfo();

    Ford ford;
    ford.input();
    ford.showInfo();

    Bajaj bajaj;
    bajaj.input();
    bajaj.showInfo();

    TVS tvs;
    tvs.input();
    tvs.showInfo();
}

```

E:\Programming\CSC-284-Lab\11-Lab.exe

```

Enter Model Type: Custom
Enter Ownership Cost: 40000
Enter warranty: 1
Enter Seating Capacity: 4
Enter Fuel Type: oil
Enter milage: 10
Enter price: 400000

Model Type: Custom
Ownership Cost: 40000
warranty: 1
Seating Capacity: 4
Fuel Type: oil
mileage: 10
price: 400000

Enter Model Type: Audi Custom
Enter Ownership Cost: 5000
Enter warranty: 2
Enter Seating Capacity: 6
Enter Fuel Type: oil
Enter milage: 12
Enter price: 50000

Model Type: Audi Custom
Ownership Cost: 5000
warranty: 2
Seating Capacity: 6
Fuel Type: oil
mileage: 12
price: 50000

```

## Algorithm for problem -11.

- ① Start .
- ② Create a class 'Vehicle' with the following 2 data members- mileage, price.
- ③ Create a subclass 'Car' of Vehicle with the following data member: cost, warranty, seat, and fueltype.
- ④ Create another subclass 'Bike' of Vehicle with the following data members:- cylinderNum, gearsNum, fuelTankSize, coolingType, and wheelType.
- ⑤ Create two subclasses 'Audi', 'Ford' of Car class with the following members:-  
model - as member data.  
'input', 'info' as member function.
6. Implement all member function with its respective code.
7. Create two subclasses 'Bajaj', 'TVS' of Bike class with the following members.  
makeType - as member data.  
'input', 'showInfo' as member function.
8. Implement all member function with its respective code.

9. Create main function.
10. Create object of Audi, Ford, Bajaj and Tvs.
11. Call 'input' and 'showInfo' function ~~of~~ for each of the class by its object.
12. Stop.

## Flowchart for Algorithm m-11

