2006167 - Ankit Srivastava

Q1 Class: (i) If access specifier not declared, by default all members are private.

(ii) Instance of class is called 'object'.

(iii) Memory allocated on Heap.

structure: (i) If access specifier not declared, by default all members are public.

(ii) Instance of structure is called structure variable.

(iii) Memory is allocated on stack.

Q3 The scope resolution operator helps to identify and specify to which context an identifier refers.
It can be used in:
(i) To access global variable when there is local variable with same name.
(ii) To define member function outside class.
(iii) Access/explicitly define class's static variable.
(iv) ~~foo~~ for accessing same named variable in different namespace.
(v) Refer to a class inside another class.

Q5 friend function: If a func^n is defined as friend func^n, it can access even the protected and private data on the class.

friend Class: All member func^n of friend class will be friend func^n to another class.

```
class z {
    :
    friend class x; // all M.F of x are friend to z.
};
```

2006167 - Ankit Srivastava

Q7 In an ideal class (for object oriented system), the data and funcⁿ are organized in a manner to prevent misuse. All Data are preffered to be kept private/protected and can only be used by member funcⁿ to avoid misuse.

Q9 Syntax ~~Error~~ : (i) occurs when we violate the rules of writing the statements of the programming language.

(ii) Program fails to compile and execute.

(iii) Syntax ~~Error~~ caught by compiler

Logical Error : (i) Logical ~~error~~ occurs due to our mistake in programming logic.

(ii) Program compiles and exeutes but dosen't give desired output.

(iii) needs to be found and corrected by people working on the program.

Q11 NULL

Q13 Header files contain the set of predefined standard library functions eg: cout, cin. The are important to aviod rewriting of code which are already written and publically accessible.

Yes, we can write simple program without header files but have to define our own funcⁿ for I/O operations. It's better and efficient to use header file though.

Q15 funcⁿ declaration : Is a prototype that specifies the function name, return type and parameters without body.

function definition : Is the actual funcⁿ containing funcⁿ name, return-type, parameters along with funcⁿ Body (which specifies the work of funcⁿ).

Q17 A default argument is given to the formal parameters of funcⁿ and are automatically assigned by the compiler if the funcⁿ call dosent pass argumentit for the funcⁿ..

```
void add (int a, int b=0){
        cout<< a+b;
}
int main(){
    add (10, 5);      // output: 15
    add (2);          // output : 2     {default b=0}
    return 0;
}
```

Q19 We can pass 2D array to a funcⁿ by pointer to pointer:

Algorithm:

```
void processArray (int ~~**~~ arr){
        // some code.
}
int main(){
    int **array;
    array = new int*[10];
    for (int i=0 ; i<10 ; i++)
            array[i] = new int[10];
    process Array (array);
}
```

2006167-Ankit Srivastava

2006167- Ankit Srivastava

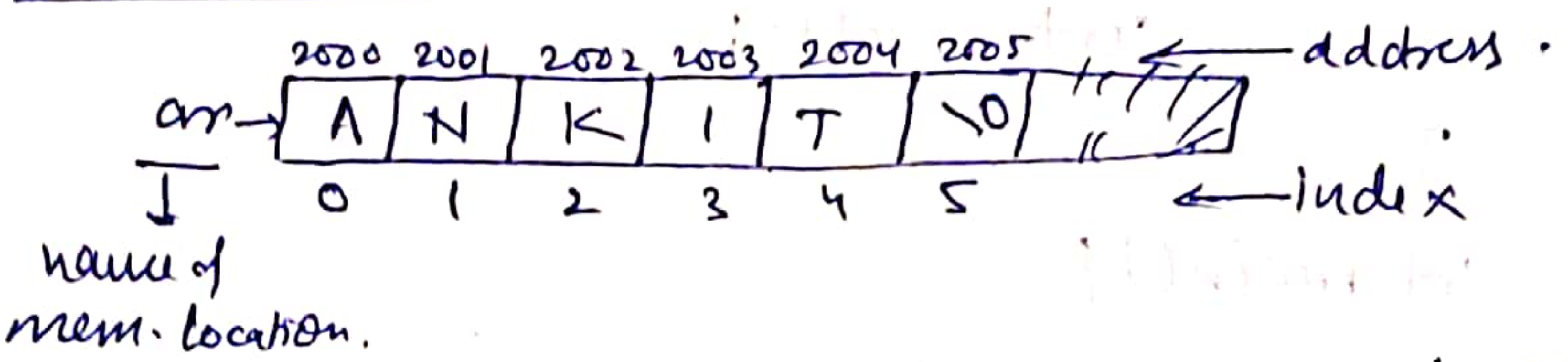
Q21 NULL

Q23 NULL

Q25

char ar[] = {"A","N","K","I","T","\0"};

↳ Representation :



string of array is stored at continous memory location (∵ array).

Q27 strings are read until we encounter a 'newline' character or a null value.

The different functions to read strings are !

① cin>>[string_variable]; // inputs until new line encountered.

② gets(<string_variable>); // reads until " ...
// dangerous, causes buffer creation.

③ fgets(<string_variable>); // safer than gets().

④ scanf("%[^\n]s", str); // reads/store new line as well, to ~~avio~~ avoid logical error in further input code.

⑤ cin.getline (str, <array size>); // reads until array size.

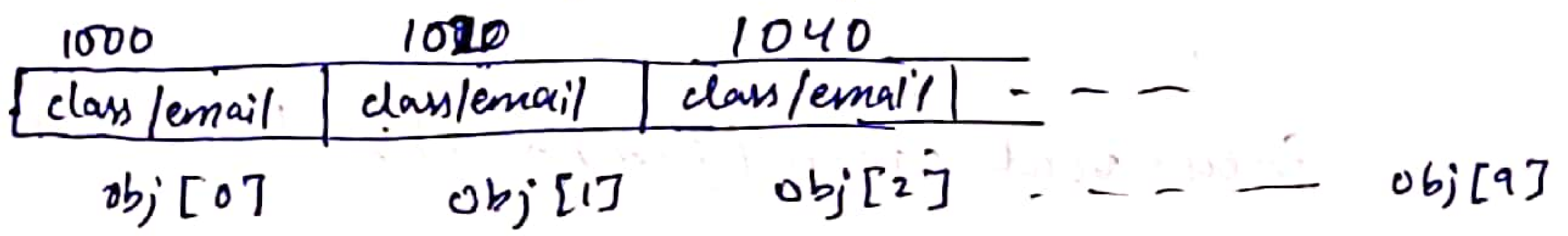
Q29 Yes, we can create array of unions and access them in similar way as of structure.

When array of union is created, each element of array is made an individual union.

i.e each element will be allocated memory equivalent to that of maximum size of the union member.

```
union student {
    int class;          // 2B
    char email[20];     // 20B

};

union student obj[10]; // creates array of unions with 10 elements
                          of union type.
```

each add → of 20B.



Q31 ~~Yes, we can use the private member of class using friend funcⁿ without the help of class object.~~

~~class data{~~
~~int a;~~
~~friend void disp~~
~~};~~

```
class data{
    int a;
Public:
    friend void disp(int x);
};

void disp(int x){
    a = 10;
    cout << a;

}

int main(){
    data s;
    disp(10); // call frnd funcⁿ, assign 10
                 to a
    return 0;
}
```

2006167 - Ankit Srivastava

Q31 Yes, we can access private data member of class without using object by friend func^n.

```
class sample {
        int length;
        Public:
           sample( int l ~~length~~ = 10) { ~~F~~ ;  //constructor
                    length = l;
           }
        friend void square (sample s);  // friend func^n declaration

   };
   void square (sample s) {
           Cout << s.length * s.length ;
   }

   int main() {
        sample obj ;
        square (obj);  // calls friend func and
                          output : 100
        return 0;
   }
```

Q33 Every object in C++ has access to its own address through an important pointer called "this" pointer. This may-be used to refer to the invoking/calling ~~fo~~ object.

```
class demo {
     int x;
    Public:
     void set( int x) {
           ~~thist~~ this -> x = x;    // this -> x  use to retrieve
                                          object's (data member).
     }                                 // 'x' is the local variable x
 int main() {
     demo obj;
     obj. set(10);
 }
```

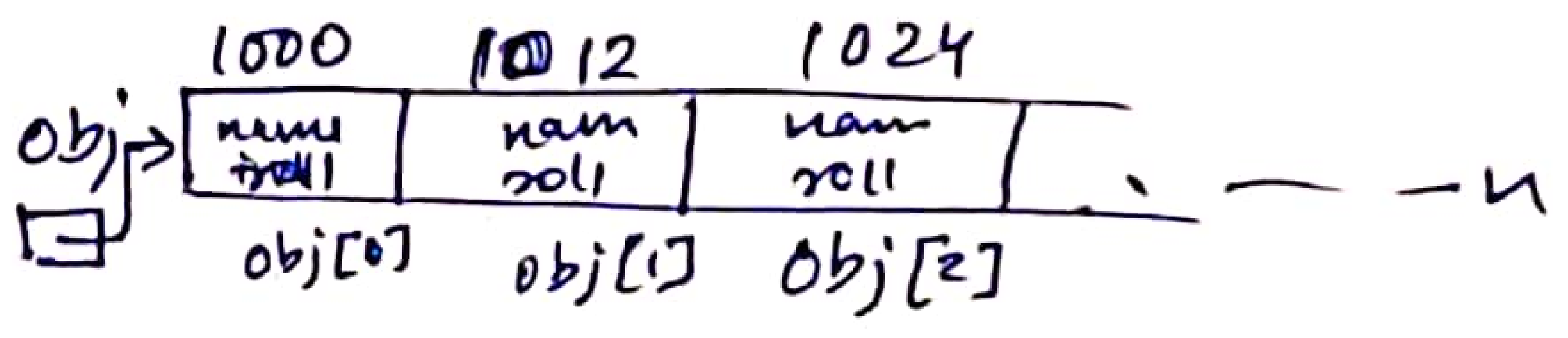
2006167 – Ankit Srivastava

Q35 NULL.

Q37: when we are unaware of the number of objects needed, we declare array of object, and in order to avoid allocating garbage value in 'n' we use Dynamic Memory Allocation.

Dynamicall Allocating array of objects creates an array of size given by user and creates that much memory blocks each of the size of class. We can now access and manipulate multiple objects under single array name with different index no..

```
class student {
        char name[10];  // 10B
        int roll;          // 2B
      Public:
          :

};

int main() {
        int n;
        cin>>n;
        student *obj = new student[n];
        for(int i=0 ; i<n; i++)
              // access obj[i];
        free(obj);
        return 0;
}
```



Q39 A "static" object is unique, it belongs to the class rather than the instance of the class.
Static variable is only allocated memory once: when the class loads.

```
class outputCounter{
        static int count;
      Public:
          void printCall(){
                cout<<"called";
                count++;
          }
          void printCount(){
                cout<<count;
          }
};

int outputCounter::count = 0;

int main(){
      outputCounter obj1, obj2;
        obj1.printCall();   // count=1
        obj2.printCall();   // count=2

      outputCounter obj3;
        obj3.printCount();  // 2 ✓
      return 0;
}
```

Output

```
called called 2
```

3 object initialized but static only once (~~com~~ class).

PTO: ① Odd short Question then
② Even short Question

2006167 - Ankit Srivastava

Odd 2006167 – Ankit Srivastava

Q1 Output : 1
Class with no data member will take atleast 1Byte memory..

Q3.
```
x=10;
int main() {
    int x=20;
    cout<<::x; //global
    cout<<x; //local
    {  int x=30;
       cout<<x; //local
    }
    return 0;
}
```
Output : 102030

Q5 Manipulators are operators used in c++ for formatting output.
The data is manipulated by users choice of display.
We do not change the value of a variable, it only modifies I/O stream. ~~using~~
```
#includ <iomanip>
cout<< setw(10) << setprecision(2)<< f <<endl;
```
setw(10): right justified in width of 10 col.
setprecision(2): precision of 2digit
f: variable (float)

Q7
```
 :
 :
static void display() {   //static Mem. func can access only static Data member.
①     cout<<x<<endl;
  }
 }
};
② int test::x;   // declaration for class static variable
int main(){
   :
③  T1.output();
 ;
```

2006187- Ankit Srivastava

Q9
```
int fun();
int fun(int x=0, int y=0);
```
In above case, problem of "ambiguity" will arrise!
If we call funcⁿ with giving any arguments, compiler may get confused between which of the above funcⁿ to call.

A appropriate alternative could be:
```
int fun();
int fun(int x, int y=0);
```
else we can change funcⁿ name too.

The advantage of Inline funcⁿ:
(c). it takes less execution time.

Q11 Yes, compiler by default create default constructor for every class. But if we define our own constructor, compiler doesn't create the default constructor.

```
class demo {
      int value
      // some code
  ;
};
int main() {
      ~~myInteger I1;~~
      demo d;
      ;
}
```
works fine!

```
class demo {
          int value
      Public:
      ~~void~~ demo(int x) {
                //code
           }
      ;
  };
int main() {
      demo d(10);  //correct
      demo d2;  // fail to call
                   demo() constructor.
      ;
                hence, error!
  };
```

2006167- Ankit Srivastava

Q13 Yes, a constructor can be private but can call it with member functions or friend functions. It will not be accessible in main funcⁿ.

In this case we may:

① create class using another (public) constructor

② calling constructor from the same class

③ calling constructor from friend class/function.

~~class Even {~~
~~int n;~~
~~Even(int a): n(a) {};~~
~~Public:~~
~~int get() const { return 2*n; };~~
~~static Even * make(int y) { return new Even(y); };~~
~~};~~

Private constructor is not accessible in main bcs, at time of initialization of object, the constructor cannot be called (as not publically accessible). and will throw error.

```
class A {
    A() {
        cout<<"constructor of A \n";
    }
    friend class B;
};

class B {
  Public:
    B() {
        A a1;
        cout<<"constructor of B\n";
    }
};
```

```
int main() {
    B b1;
    return 0;
}
```

OUTPUT

```
constructor of A
constructor of B
```

2006107- Ankit Srivastava

Q15 1

as static variables do ~~a~~ not contribute in the size of class and its instances.
Hence output : 1 ~~B~~ (1, ∵ class takes atleast 1B).

Even

2006167 – Ankit Srivastava

Q2

```
:
int fun ( int x, int y) {
        return x+y ;
}
double fun ( double x, double y) {
        return x*y;
}
:
int main(){
    cout<< fun ( 5,10);
    return 0;
}
```

OUTPUT: 15

//func overloading ~~count~~ checks the
// parameter list! (count / datatype)
// Not the func return-type.

Q4

```
void display ( int *p , int n) {
    for (int i=0 ; i<n ; i++)
            cout<< p[i] << " ";
    cout <<endl;
}
:
main→ :
int even =0 , odd =0;
for (int i=0 ; i<10; i++) {
        cin>> a[i];
        if( a[i] %2 ==0)
            even++;
        else
            odd++;
}
int *e = new int [even];
int *o = new int [odd];
}
for (int i=0, j=0, k=0; i<10; i++)
{
    if (a[i] %2 ==0) {
        e[j] = a[i];
        j++;
    }
    else {
        o[k] = a[i];
        k++;
    }
}
cout<<"even: ";
display (e, even);
cout << "odd: ";
display (o, odd);
delete e;
delete o;
return 0;
}
```

2006167 - Ankit Srivastava

Q6
```
class person {
        int age;
        Public:
        person (int a) { age = a; }

        Person max_age(person obj)
          {
              obj.age = obj.age > age ? obj.age : age;
              return obj;
          }
};
```
Mem. func^n definition

Q8 Yes, member func^n can be declared as private member but can only be called via a friend func^n/class or another ~~public~~ member func^n of the class.
We ~~can't~~ may access the func^n by calling another func^n in public which internally calls the private func^n.

* func^n overloading.

```
Int ~~void~~ product ( int a , int b , int c =1 , int d =1)
     {
         return (a*b*c*d);
     }

int main() {
   cout<< product (2,3);      // 2×3×1×1 = 6 ✓ output
   cout<< product (2,3,3);    // 18
   cout<< product (2,2,2,2);  // 16
}
```

Scanned with CamScanner

Q10 (i) Default constructor ⇒ 2

(ii) Parameterized constructor ⇒ 2

(iii) copy constructor ⇒ 1

(iv) Destructor ⇒ 5

Q12 ~~N~~ NULL

Q14

(i) void func (int i=5; int j=4; int k){ }

False, only trailing arguments can have default value.

(ii) Friendship is implicitly specified

True, friend func^n/classes are specified friend inside the class with 'friend' keyword in front.

(iii) Default constructor does not take any arguments,

True, constructor taking parameter are called parameterized constructor.

(iv) In the declaration class Demo{int a;}; a is private member

True, default access specifier of class private.

Q16 10 20 20

ASSIGNMENT

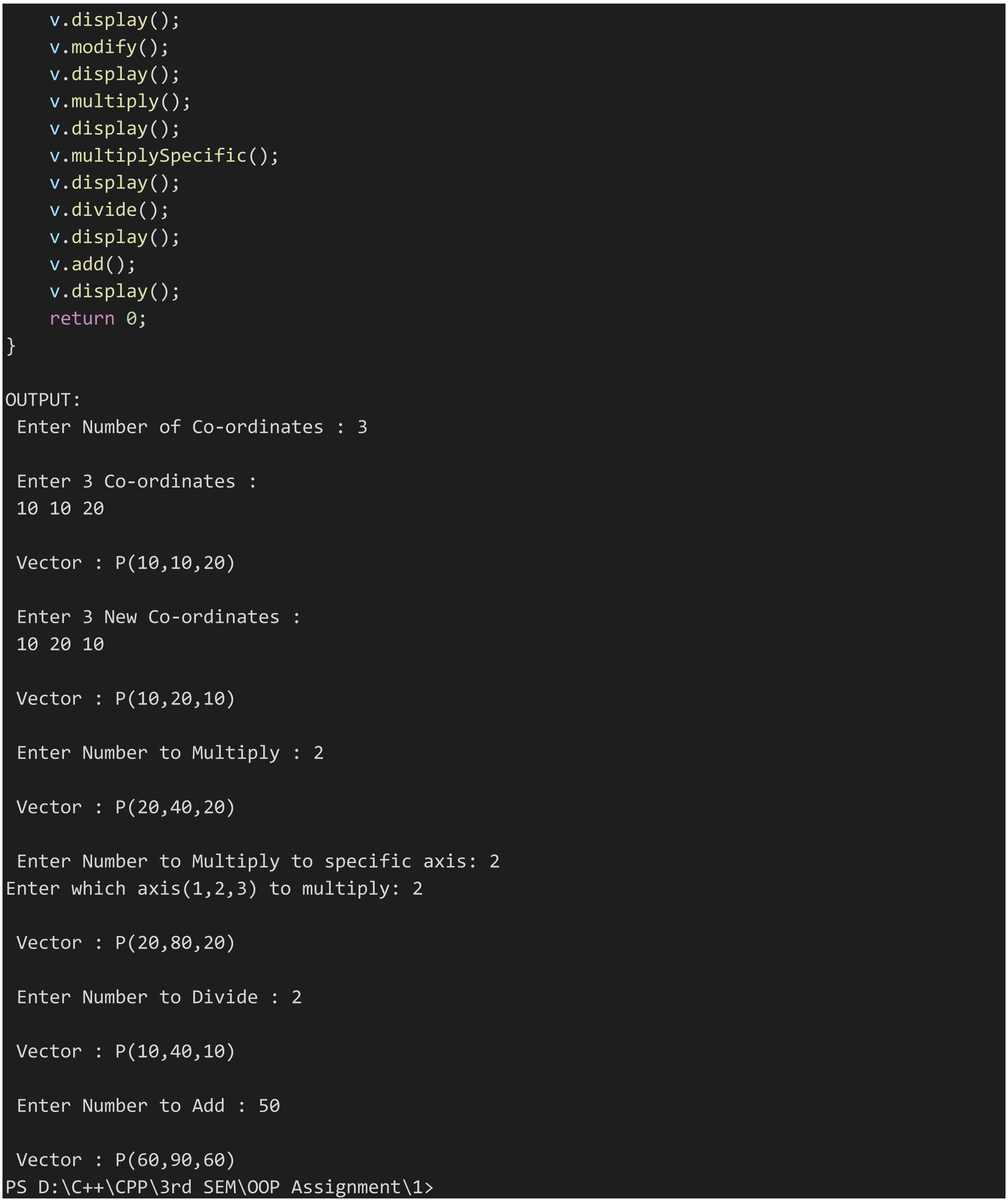
Programming Questions: (2006167-Ankit Srivastava)

1. WAP to add, multiply two polynomial using classes and objects.

```cpp
#include <iostream>
using namespace std;

class operation
{
    int *array1 = NULL;
    int *array2 = NULL;
    int n;
    int m;
    void getdata(int x, int *p);
    void display(int *ar, int n);

public:
    operation()
    {
        cout << "Enter Number Of Terms in Polynomial 1: ";
        cin >> n;
        cout << "Enter Number Of Terms in Polynomial 2: ";
        cin >> m;
        array1 = new int[n];
        array2 = new int[m];
        getdata(n, array1);
        getdata(m, array2);
    }
    void addition();
    void multiply();
};

void operation::getdata(int x, int *p)
{
    cout << "Enter Coeff of: \n";
    for (int i = 0; i < x; ++i)
    {
        cout << "x^" << i << ": ";
        cin >> p[i];
    }
}

void operation::display(int *ar, int n)
{
    for (int i = 0; i < n; i++)
    {
        cout << ar[i] << "x^" << i;
        if (i != n - 1)
        {
            cout << " + ";
        }
    }
    cout << endl
         << endl;
}

void operation::addition()
{
    int size = m > n ? m : n;
    int *sum = new int[size];
    for (int i = 0; i < size; i++)
        sum[i] = 0;
    for (int i = 0; i < n; i++)
    {
        sum[i] = array1[i];
    }
    for (int i = 0; i < m; i++)
    {
        sum[i] += array2[i];
    }
    cout << "After Addition: \n";
    display(sum, size);
    delete sum;
}

void operation::multiply()
{
    int size = m + n - 1;
    int *mul = new int[size];
    for (int i = 0; i < size; i++)
        mul[i] = 0;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < m; j++)
        {
            mul[i + j] += array1[i] * array2[j];
        }
    }
    cout << "After Multiplication: \n";
    display(mul, size);
    delete mul;
}
int main()
{
    operation obj;
    obj.addition();
    obj.multiply();
    return 0;
}

OUTPUT:
Enter Number Of Terms in Polynomial 1: 3
Enter Number Of Terms in Polynomial 2: 5
Enter Coeff of:
x^0: 2
x^1: 0
x^2: 10
Enter Coeff of:
x^0: 10
x^1: 2
x^2: 15
x^3: 0
x^4: 6
After Addition:
12x^0 + 2x^1 + 25x^2 + 0x^3 + 6x^4

After Multiplication:
20x^0 + 4x^1 + 130x^2 + 20x^3 + 162x^4 + 0x^5 + 60x^6

PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

2. WAP using classes and objects to represent the vector in 3 D space and include the member functions to perform the following tasks:
   - Create the vector
   - Modify the value of a given axis
   - Multiply by scalar value for a given axis
   - Multiply by scalar value for all axes
   - Display the vector in the form P(10, 20, 30)
   - Divide by scalar value for all axes
   - Add by scalar value for all axes.

```
#include <iostream>
#include <stdio.h>
using namespace std;

class vector
{
    int size;
    int *coord;

public:
    vector();
    void modify();
    void display();
    void multiply();
    void multiplySpecific();
    void divide();
    void add();
};
vector::vector()
{
    cout << "\n Enter Number of Co-ordinates : ";
    cin >> size;
    coord = new int[size];
    cout << "\n Enter " << size << " Co-ordinates : \n";
    for (int i = 0; i < size; i++)
    {
        cout << " ";
        cin >> coord[i];
    }
}
void vector::modify()
{
    cout << endl
         << "\n Enter " << size << " New Co-ordinates : \n";
    for (int i = 0; i < size; i++)
    {
        cout << " ";
        cin >> coord[i];
    }
}
void vector::multiply()
{
    int num;
    cout << endl
         << "\n Enter Number to Multiply : ";
    cin >> num;
    for (int i = 0; i < size; i++)
    {
        coord[i] = coord[i] * num;
    }
}

void vector::multiplySpecific()
{
    int num;
    cout << endl
         << "\n Enter Number to Multiply to specific axis: ";
    cin >> num;
    cout << "Enter which axis(1,2,3) to multiply: ";
    int ch;
    cin >> ch;
    coord[ch - 1] = coord[ch - 1] * num;
}

void vector::divide()
{
    int num;
    cout << endl
         << "\n Enter Number to Divide : ";
    cin >> num;
    for (int i = 0; i < size; i++)
    {
        coord[i] = coord[i] / num;
    }
}

void vector::add()
{
    int num;
    cout << endl
         << "\n Enter Number to Add : ";
    cin >> num;
    for (int i = 0; i < size; i++)
    {
        coord[i] = coord[i] + num;
    }
}
void vector::display()
{
    cout << "\n Vector : P(";
    for (int i = 0; i < size; i++)
    {
        cout << coord[i];
        if (i != size - 1)
            cout << ",";
    }
    cout << ")";
}
int main()
{
    vector v;
```



```
    v.display();
    v.modify();
    v.display();
    v.multiply();
    v.display();
    v.multiplySpecific();
    v.display();
    v.divide();
    v.display();
    v.add();
    v.display();
    return 0;
}

OUTPUT:
 Enter Number of Co-ordinates : 3

 Enter 3 Co-ordinates :
 10 10 20

 Vector : P(10,10,20)

 Enter 3 New Co-ordinates :
 10 20 10

 Vector : P(10,20,10)

 Enter Number to Multiply : 2

 Vector : P(20,40,20)

 Enter Number to Multiply to specific axis: 2
Enter which axis(1,2,3) to multiply: 2

 Vector : P(20,80,20)

 Enter Number to Divide : 2

 Vector : P(10,40,10)

 Enter Number to Add : 50

 Vector : P(60,90,60)
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

ASSIGNMENT

3. WAP that uses a date structure within a class. Enter any date and your birth date. The program must display your exact age in years, months and days.

```cpp
#include <iostream>
using namespace std;

class ageCalculator
{
    int pd, pm, py;
    int bd, bm, by;

public:
    ageCalculator()
    {
        cout << " Enter the present date in the format dd mm yyyy : ";
        cin >> pd >> pm >> py;
        cout << " Enter the birth date in the format dd mm yyyy : ";
        cin >> bd >> bm >> by;
    }
    void age();
};

void ageCalculator::age()
{
    int d, m, y;
    int md[12];
    y = py - by;
    if (pm < bm)
    {
        y--;
        m = 12 - (bm - pm);
    }
    else
    {
        m = pm - bm;
    }
    if (pd < bd)
    {
        m--;
        d = md[pm - 1] - (bd - pd);
    }
    else
    {
        d = pd - bd;
    }
    cout << "your age is : ";
    cout << y << " years " << m << " months " << d << " days. ";
}

int main()
{
    ageCalculator a;
    a.age();
    return 0;
}

OUTPUT:
Enter the present date in the format dd mm yyyy : 30 08 2021
Enter the birth date in the format dd mm yyyy : 27 11 2002
your age is : 18 years 9 months 3 days.
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

4. WAP that uses a class within a class.

```
#include <iostream>
using namespace std;
class A
{
public:
    class B
    {
    private:
        int num;

    public:
        void getdata(int n)
        {
            num = n;
        }
        void putdata()
        {
            cout << "The number is " << num;
        }
    };
};
int main()
{
    cout << "Nested classes in C++" << endl;
    A ::B obj;
    obj.getdata(9);
    obj.putdata();
    return 0;
}

OUTPUT:
Nested classes in C++
The number is 9
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

5. WAP that reads records of n students and find the

- average mark of each student
- # of students above average mark in the class.
- # of students below average mark in the class.
- Sort students in ascending order of their mark.
- Display the name of the student secured highest mark.

Display the roll number of the student secured highest mark from bottom.

```
#include <iostream>
#include <string>
#include <iomanip>
using namespace std;

class students
{
    int marks[5];
    long long int roll;
    static int classAvg;
    static int above;
    static int below;
    static int equal;

public:
    string name;
    int studentAvg;
    void getdata();
    void avg_marks();
    void display();
    friend void class_avg(int n);
    friend void compare_avg(int avg);
    friend void printCompare();
};

int students::classAvg = 0;
int students::above = 0;
int students::below = 0;
```

ASSIGNMENT

```
int students::equal = 0;

void students::getdata()
{
    cin.ignore();
    cout << "Enter Name: ";
    getline(cin, name);
    cout << "Enter Roll Number: ";
    cin >> roll;
    cout << "Enter Marks in 5 Subjects: ";
    for (int i = 0; i < 5; ++i)
    {
        cin >> marks[i];
    }
    avg_marks();
}
void students::avg_marks()
{
    studentAvg = 0;
    for (int i = 0; i < 5; ++i)
    {
        studentAvg += marks[i];
    }
    studentAvg /= 5;
    classAvg += studentAvg;
}

void class_avg(int n)
{
    students::classAvg /= n;
}

void compare_avg(int avg)
{
    if (avg > students::classAvg)
    {
        ++students::above;
    }
    else if (avg < students::classAvg)
    {
        ++students::below;
    }
    else
    {
        ++students::equal;
    }
}
```

```cpp
void printCompare()
{
    cout << "No. of Students above Class Avg: " << students::above << endl;
    cout << "No. of Students equal Class Avg: " << students::equal << endl;
    cout << "No. of Students below Class Avg: " << students::below << endl;
}

void students::display()
{
    cout << name << setw(10) << roll << setw(10) << studentAvg << endl;
}

int main()
{
    int n;
    cout << "Enter Number of Students: ";
    cin >> n;
    students *ar = new students[n];
    for (int i = 0; i < n; ++i)
    {
        ar[i].getdata();
    }
    class_avg(n);
    for (int i = 0; i < n; ++i)
    {
        compare_avg(ar[i].studentAvg);
    }
    printCompare();
    for (int i = 0; i < n; ++i)
    {
        int counter = 0;
        while (counter < n - 1)
        {
            for (int i = 0; i < n - 1; i++)
            {
                if (ar[i].studentAvg > ar[i + 1].studentAvg)
                {
                    students temp = ar[i];
                    ar[i] = ar[i + 1];
                    ar[i + 1] = temp;
                }
            }
            counter++;
        }
    }
    int max = -1, flag;
    cout << "Name" << setw(10) << "Roll" << setw(10) << "Avg Marks" << endl;
    for (int i = 0; i < n; i++)
    {
        ar[i].display();
        if (max <= ar[i].studentAvg)
            flag = i;
    }
    cout << "Topper of class: " << ar[flag].name << endl;
    return 0;
}

OUTPUT:
Enter Number of Students: 3
Enter Name: Ankit
Enter Roll Number: 2006167
Enter Marks in 5 Subjects: 25 25 25 25 25
Enter Name: Arundhati
Enter Roll Number: 27112002
Enter Marks in 5 Subjects: 100 100 100 100 100
Enter Name: unknown
Enter Roll Number: 2006
Enter Marks in 5 Subjects: 50 50 50 50 50
No. of Students above Class Avg: 1
No. of Students equal Class Avg: 0
No. of Students below Class Avg: 2
Name          Roll        Avg Marks
Ankit         2006167     25
unknown       2006        50
Arundhati     27112002    100
Topper of class: Arundhati
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

6. Write a program to enter the code, name and price of items. The user must feed the quantity in which he wants a product. The program must calculate and display the final bill of 10 items in the following format.

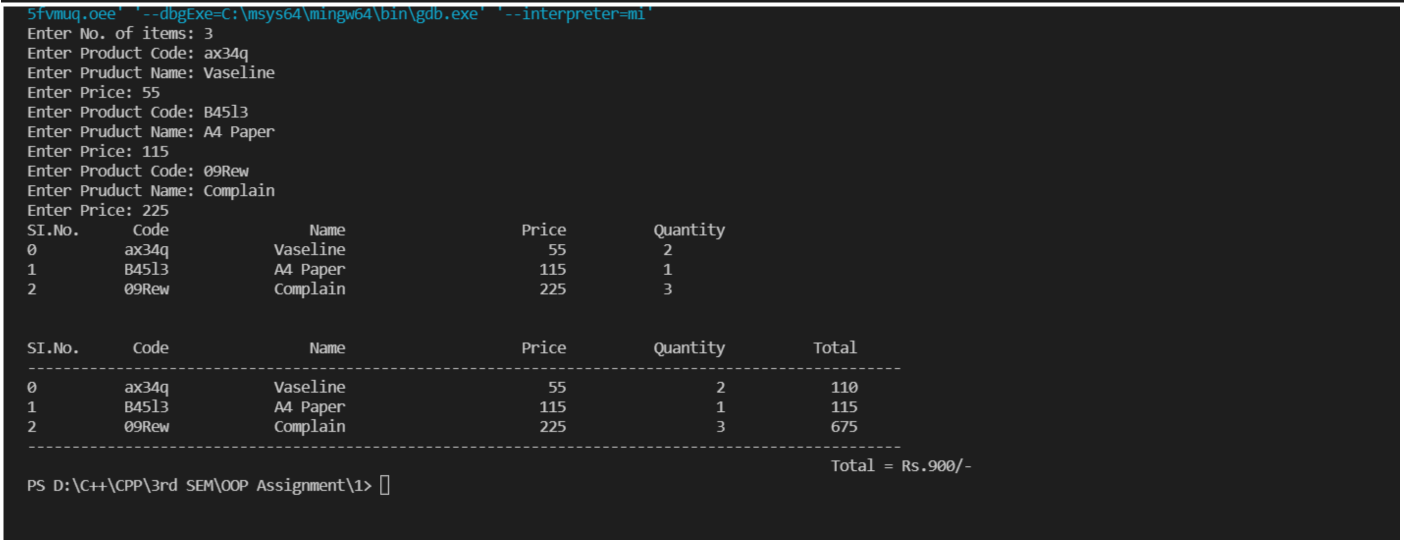
```
Sl.No.    Code      Name            Price      Quantity      Total
-------------------------------------------------------------------------------------
1.          rib001   Printercatrige     300              2         600
2.          Pap45   A4 size paper     200              0           0
3.          Bk216   Computer book    350              5        1750
-------------------------------------------------------------------------------------
                                                      Total = Rs.2350/-
```

ASSIGNMENT

```
#include <iostream>
#include <iomanip>
#include <string>
using namespace std;

class bill
{
    int serial;
    string code;
    string name;
    int price;
    int quantity;
    int total;
    static int netTotal;

public:
    void getProduct()
    {
        cin.ignore();
        cout << "Enter Product Code: ";
        getline(cin, code);
        cout << "Enter Pruduct Name: ";
        getline(cin, name);
        cout << "Enter Price: ";
        cin >> price;
    }
    void getQuantity(int n)
    {
        serial = n;
        cout << serial << setw(15) << code << setw(20) << name << setw(25) << price << "\t
\t";
        cin >> quantity;
    }
    void displayBill()
    {
        total = quantity * price;
        netTotal += total;
        cout << serial << setw(15) << code << setw(20) << name << setw(25) << price << set
w(18) << quantity << setw(15) << total << endl;
    }
    friend void displaytotal();
};

int bill::netTotal = 0;

void displaytotal()
{
```

```cpp
    cout << setw(102) << "Total = Rs." << bill::netTotal << "/-" << endl;
}

int main()
{
    int n;
    cout << "Enter No. of items: ";
    cin >> n;
    bill *p = new bill[n];
    for (int i = 0; i < n; ++i)
    {
        p[i].getProduct();
    }
    cout << "SI.No." << setw(10) << "Code" << setw(20) << "Name" << setw(25) << "Price" <<
 setw(18) << "Quantity" << endl;
    for (int i = 0; i < n; ++i)
    {
        p[i].getQuantity(i);
    }
    cout << endl
         << endl;
    cout << "SI.No." << setw(10) << "Code" << setw(20) << "Name" << setw(25) << "Price" <<
 setw(18) << "Quantity" << setw(15) << "Total" << endl;
    cout << "-----------------------------------------------------------------------------
----------------------\n";
    for (int i = 0; i < n; ++i)
    {
        p[i].displayBill();
    }
    cout << "-----------------------------------------------------------------------------
----------------------\n";
    displaytotal();
    return 0;
}
```

OUTPUT:



```
5fvmuq.oee' '--dbgExe=C:\msys64\mingw64\bin\gdb.exe' '--interpreter=mi'
Enter No. of items: 3
Enter Product Code: ax34q
Enter Pruduct Name: Vaseline
Enter Price: 55
Enter Product Code: B4513
Enter Pruduct Name: A4 Paper
Enter Price: 115
Enter Product Code: 09Rew
Enter Pruduct Name: Complain
Enter Price: 225
SI.No.      Code                Name                    Price           Quantity
0          ax34q            Vaseline                       55             2
1          B4513            A4 Paper                      115             1
2          09Rew            Complain                      225             3


SI.No.      Code                Name                    Price           Quantity            Total
---------------------------------------------------------------------------------------------------
0          ax34q            Vaseline                       55                   2              110
1          B4513            A4 Paper                      115                   1              115
2          09Rew            Complain                      225                   3              675
---------------------------------------------------------------------------------------------------
                                                                                          Total = Rs.900/-
PS D:\C++\CPP\3rd SEM\OOP Assignment\1> ▯
```

ASSIGNMENT

7. Define a class named as FRACTION that contains two data members that represent the numerator and denominator of a fraction. By defining necessary member functions, write a program to add, subtract and multiply two fractions. The add accepts the objects using callby- value technique, subtract using call-by-reference and multiply using call-by-address technique.

Sample input/Output
For Fraction-1
Enter the numerator: 3
Enter the denominator: 5
For Fraction-2
Enter the numerator: 4
Enter the denominator: 9
Result of addition = 47/45
Result of subtraction = 7/45
Result of multiplication = 4/15

```
#include <iostream>
using namespace std;

int findGCD(int n1, int n2)
{
    int gcd;
    for (int i = 1; i <= n1 && i <= n2; i++)
    {
        if (n1 % i == 0 && n2 % i == 0)
            gcd = i;
    }
    return gcd;
}

class fraction
{
    int num;
    int den;

public:
    fraction()
    {
        cout << "Enter the numerator: ";
        cin >> num;
        cout << "Enter the denominator: ";
        cin >> den;
```

```cpp
    }
    void addition(fraction *p)
    {
        int lcm = (p->den * den) / findGCD(p->den, den);
        int sum = (p->num * lcm / p->den) + (num * lcm / den);
        int num3 = sum / findGCD(sum, lcm);
        lcm = lcm / findGCD(sum, lcm);
        cout << "Result of Addition = " << num3 << "/" << lcm << endl;
    }
    void substraction(fraction *p)
    {
        int lcm = (p->den * den) / findGCD(p->den, den);
        int sum = (p->num * lcm / p->den) - (num * lcm / den);
        int num3 = sum / findGCD(sum, lcm);
        lcm = lcm / findGCD(sum, lcm);
        cout << "Result of Substraction = " << num3 << "/" << lcm << endl;
    }
    void multiplication(fraction *p)
    {
        int numerator = num * p->num;
        int denominator = den * p->den;
        for (int i = denominator * numerator; i > 1; i--)
        {
            if ((denominator % i == 0) && (numerator % i == 0))
            {
                denominator /= i;
                numerator /= i;
            }
        }
        cout << "Result of Multiplication = " << numerator << "/" << denominator << endl;
    }
};

int main()
{
    cout << "For Fraction-1:\n";
    fraction f1;
    cout << "For Fraction-2:\n";
    fraction f2;
    f2.addition(&f1);
    f2.substraction(&f1);
    f2.multiplication(&f1);
    return 0;
}
```

```
OUTPUT:
For Fraction-1:
Enter the numerator: 3
Enter the denominator: 5
For Fraction-2:
Enter the numerator: 4
Enter the denominator: 9
Result of Addition = 47/45
Result of Substraction = 7/45
Result of Multiplication = 4/15
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```

8.
If a program can be written using either function overloading or using default arguments, which one is preferable?Explain giving example.

Write overloaded functions to find the area of scalene,isosceles and equilateral triangle
Area of scalene triangle = $\sqrt{(s(s-a)(s-b)(s-c))}$
Area of isosceles triangle = $\frac{1}{2}[\sqrt{(a^2 - b^2/4)} \times b]$
Area of equilateral triangle= $(\sqrt{3}/4)a^2$

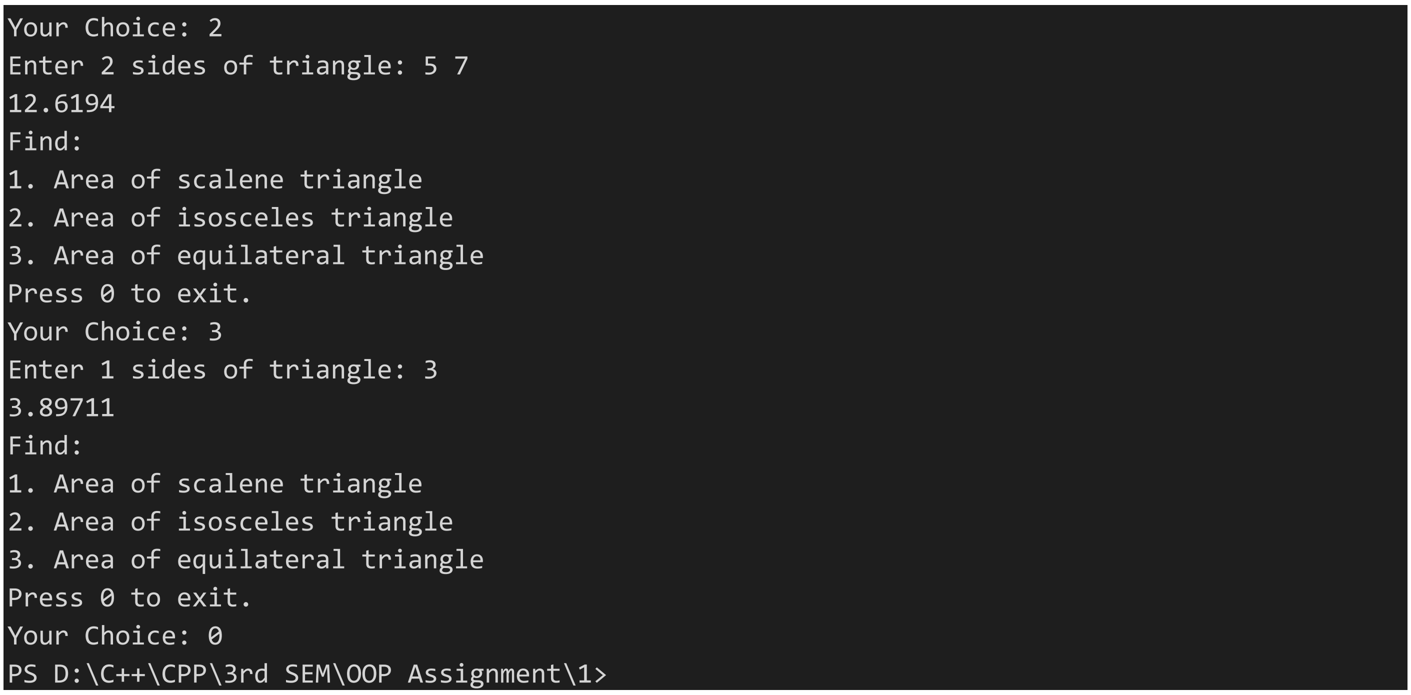
```cpp
#include <iostream>
#include <math.h>
using namespace std;

void areaTriangle(int a)
{
    float area = sqrt(3) / 4 * a * a;
    cout << area;
}

void areaTriangle(int a, int b)
{
    float area = (sqrt(((a * a) - (b * b / 4))) * b) / 2;
    cout << area;
}

void areaTriangle(int a, int b, int c)
{
    float s = (a + b + c) / 2;
    float area = sqrt((s * (s - a) * (s - b) * (s - c)));
    cout << area;
}
```

ASSIGNMENT

```
int main()
{
    cout << "Find:\n";
    cout << "1. Area of scalene triangle\n";
    cout << "2. Area of isosceles triangle\n";
    cout << "3. Area of equilateral triangle\n";
    cout << "Your Choice: ";
    int n;
    cin >> n;
    switch (n)
    {
        int a, b, c;
    case 1:
        cout << "Enter 3 sides of triangle: ";
        cin >> a >> b >> c;
        areaTriangle(a, b, c);
        break;
    case 2:
        cout << "Enter 2 sides of triangle: ";
        cin >> a >> b;
        areaTriangle(a, b);
        break;
    case 3:
        cout << "Enter 1 sides of triangle: ";
        cin >> a;
        areaTriangle(a);
        break;
    default:
        cout << "Enter Valid Option!\n";
    }
}
```

OUTPUT:

```
Find:
1. Area of scalene triangle
2. Area of isosceles triangle
3. Area of equilateral triangle
Press 0 to exit.
Your Choice: 1
Enter 3 sides of triangle: 3 4 5
6
Find:
1. Area of scalene triangle
2. Area of isosceles triangle
3. Area of equilateral triangle
Press 0 to exit.
```



```
Your Choice: 2
Enter 2 sides of triangle: 5 7
12.6194
Find:
1. Area of scalene triangle
2. Area of isosceles triangle
3. Area of equilateral triangle
Press 0 to exit.
Your Choice: 3
Enter 1 sides of triangle: 3
3.89711
Find:
1. Area of scalene triangle
2. Area of isosceles triangle
3. Area of equilateral triangle
Press 0 to exit.
Your Choice: 0
PS D:\C++\CPP\3rd SEM\OOP Assignment\1>
```