Teacher's Sign / Remarks

classmate Date Page

Prerequisite Programs

Q1 Write C++ program for addition of two numbers.

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

void main(){
    int a, b, sum;
    cout << "Enter values of a & b";
    cin >> a>>b;
    sum = a+b;
    cout << "The addition is " << sum;
}
```

Q2 Write a C++ program to perform arithmetic operation using switch case

```
#include <iostream.h>
using namespace std;
void main(){
    double num1, num2;
    char op;
    cout << "Enter two numbers";
    cin >> num1 >> num2
    cout << "Enter operator :";
    cin >> op;
    switch (op) {
        case '+':
            cout << "Result" << (num1 + num2) << endl;
            break
```

```
case '-':
cout << "Result" << (num1 - num2) << endl;
break;

case '*':
cout << "Result" << (num1 * num2) << endl;
break;

case '/':
if(num2 != 0) cout<< "Result :" << (num1 / num2)
else cout << "Error
}
case '%':
  if( num2 != 0 )
  cout << "Result << (num1 % num2) endl

default:
cout << "Invalid operator!" <<endl;

}
```

Q3 Check if even or odd

```
#include <iostream.h>
using namespace std;
int main(){
   int num;
   cout << "Enter a number";
   cin >> num;
   cout num%2 == 0 ? cout<<"odd" : cout << "Even";<<endl;
}
```

Q4 1-10 using for loop

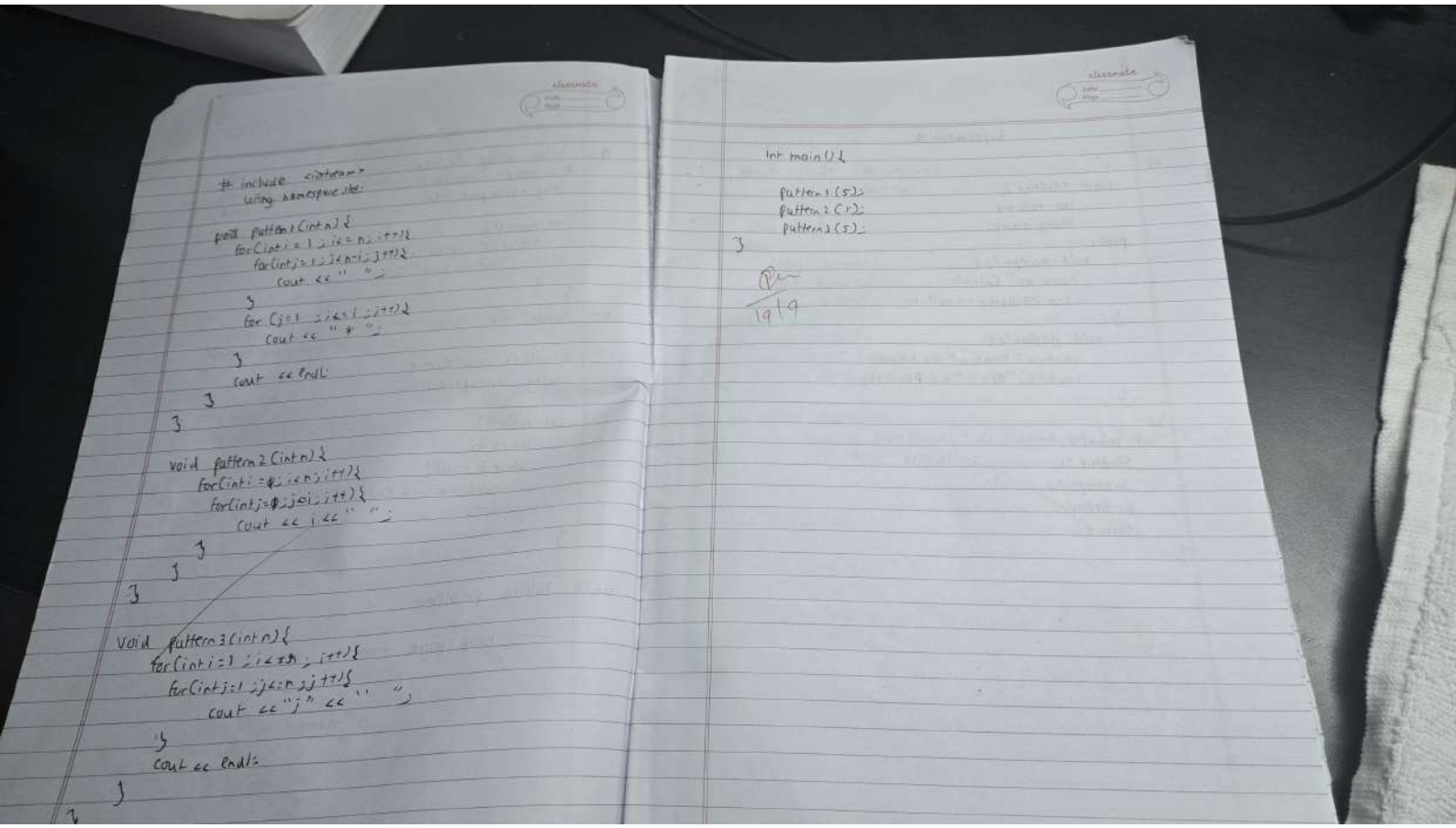
```
# include <iostream>
using namespace std;

int main (){
   for(int i = 1; i<= 10 ; i++){
      cout << i << " ";
   }
}
```

Q5 While loop 10-1

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

int main(){
   int i = 10
   while (i != 0){
      cout << i << endl;
      i--;
   }
}
```

Q6,7,8 Pattern problems

— next page —



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

void pattern1(int n){
    for(int i=1; i<=n; i++){
        for(int j=1; j<n-i; j++){
            cout << " ";
        }
        for(j=1; j<=i; j++){
            cout << "*";
        }
        cout << endl;
    }
}

void pattern2(int n){
    for(int i=0; i<n; i++){
        for(int j=0; j<=i; j++){
            cout << i+1 << " ";
        }
    }
}

void pattern3(int n){
    for(int i=1; i<=n; i++){
        for(int j=1; j<=n; j++){
            cout << "j" << " ";
        }
        cout << endl;
    }
}
```

```
int main(){

    pattern1(5);
    pattern2(5);
    pattern3(5);
}
```

19/9

Experiment 7

1)

```
class Student {
    int roll_no;
    String name;
public :
    void accept() {
        cout << "Enter";
        cin >> name >> roll_no;
    }
    void display() {
        cout << "Name :" << name;
        cout << "Roll : " << Roll_no;
    }
};
int main() {
    Student S1;
    S1.accept();
    S1.display();
    return 0;
}
```

2)

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

class Book {
public :
    String b_name;
    int b_price;
    int b_pages;
public :
    void accept() {
        cout << "Enter the data";
        cin >> b_name >> b_price >> b_pages;
    }
    void display() {
        cout << "Book Name :" << b_name << b_Price << "₹" << b_Pages
        << "Pages." << endl;
    }
};
    int main() {
        book B1, B2;
        B1.accept();
        B2.accept();
        B1.price > B2.price ? B1.display() : B2.display();
    }
```

3)

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

class time{
    int h,m,s;
    char c;
public:
    void accept(){
        cout<<"Enter time in HH:MM:SS";
        cin>>h>>c>>m>>c>>s;
    }
    void display(){
        int total_seconds = h*3600 + m*60 + s;
        cout<<"Time in seconds:"<<totalseconds<<endl;
    }
};
int main(){
    time t;
    t.accept();
    t.display();
}
```

1)

Experiment 7

1

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

class city{
public:
    int popul;
    string name;

public:
    void accept(){
        cout<<"City:"<<"population:"<endl;
        cin>>name>>popul;
    }
    void display(){
        cout<<"Name:"<<name<<endl;
        cout<<"population:"<<popul<<endl;
    }
};
int main(){
    city c[5];
    int max_pop=0;
    for(int i=0;i<5;i++){
        c[i].accept();
    }
    for(int i=0;i<5;i++){
        if(c[i].popul > c[max_popul].popul){
            max_pop = i;
        }
    }
    cout<<"City with maximum population"<<c[max_pop].display();
}
```

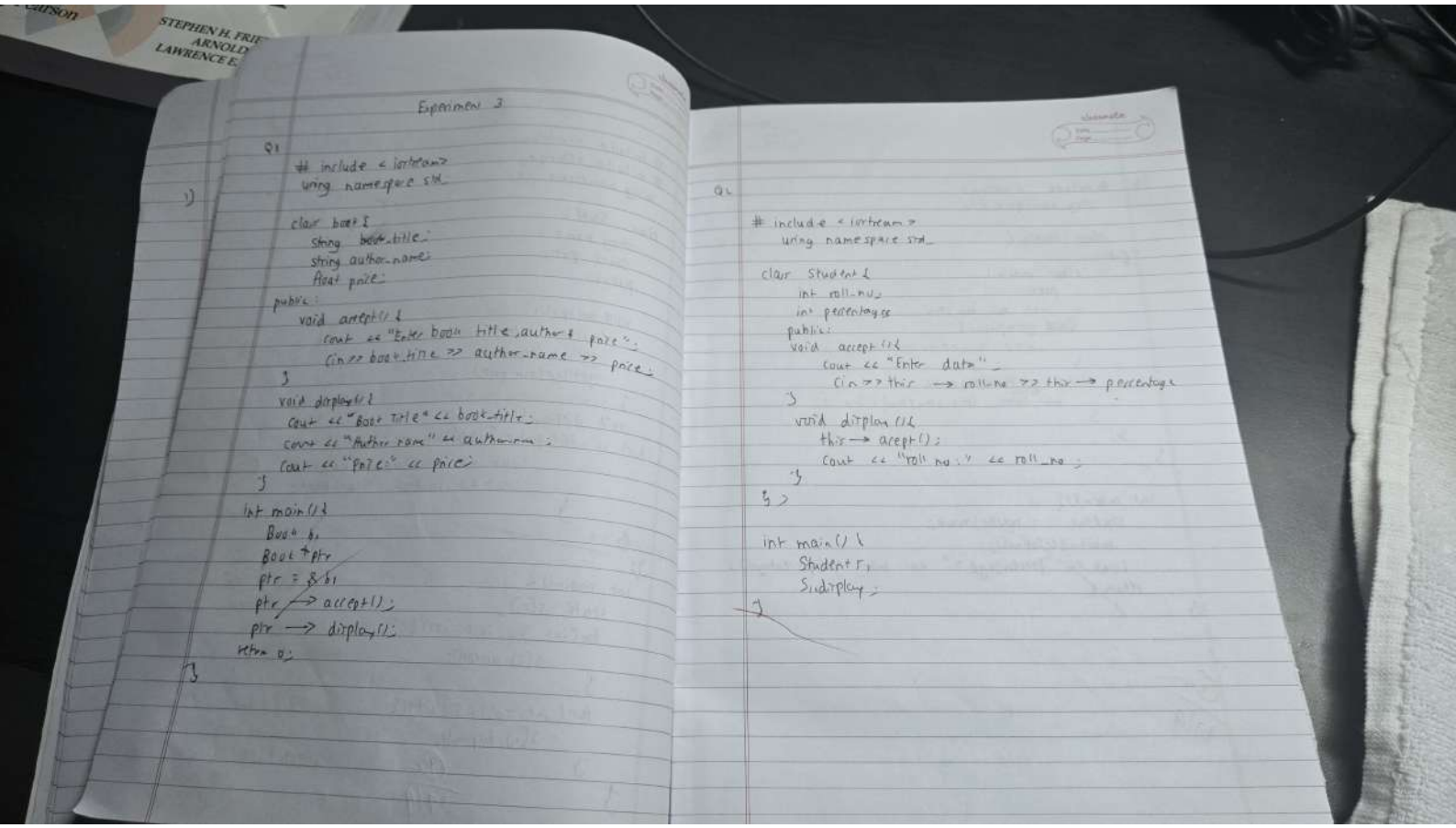
Q2

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

class account{
  int acc_no;
  int balance;

  void accept(){
    cout << "Enter the details";
    cin >> acc_no >> balance;
  }
  void display(){
    if(balance > 1000){
      cout << "Account No: " << acc_no << endl;
      balance = balance * 0.1;
      cout << " balance: " << balance;
    }
    cout << "Acc no: " << acc_no << "balance: " << balance;
  }
int main(){
  account a[10];
  for(int i=0; i<10; i++){
    a[i].accept();
  }
  cout << "greater balance than 1000 :";
  for(int i=0; i<10; i++){
    a[i].display();
  }
}
```

Q3

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

class Staff{
  string name;
  string post;
  public:

  void accept(){
    cout << "Enter
    getline(cin, name);
    getline(cin, post);
  }
  void display(){
    if(post == "HOD" || post == "hod"){
      cout << "\n Name :" << name;
      cout << "\n Post :" << post;
    }
  }
};
int main(){
  Staff s[5];
  for(int i=0; i<5; i++){
    s[i].accept();
  }
  for(int i=0; i<5; i++){
    s[i].display();
  }
}
```

19/9



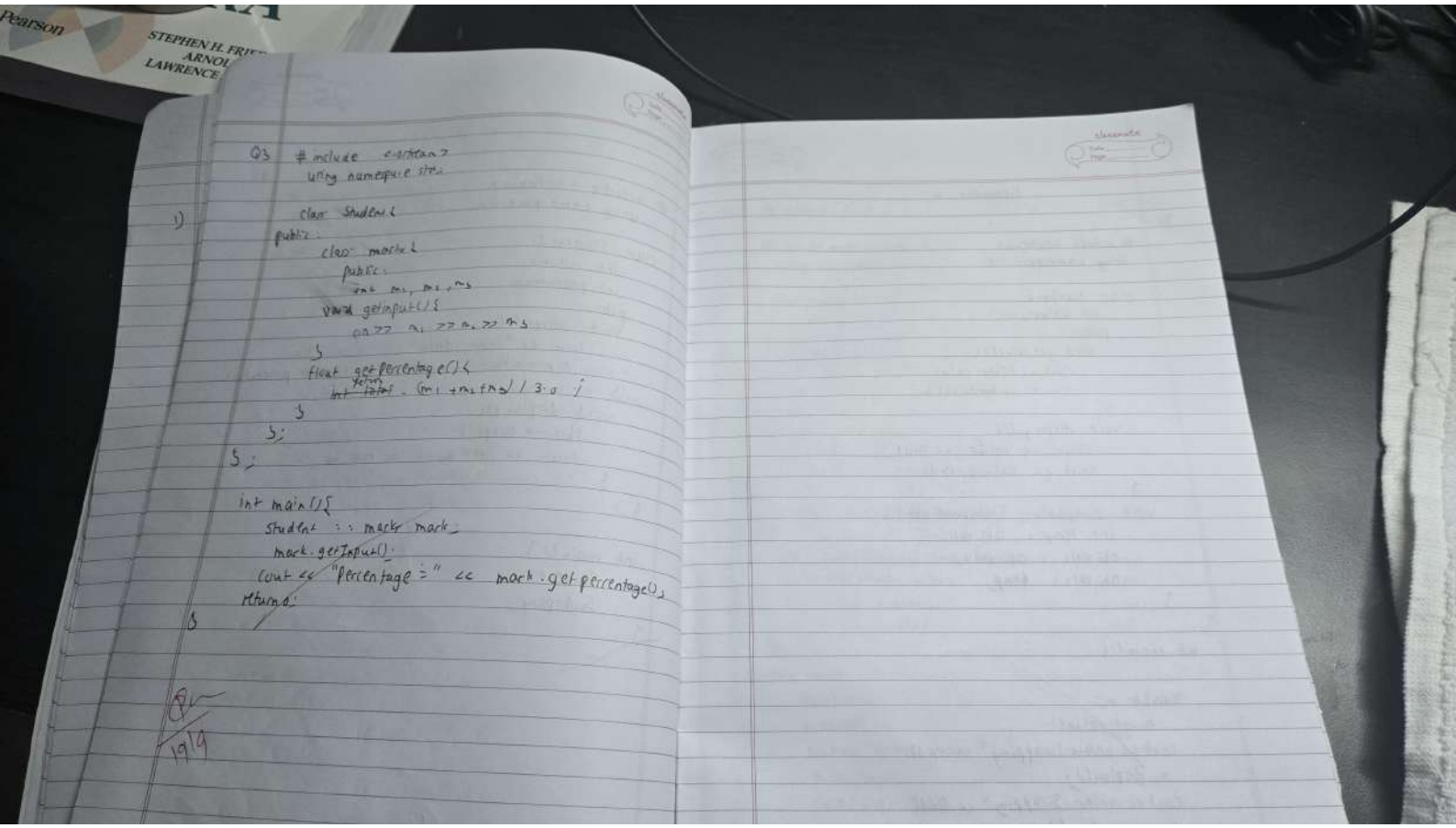
Experiment 3

Q1

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

class book {
    string book_title;
    string author_name;
    float price;
public:
    void accept() {
        cout << "Enter book title, author & price";
        cin >> book_title >> author_name >> price;
    }
    void display() {
        cout << "Book Title" << book_title;
        cout << "Author name" << author_name;
        cout << "price:" << price;
    }
int main() {
    Book b;
    Book *ptr
    ptr = &b;
    ptr -> accept();
    ptr -> display();
    return 0;
}
```

Q2

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

class Student {
    int roll_no;
    int percentage
public:
    void accept() {
        cout << "Enter data";
        cin >> this -> roll_no >> this -> percentage
    }
    void display() {
        this -> accept();
        cout << "roll no:" << roll_no;
    }
};

int main() {
    Student s;
    s.display;
}
```


Pearson
STEPHEN H. FRI
ARNOL
LAWRENCE
Q3 #include <iostream>
using namespace std;
class Student {
public:
class marks {
public:
int m1, m2, m3
void getInput() {
cin >> m1 >> m2 >> m3
}
float getPercentage() {
int total = (m1 + m2 + m3) / 3.0 ;
}
};
};
int main() {
Student :: marks mark;
mark.getInput();
cout << "Percentage = " << mark.getPercentage();
return 0;
}
19/9

Experiment 4

Q1

1)

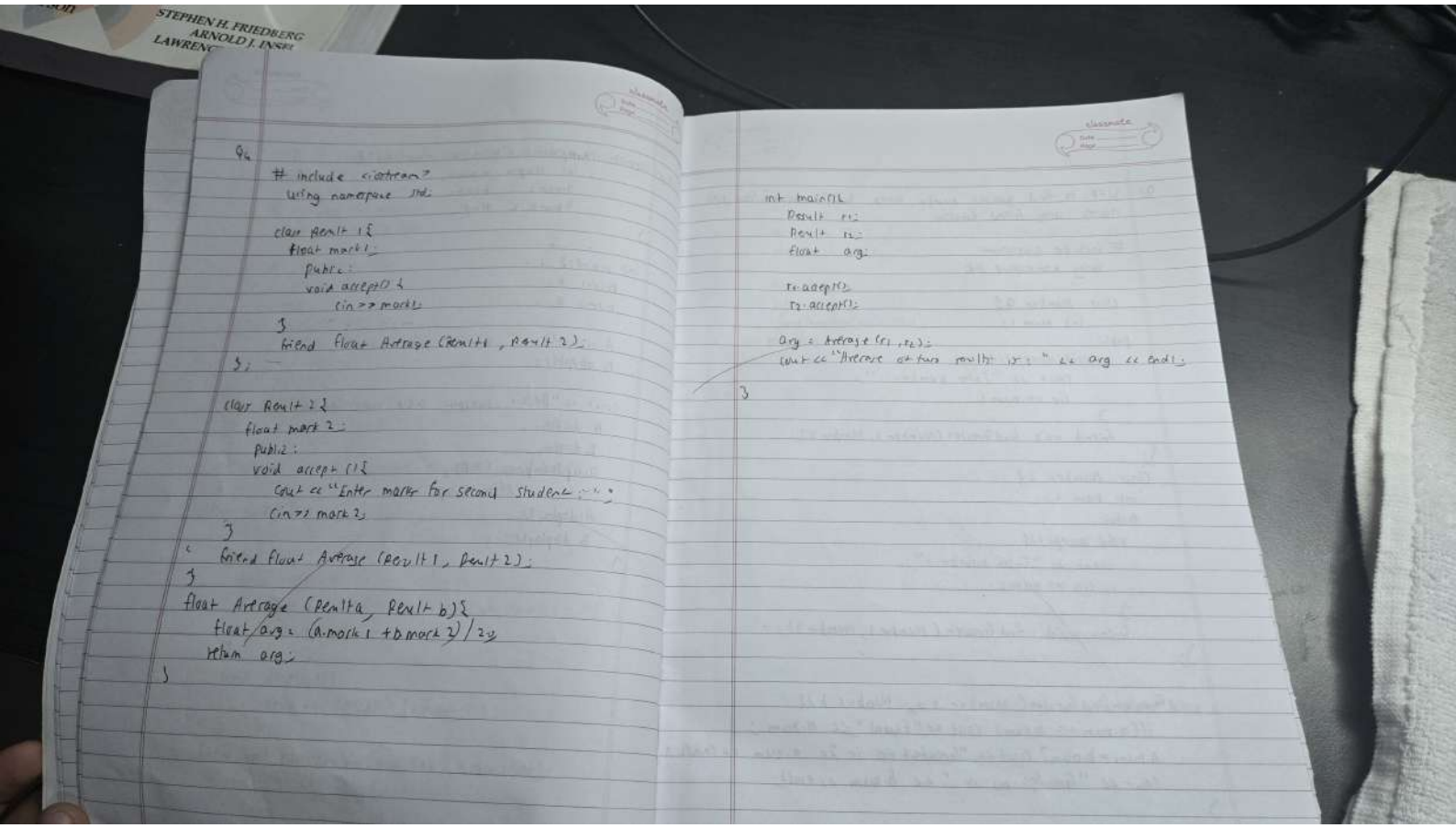
```
#include <iostream>
using namespace std;

class Number{
    int value1, value2;
public:
    void getvalue(){
        cout << "Enter value : ";
        cin >> value1 >> value2;
    }
    void display(){
        cout << value1 << endl;
        cout << value2 << endl;
    }
    void swapValue (Number &obj){
        int temp = obj.val1;
        obj.val1 = obj.val2;
        obj.val2 = temp;
    }
};
int main(){

    Number n;
    n.getvalue();
    cout << "before swapping" << endl;
    n.display();
    cout << "after swapping" << endl;
    n.swapvalue();
    n.display();
}
```

Q2

```
# friend function

#include <iostream>
using namespace std;

class Number{
    int n1, n2;
public:
    void accept(){
        cout << "Enter the number";
        cin >> n1 >> n2;
    }
    void display(){
        cout << "n1 = " << n1 << endl;
        cout << "n2 = " << n;
    }
    friend void swapNumber(Number &n);
};
void swapNumber(Number &n){
    int temp = n.n1;
    n.n1 = n.n2;
    n.n2 = temp;
}
int main (){
    Number n;
    n.accept;
    cout << "Before swap :";
    n.display();
    swapNumber (n);
    cout << "After swap :";
    n.display();
}
```

Q3 WAP to swap 2 number from different classes using friend function

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

class num1{
    int n;
  public:
    void accept(){
      cout<<"Enter number in class 1: ";
      cin>>n;
    }
    void display(){
      cout<<"class 1 "<<n;
    }
    friend void swapNumber(num1&obj, num2&ob);
};
class num2{
    int m;
  public:
    void accept(){
      cout<<"Enter the number in class 2: ";
      cin>>m;
    }
    void display(){
      cout<<"class 2 "<<m;
    }
  friend void swapNumber(num1&obj, num2&ob);
};

void swapNumber(num1&a, num2&b){
    int temp=a.num1;
    a.num1 = b.num2;
    b.num2 = temp;
}

int main(){
  num1 A;
  num2 B;

  A.accept();
  B.accept();

  cout<<"Before swapping a&b are: "<<a<<b;
  A.display();
  B.display();
  swapNumber(A,B);
  cout<<"After swap";
  A.display();
  B.display();
}
```


STEPHEN H. FRIEDBERG
ARNOLD J. INSEL
Q4
#include <iostream>
using namespace std;
class Result 1 {
float mark1;
public:
void accept() {
cin >> mark1;
}
friend float Average (Result1, Result 2);
};
class Result 2 {
float mark 2;
public2:
void accept () {
cout << "Enter marks for second student :";
cin >> mark 2;
}
friend float Average (Result 1, Result 2);
}
float Average (Result a, Result b) {
float avg = (a.mark 1 + b.mark 2) / 2;
return avg;
}
int main() {
Result r1;
Result r2;
float avg;
r1.accept();
r2.accept();
avg = Average (r1, r2);
cout << "Average of two results :" << avg << endl;
}

Q: WAP to find greater number among two numbers from different class using friend function.

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

class Number 2{
    int num 1;
public:
    void accept(){
        cout << "Enter number: ";
        cin >> num 1;
    }
    friend void findGreater(Number 1, Number 2);
};
class Number 2{
    int num 2;
public:
    void accept(){
        cout << "Enter number: ";
        cin >> num 2;
    }
    friend void findGreater(Number 1, Number 2);
};

void findGreater(Number 1 a, Number 2 b){
    if(a.num == b.num) cout << "Equal" << a.num;
    a.num > b.num? cout << "Greater no is" << a.num << endl
    cout << "Greater no is" << b.num << endl;
}
```

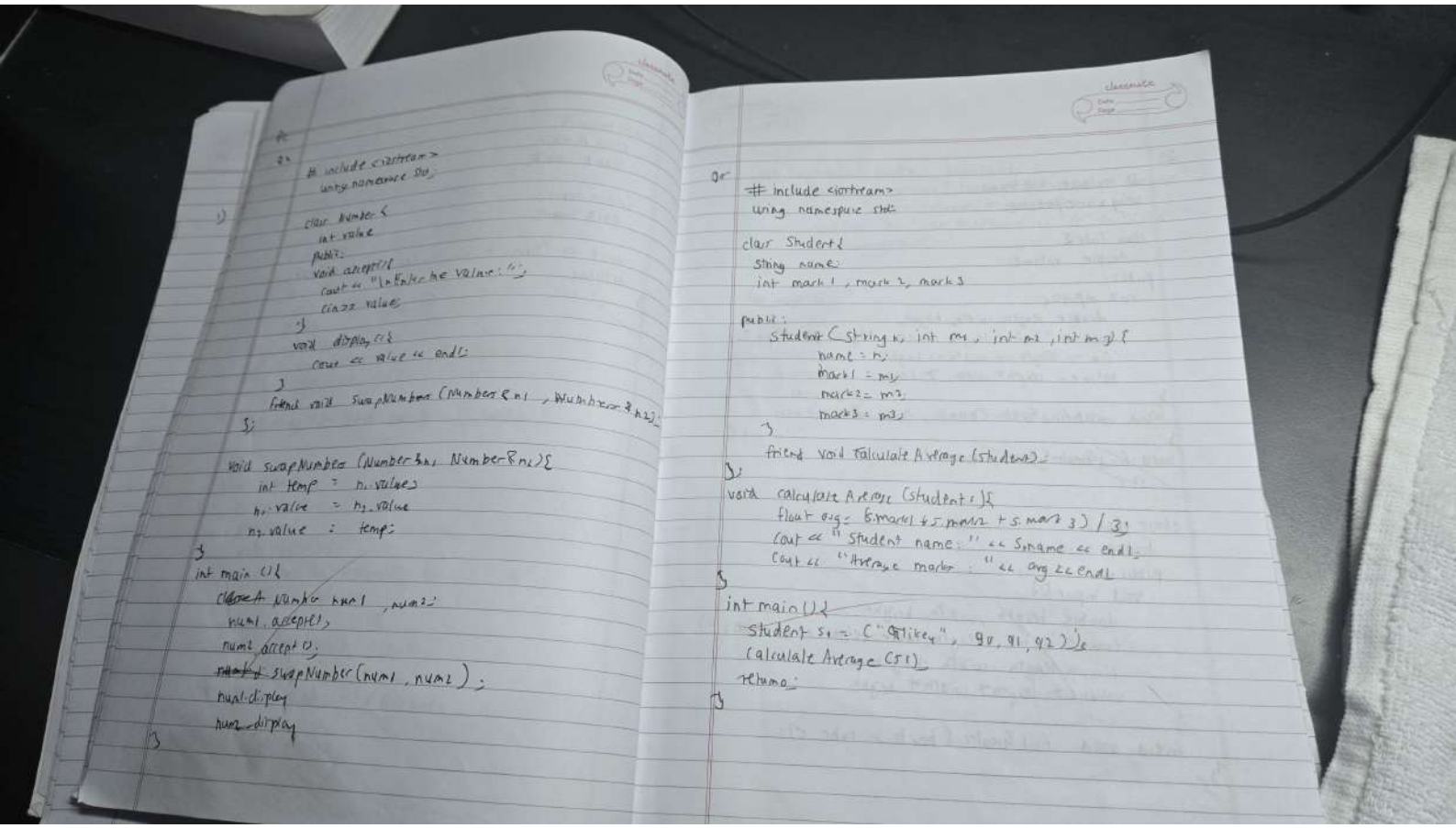
```
int main(){

    Number 1 obj1;
    Number 2 obj2;

    obj1.accept();
    obj2.accept();
    findGreater(obj1, obj2);
    return 0;
}
```

Practice Question

Q1.

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

class Number{
int value;

class ClsA{
int valueA;
public:
void input(){
cout<< "Enter value for ClassA: ";
cin >> valueA;
}
friend int sum( clsA a , clsB b);

};
class ClsB{
int valueB;
public:
void input(){
cout<< "Enter value for ClsB";
cin >> valueB;
}
friend int sum(clsA a, clsB b);
int Sum ( clsA a, ClsB b){
return a.valueA + b.valueB;
}
};
```

```
int main(){
ClassA objA;
ClassB objB;

objA.input();
objB.input();

cout << "Sum: " << Sum(objA, objB) << endl;
return 0;
}
```



Q2

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

class Number {
    int value;
public:
    void accept() {
        cout << "Enter the value: ";
        cin >> value;
    }
    void display() {
        cout << value << endl;
    }
    friend void swapNumbers(Number &n1, Number &n2);
};

void swapNumbers(Number &n1, Number &n2) {
    int temp = n1.value;
    n1.value = n2.value;
    n2.value = temp;
}
int main() {
    Number num1, num2;
    num1.accept();
    num2.accept();
    swapNumbers(num1, num2);
    num1.display();
    num2.display();
}
```

Or

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

class Student {
    string name;
    int mark1, mark2, mark3;
public:
    Student(string n, int m1, int m2, int m3) {
        name = n;
        mark1 = m1;
        mark2 = m2;
        mark3 = m3;
    }
    friend void calculateAverage(Student);
};
void calculateAverage(Student s) {
    float avg = (s.mark1 + s.mark2 + s.mark3) / 3;
    cout << "Student name: " << s.name << endl;
    cout << "Average marks: " << avg << endl;
}
int main() {
    Student s1("Mike", 90, 91, 92);
    calculateAverage(s1);
    return 0;
}
```

Q3

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

class Cube{
    double volume;
public:
    void input(){
        double length, width, height;
        cout << "Enter length, width, height of a cube: ";
        cin >> length >> width >> height;
        volume = length * width * height;
    }
    friend void FindGreater(Box b, Cube c);
};
void FindGreater(Box b, Cube c);

class Box{
    double volume;
public:
    void input(){
        double length, width, height;
        cout << "Enter length, width, height of a cube: ";
        cin >> length >> width >> height;
        volume = length * width * height;
    }
    friend void FindGreater(Box b, Cube c);
};
```

```
void FindGreater(Box b, Cube c){
    b.volume > c.volume ? cout << "Box volume is Greater" :
    cout << "Cube volume is greater";
    cout << "box" << b.volume << "/t" << "cube =" <<
    c.volume << endl;
}

int main(){
    Box b;
    Cube c;
    b.input();
    c.input();
    FindGreater(b, c);
};
```

When we have more than one constructor in same class then it is called as constructor overloading.

1) Write the CPP code to create the class rectangle having data member as L & B.

Calculate the area of rectangle for 3 objects. Use constructor overloading to initialize that object.

#include <iostream>

int main()

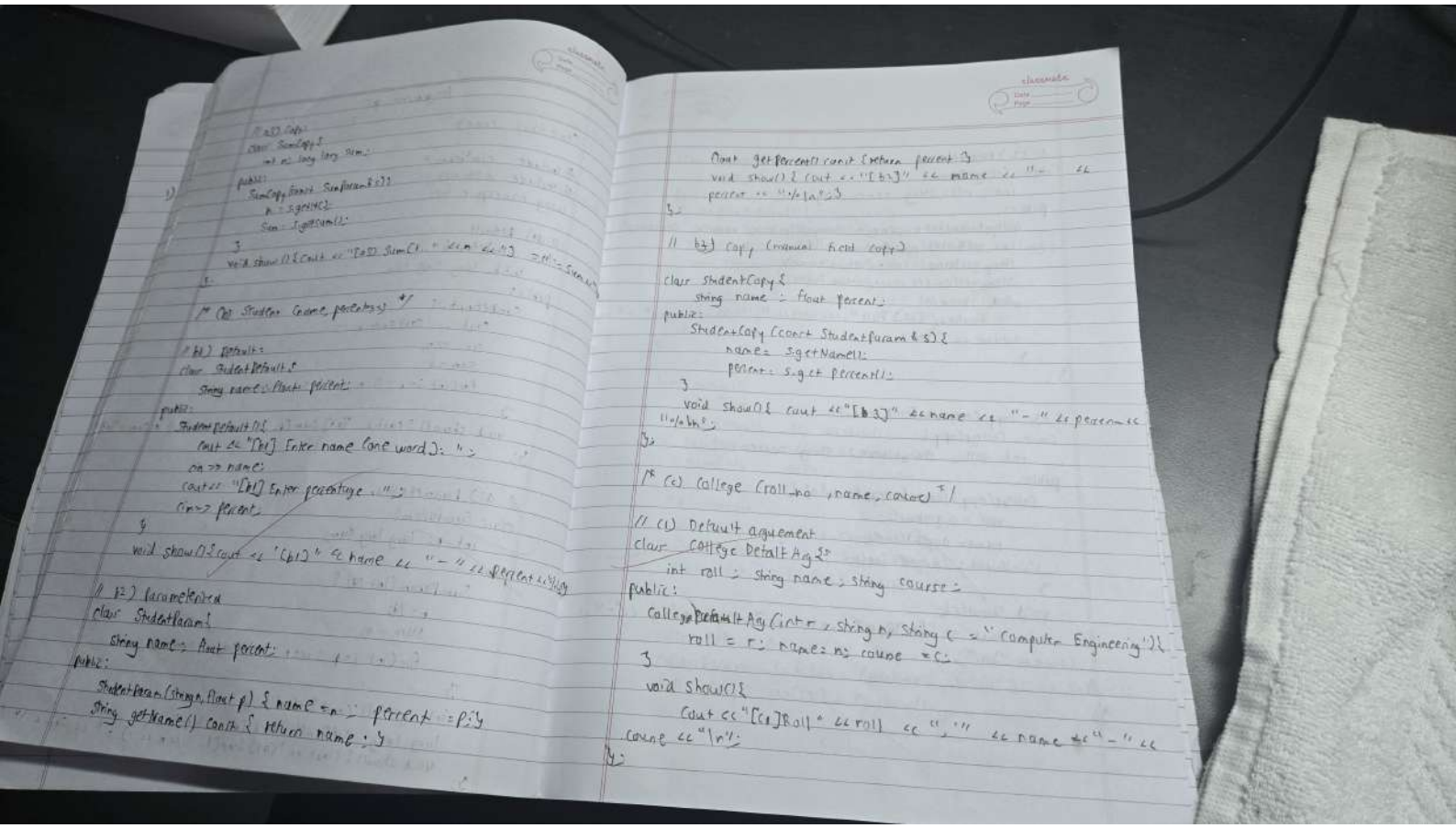
Assignment #5

Cone shot code)

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

// a1) Default:
class SumDefault{
    int n; long long sum;
public:
    SumDefault(){
        cout << "[a1] Enter n: ";
        cin >> n;
        sum = 0;
        for(int i=1; i<=n; ++i) sum = sum+i;
    }
    void show(){ cout << "[a1] Sum(1.." << n << ") = " << sum << "\n"; }
};

// a2) Parameterized
class SumParam{
    int n; long long sum;
public:
    SumParam(int N){
        n = N;
        sum = 0;
        for(int i=1; i<=n; ++i) sum = sum+i;
    }
    int getN() const{ return n; }



// a3) Copy:
class SumCopy {
int n; long long sum;
public:
SumCopy(const SumParam& s) {
n = s.getN();
sum = s.getSum();
}
void show() { cout << "[a3] SumCopy n=" << n << " sum=" << sum << endl; }
};

/* (b) Student (name, percentage) */

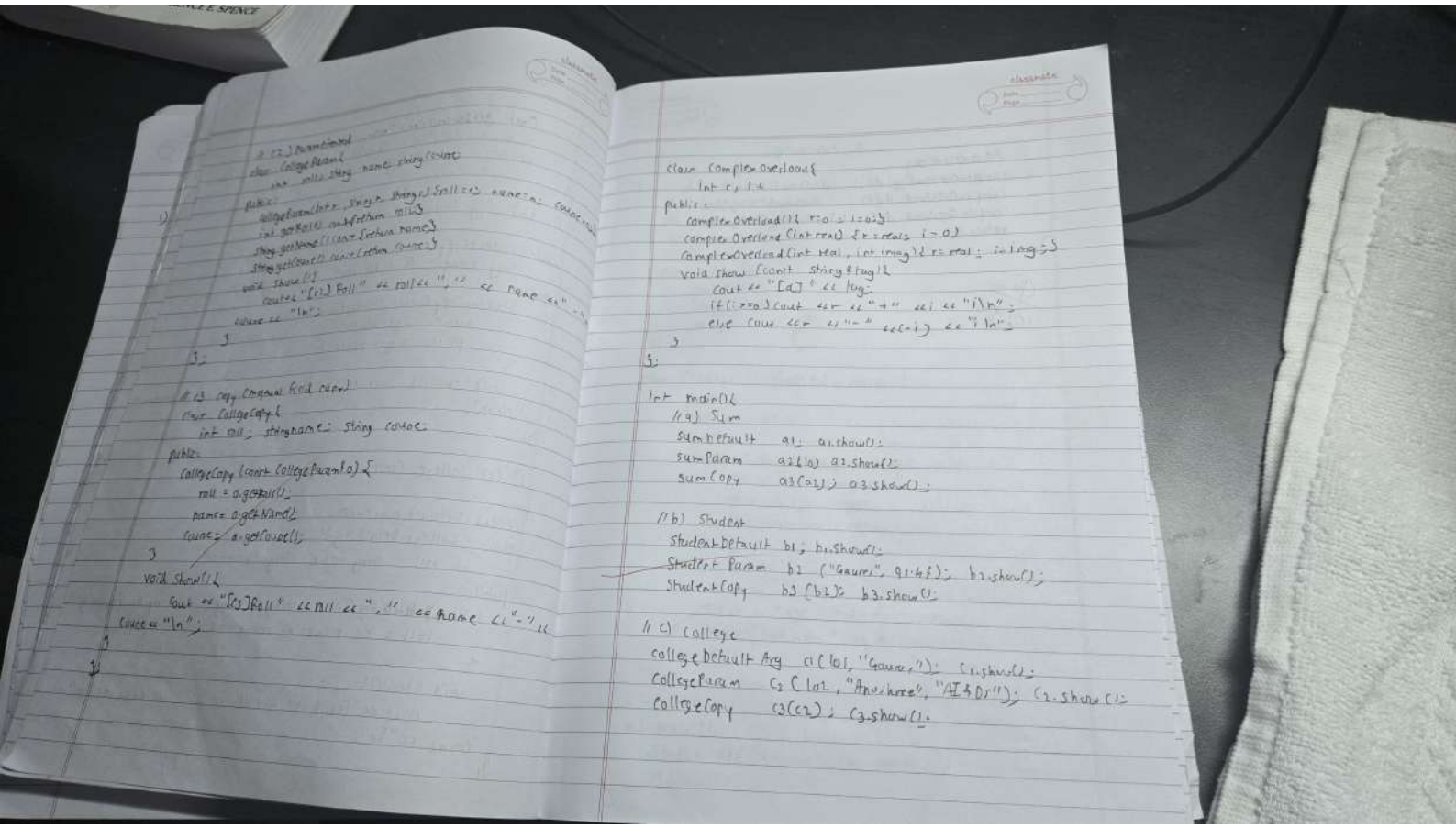
// b1) Default:
class StudentDefault {
string name; float percent;
public:
StudentDefault() {
cout << "[b1] Enter name (one word): ";
cin >> name;
cout << "[b1] Enter percentage: ";
cin >> percent;
}
void show() { cout << "[b1] " << name << " - " << percent << "%\n"; }
};

// b2) Parameterized
class StudentParam {
string name; float percent;
public:
StudentParam(string n, float p) { name = n; percent = p; }
string getName() const { return name; }
float getPercent() const { return percent; }
void show() { cout << "[b2] " << name << " - " << percent << "%\n"; }
};

// b3) Copy (Canonical Form Copy)
class StudentCopy {
string name; float percent;
public:
StudentCopy(const StudentParam& s) {
name = s.getName();
percent = s.getPercent();
}
void show() { cout << "[b3] " << name << " - " << percent << "%\n"; }
};

/* (c) College (roll_no, name, course) */

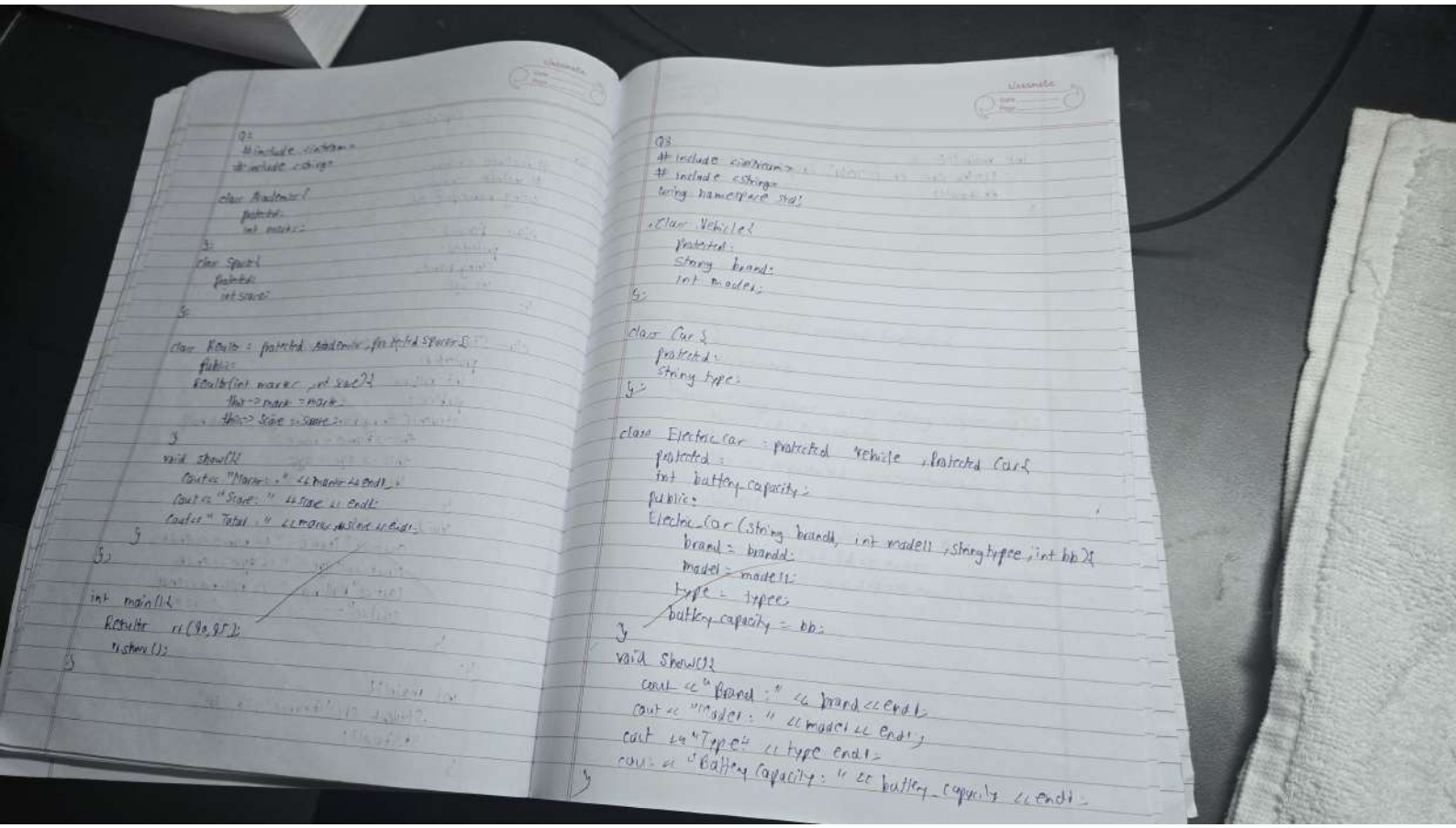
// c1) Default argument
class CollegeDefaultArg {
int roll; string name, course;
public:
CollegeDefaultArg(int r, string n, string c = "Computer Engineering") {
roll = r; name = n; course = c;
}
void show() {
cout << "[c1] Roll " << roll << ", " << name << " - " << course << "\n";
}
};



Experiment - 6

Q1.

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

class Person{
    protected:
    string name;
    int age;
};

class Student : protected Person{
    protected:
    int rollno;
    public:
    Student(string name, int age, int rollno){
        this->name = name;
        this->age = age;
        this->rollno = rollno;
    }
    void show(){
        cout<<"Name: "<<name<<endl;
        cout<<"Age: "<<age<<endl;
        cout<<"Roll no: "<<rollno<<endl;
        cout<<"-----------"<<endl;
    }
};
int main(){
    Student st("Gaurav", 20, 101);
    st.show();
}
```



Q2
#include <iostream>
#include <string>

class Academic{
protected:
int marks;
};
class Sports{
protected:
int score;
};

class Result : protected Academic, protected Sports{
public:
Result(int marks, int score){
this->marks = marks;
this->score = score;
}
void show(){
cout<< "Marks: " << marks << endl;
cout<< "Score: " << score << endl;
cout<< "Total: " << marks+score << endl;
}
};

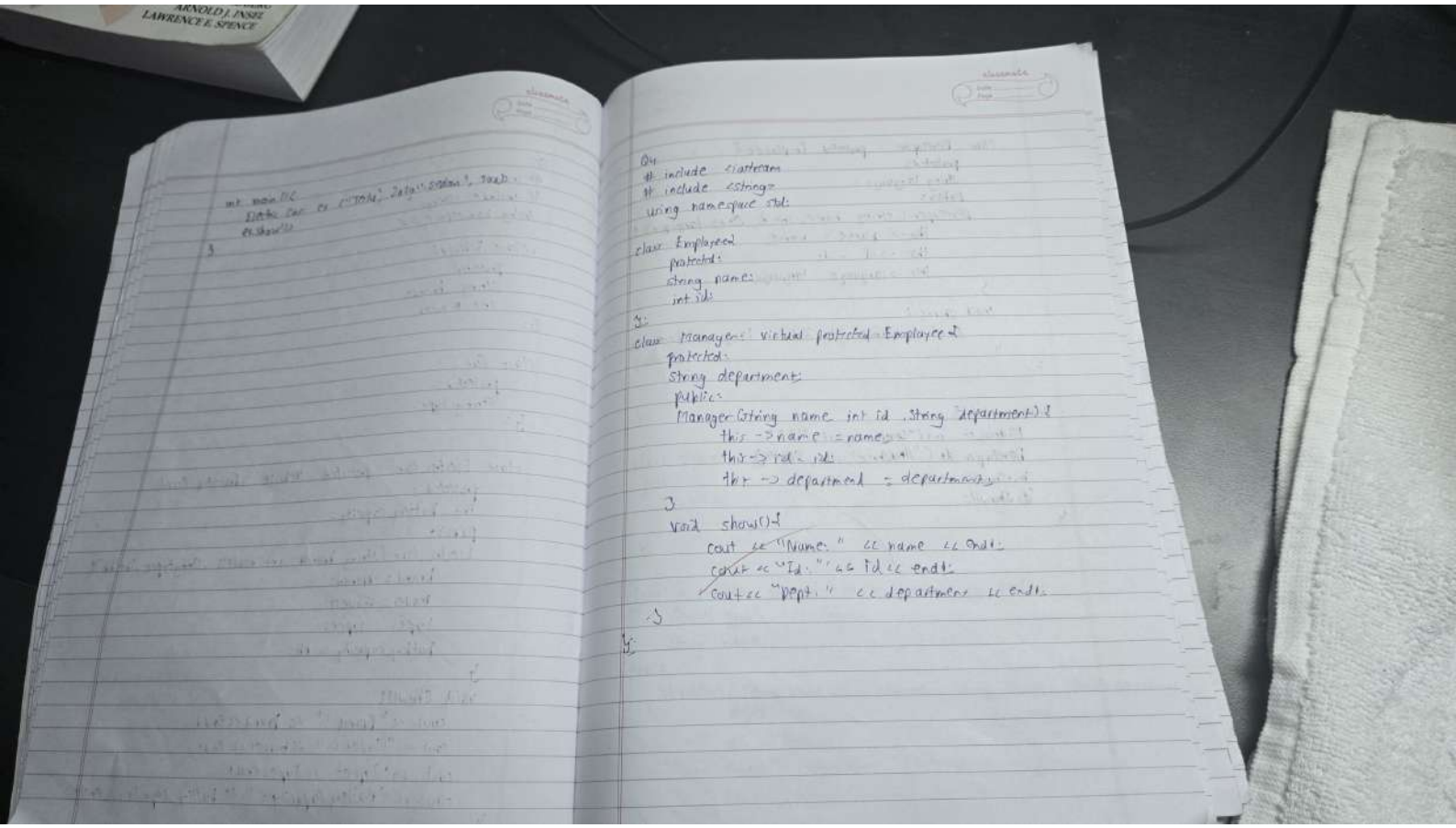
int main(){
Result r(80, 95);
r.show();
}

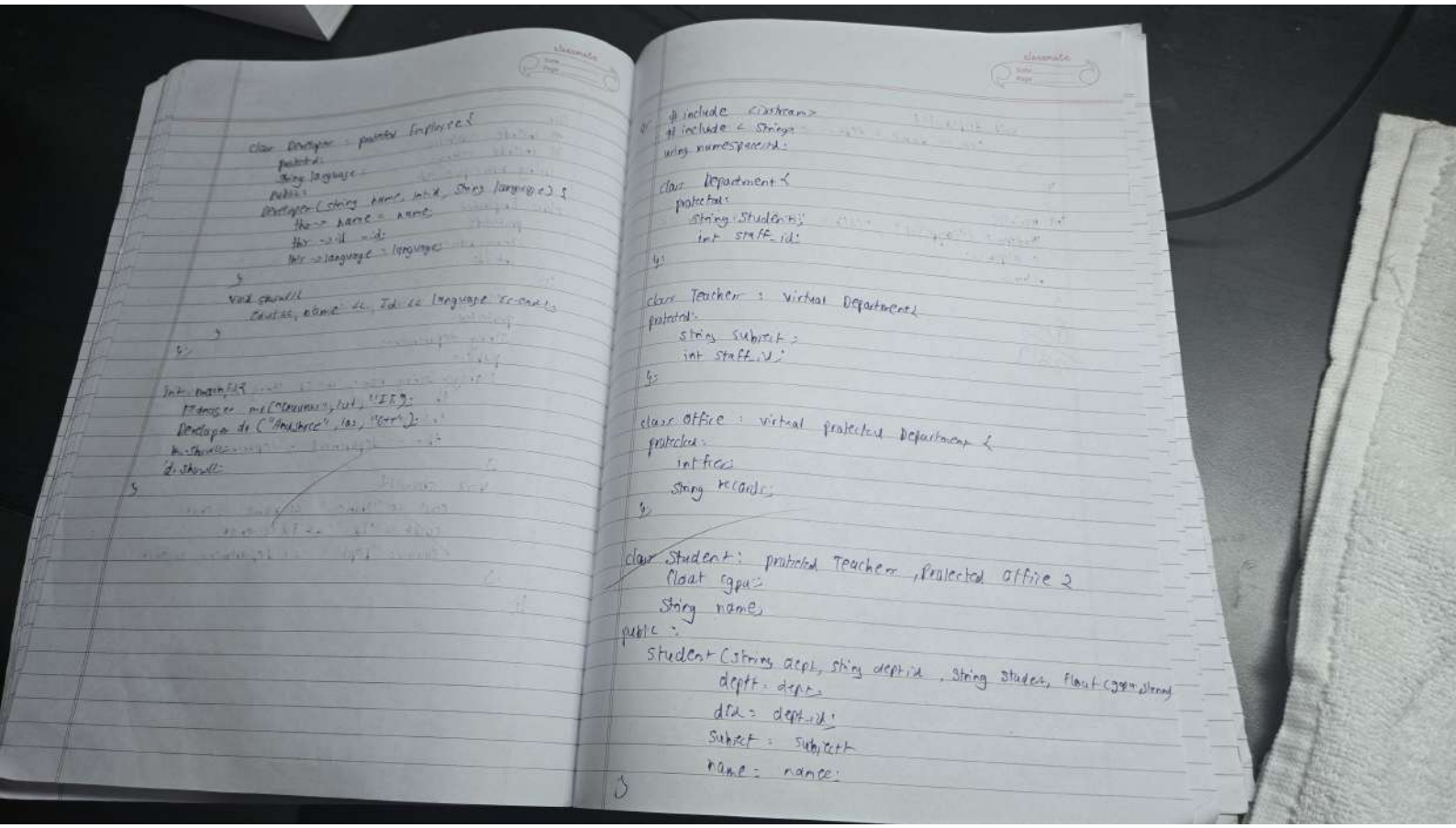
Q3
#include <iostream>
#include <string>
using namespace std;

class Vehicle{
protected:
string brand;
int model;
};

class Car{
protected:
string type;
};

class Electric_car : protected Vehicle, protected Car{
protected:
int battery_capacity;
public:
Electric_Car(string brand, int model, string type, int bb){
brand = brand;
model = model;
type = type;
battery_capacity = bb;
}
void show(){
cout << "Brand: " << brand << endl;
cout << "Model: " << model << endl;
cout << "Type: " << type << endl;
cout << "Battery capacity: " << battery_capacity << endl;
}


ARNOLD I. INSEL
LAWRENCE E. SPENCE
int main()
3
Q4
#include <iostream>
#include <string>
using namespace std;
class Employee
protected:
string name;
int id;
};
class Manager: virtual protected Employee {
protected:
string department;
public:
Manager(string name, int id, string department) {
this->name = name;
this->id = id;
this->department = department;
}
void show() {
cout << "Name: " << name << endl;
cout << "Id: " << id << endl;
cout << "Dept: " << department << endl;
}
};



class Developer : protected Employee {
protected:
string language;
public:
Developer(string name, int id, string language) {
this->name = name;
this->id = id;
this->language = language;
}
void showAll()
cout<<< name << id << language << endl;
}
};

int main() {
Manager m("Ramesh", 101, "IT");
Developer d("Anushka", 102, "C++");
m.showAll();
d.showAll();
}

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

class Department {
protected:
string studentS;
int staff_id;
};

class Teacher : virtual Department {
protected:
string subject;
int staff_id;
};

class office : virtual protected Department {
protected:
int fno;
string records;
};

class Student: protected Teacher, protected office {
float cgpa;
string name;
public:
Student(string dept, string deptid, string studes, float cgpa, string
deptt = dept;
did = deptid;
subject = subject
name = name;
}

Experiment - 7

Q1

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

class Area {
  public:
    float calculate (float length, float breadth){
      return length * breadth;
    }
    float calculate (float side){
      return side* side;
    }
};
int main (){
  Area a;
  cout<<a.calculate (10, 6) << endl;
  cout<<a.calculate (0.5, 0.5) << endl;
```

Q2

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

class Sum{
  public:
    int total(int a [], int n){
      for(int i=0; i<n; i++){
        s+= a[i];
      }
      return s;
    }
```

```
float total(float a[], int b[])
    float sum;
    for(int i=0;i<5;i++)
        sum+=a[i]*b[i];
    return sum;
}
int main(){
    cout << "Sum of int" << total({10,20,30,40,50}, {5}) << endl;
    cout << "Sum of float" << total({1.5,2.5,3.5,4.5,5.5}, {4}) << endl;
    return 0;
}
```

```
#include <iostream>
using namespace std;
class Teacher {
    int experience;
public:
    Teacher(int e) {
        experience = e;
    }
    void display() {
        cout << "Experience: " << experience << endl;
    }
    void operator-() {
        experience = -experience;
    }
};
```

```
int main(){
    Teacher t1(10);
    t1.display();
    -t1;
    t1.display();
}
```

```
#include <iostream>
using namespace std;
class Student{
    int count;
public:
    Student(){
        count = 0;
    }
    void operator++(){
        ++count;
    }
    void operator++(int){
        count++;
    }
    void display(){
        cout << "Student count: " << count << endl;
    }
};
int main(){
    Student s1, s2;
    s1.display();
    ++s1; s1.display();
    s1++;
    s1.display();
}
```

Experiment-2

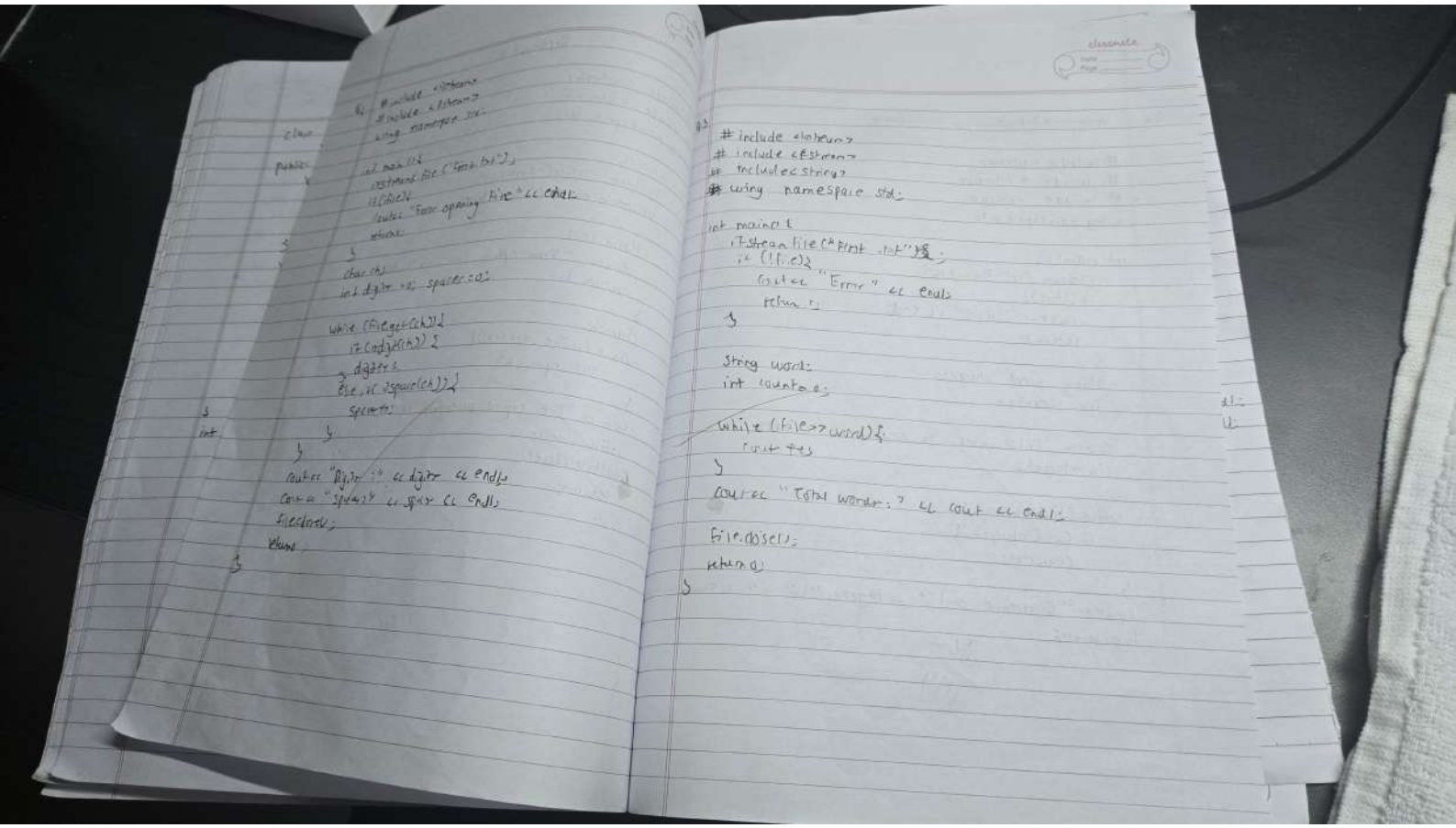
```
#include <iostream>
#include <string>
using namespace std;

class Combine{
    string str
public:
    Combine(string
    string str
public:
    Combine(string s=""){
        str=s;
    }
    Combine operator+(Combine &obj){
        return Combine(str+obj.str);
    }
    void display(){
        cout<<str<<endl;
    }
};
int main(){
    Combine s1("Hi"), s2("par"), s3;
    s3=s1+s2;
    cout<<"Concatenated String:";
    s3.display();
}
```

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

class Login{
protected:
    string name, password;
public:
    virtual void accept(){
        cout<<"Enter name:";
        cin>>name;
        cout<<"Enter password:";
        cin>>password;
    }
    virtual void display(){
        cout<<"Name:"<<Name<<name<<endl;
        cout<<"Password"<<password<<endl;
    }
};
class Email_login: public Login{
    String Email;
public:
    void accept() override{
        cout<<"Email Email ID :";
        cin>>email;
        Login::accept();
    }
    void display() override{
        cout<<"\n---Email login Details---"<<endl;
        cout<<"Email ID"<<email<<endl;
        Login::display();
    }
};
```

```
class Membership login : public Login{
    string membership;
public:
    void accept(){ cout<<"Enter Membership ID:";
    cin>>membership;
    login::accept();
    }
    void display(){ cout<<"In ---- Membership login Details ---- "<<endl;
    cout<<"Membership ID: "<<membership<<endl;
    login::display();
    }
};
int main(){
    Login *login;
    Email login;
    Membership login;
    login->accept();
    login->display();
    login=&m;
    login->accept();
    login->display();
    return 0;
}
```

Experiment 4

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

int main(){
    ifstream inFile("First.txt");
    ofstream outFile("Second.txt");

    if(!inFile){
        cout<<"Error opening First.txt"<<endl;
        return 1;
    }
    char ch;
    while(inFile.get(ch)){
        outFile.put(ch);
    }
    cout<<"File copied successfully"<<endl;
    inFile.close();
    outFile.close();
    return 0;
}
```



#include <iostream>
#include <fstream>
#include <string>
using namespace std;

int main() {
    ifstream file("First.txt");
    if (!file) {
        cout << "Error" << endl;
        return 1;
    }

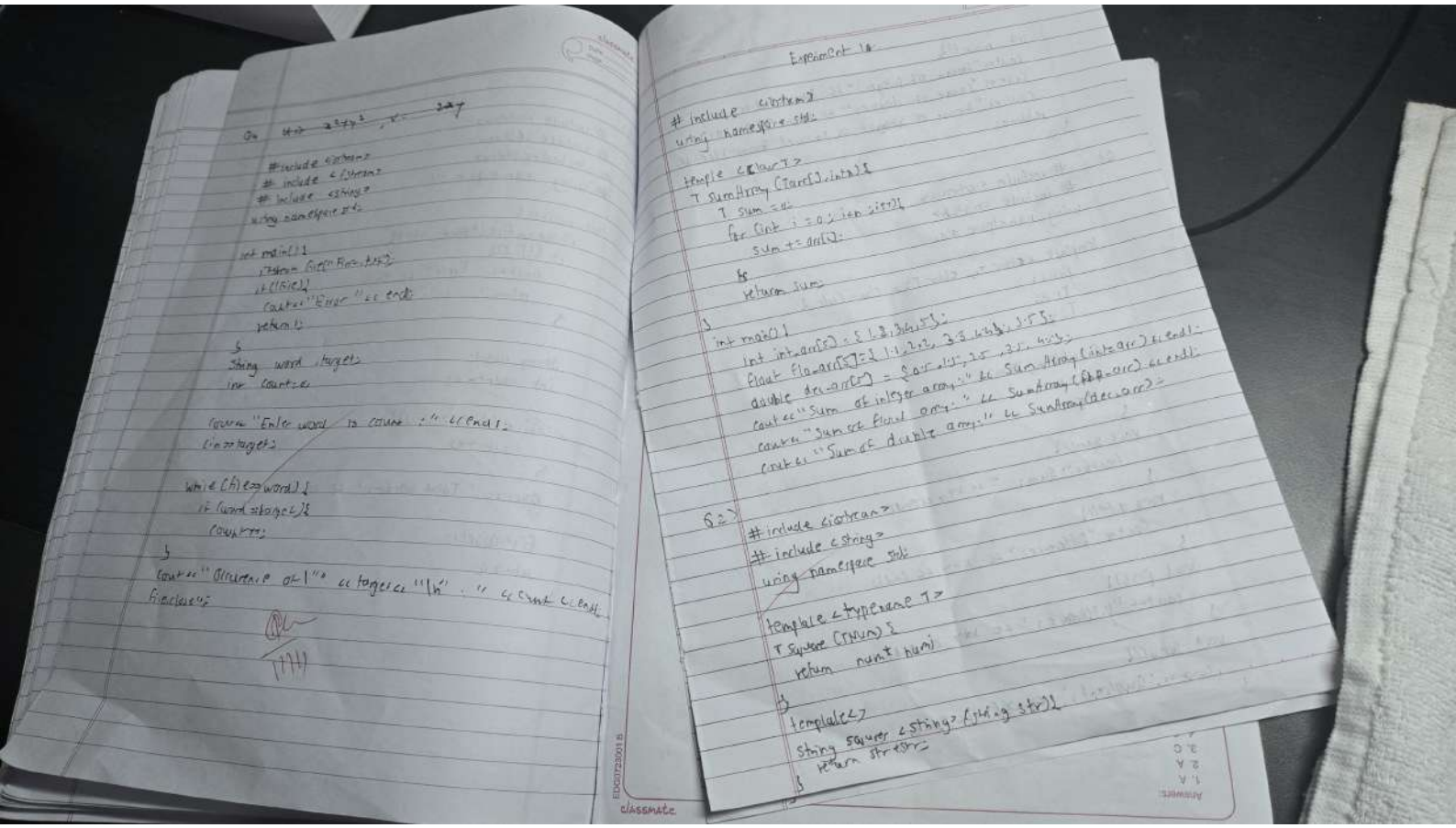
    string word;
    int count = 0;

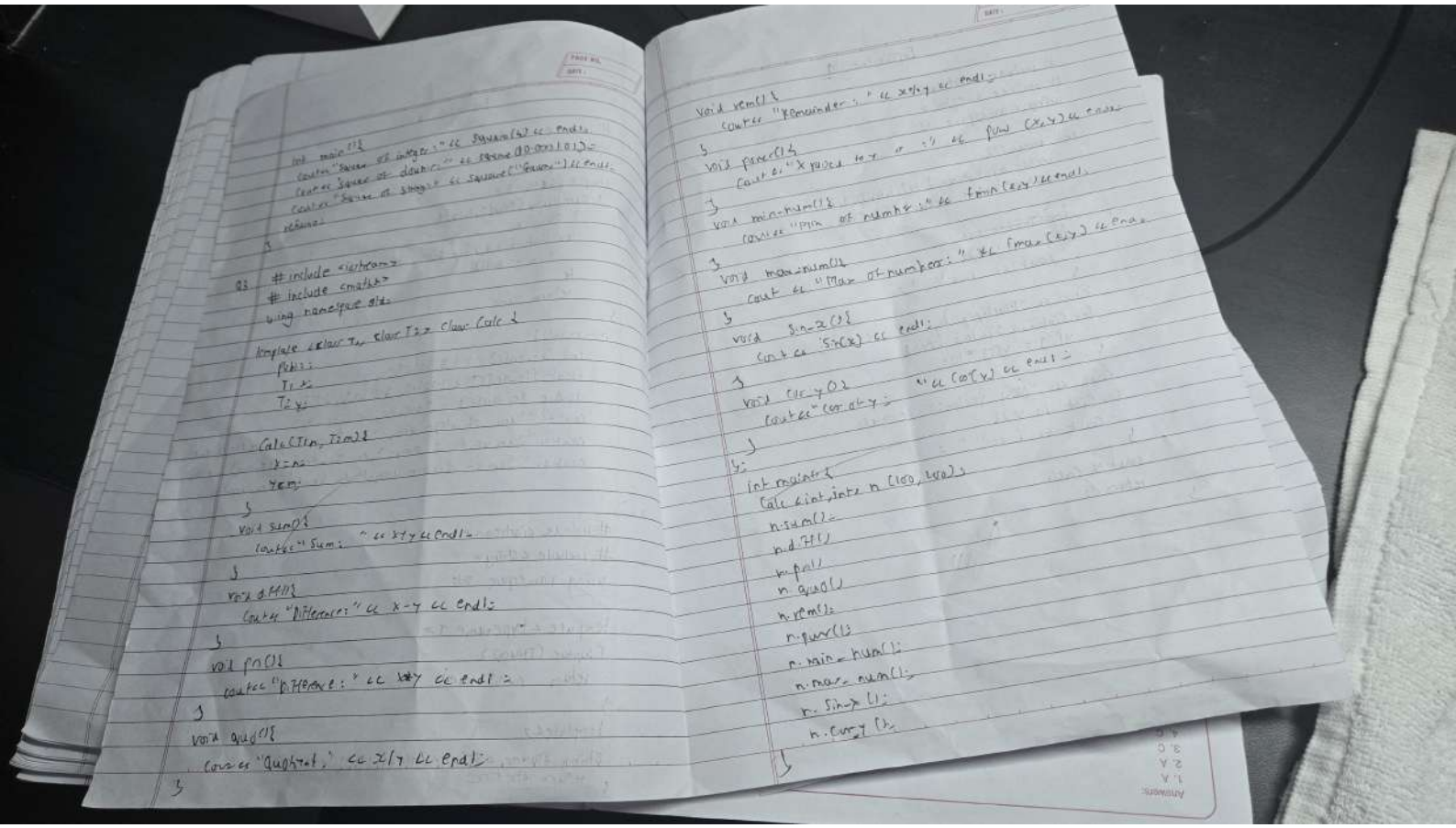
    while (file >> word) {
        count++;
    }

    cout << "Total words: " << count << endl;

    file.close();

    return 0;
}





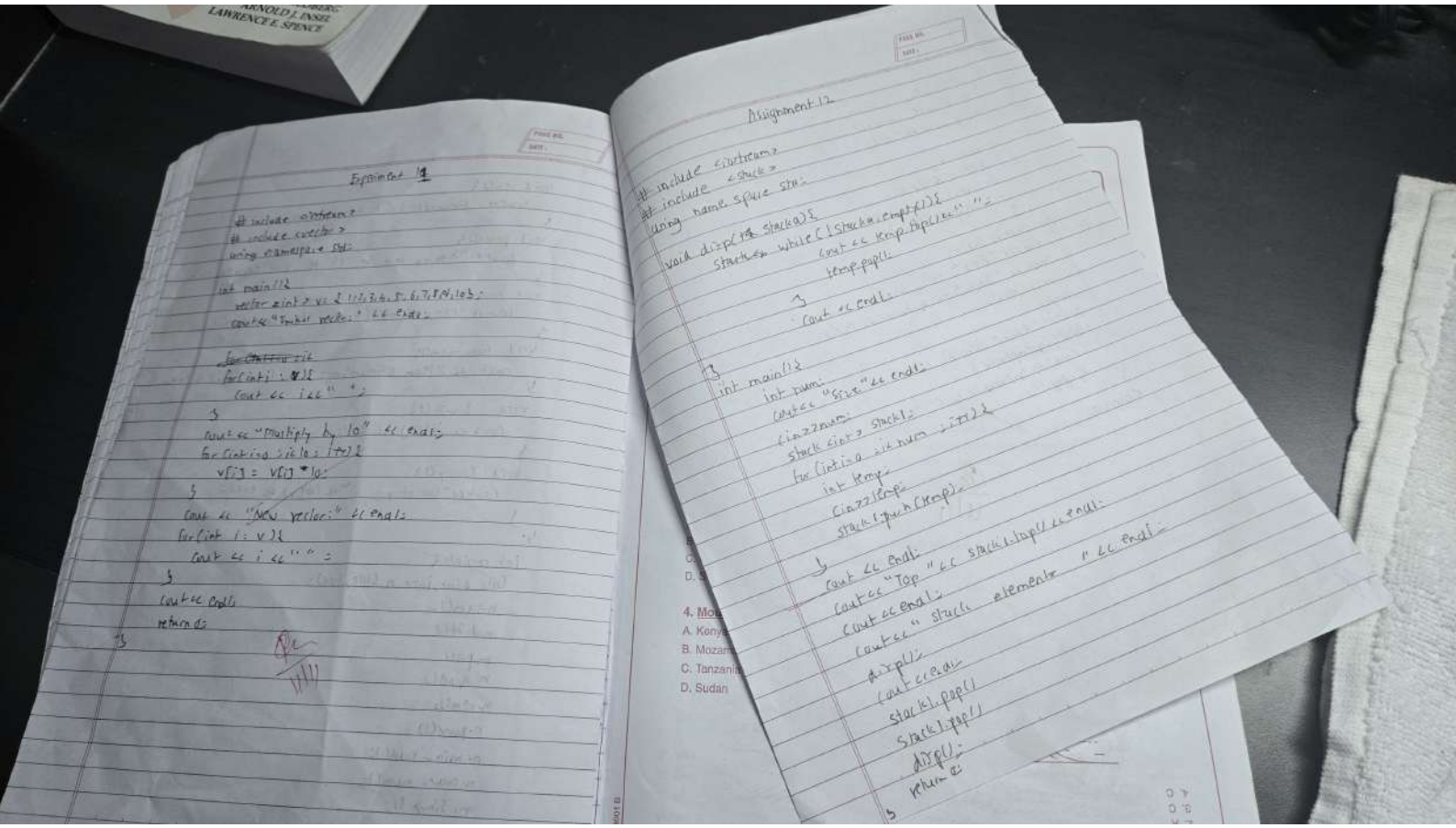
int main(){
cout<<"Square of integer:"<<square(5)<<endl;
cout<<"Square of double:"<<square(10.5)<<endl;
cout<<"Square of string:"<<square("Square")<<endl;
return 0;
}

Q3. #include <iostream>
#include <math.h>
using namespace std;

template <class T1, class T2> class Calc {
public:
T1 x;
T2 y;

Calc(T1, T2){
x=x;
y=y;
}
void sum(){
cout<<"Sum: "<<x+y<<endl;
}
void diff(){
cout<<"Difference:"<<x-y<<endl;
}
void pro(){
cout<<"Product:"<<x*y<<endl;
}
void quo(){
cout<<"Quotient:"<<x/y<<endl;
}

void rem(){
cout<<"Remainder:"<<x%y<<endl;
}
void pow(){
cout<<x<<"raised to"<<y<<":"<<pow(x,y)<<endl;
}
void min_num(){
cout<<"Min of numbers:"<<fmin(x,y)<<endl;
}
void max_num(){
cout<<"Max of numbers:"<<fmax(x,y)<<endl;
}
void sin_x(){
cout<<sin(x)<<endl;
}
void cos_y(){
cout<<"cos of y:"<<cos(y)<<endl;
}
};
int main(){
Calc<int,int> n(100,500);
n.sum();
n.diff();
n.pro();
n.quo();
n.rem();
n.pow();
n.min_num();
n.max_num();
n.sin_x();
n.cos_y();
}


ARNOLD J. ENSEL
LAWRENCE E. SPENCE
Experiment 11
Assignment 12
4. Mo
A. Kenya
B. Mozam
C. Tanzania
D. Sudan

STEPHEN H. FRIEDBERG
ARNOLD J. INSEL
LAWRENCE E. SPENCE

PAGE NO.
DATE :

2) #include <iostream>
#include <queue>
using namespace std;

int main() {
queue <int> q;
for (int i=0 ; i<=10 ; i++) {
q.push (i*10);
}
cout<<endl;
cout<<"Front" << q.front() << endl;
cout << "Back" << q.back() << endl;
cout<<endl;
q.pop();
cout<<"queue elements:" ;
while(!q.empty()) {
cout<<q.front() <<" ";
q.pop();
}
return 0;

11/11