

Slip 1 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Student {
4.     int roll_no;
5.     float percentage;
6. public:
7.     void input() {
8.         cout << "Enter roll number: ";
9.         cin >> roll_no;
10.        cout << "Enter percentage: ";
11.        cin >> percentage;
12.    }
13.    float getPercentage() {
14.        return percentage;
15.    }
16.    int getRollNo() {
17.        return roll_no;
18.    }
19. };
20. int main() {
21.     Student s1, s2;
22.     s1.input();
23.     s2.input();
24.     if (s1.getPercentage() > s2.getPercentage())
25.         cout << "Highest percentage roll no: " << s1.getRollNo();
26.     else
27.         cout << "Highest percentage roll no: " << s2.getRollNo();
28. }
29.
```

Slip 1 Q2.cpp

```
1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4. int main() {
5.     int n;
6.     cout << "Enter size of array: ";
7.     cin >> n;
8.     int a[n];
9.     for (int i = 0; i < n; i++) {
10.         cout << "Enter element " << i + 1 << ": ";
11.         cin >> a[i];
12.     }
13.     for (int i = 0; i < n; i++) {
14.         try {
15.             if (a[i] < 0)
16.                 throw a[i];
17.             cout << "Square root of " << a[i] << " = " << sqrt(a[i]) << endl;
18.         } catch (int x) {
19.             cout << "Exception: negative number " << x << endl;
20.         }
21.     }
22. }
23.
```

Slip 3 Q1.cpp

```
1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int main() {
5.     ifstream fin("Book.txt");
6.     string line;
7.     if (!fin) {
8.         cout << "File not found";
9.         return 0;
10.    }
```

```
11.     while (getline(fin, line)) {
12.         cout << line << endl;
13.     }
14.     fin.close();
15. }
16.
```

Slip 3 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Point {
4.     int x, y;
5. public:
6.     Point() {
7.         x = 0; y = 0;
8.     }
9.     Point(int a, int b) {
10.        x = a; y = b;
11.    }
12.     Point(const Point &p) {
13.        x = p.x; y = p.y;
14.    }
15.     void display() {
16.         cout << "(" << x << "," << y << ")" << endl;
17.     }
18. };
19. int main() {
20.     Point p1, p2(5, 10), p3(p2);
21.     cout << "p1 = "; p1.display();
22.     cout << "p2 = "; p2.display();
23.     cout << "p3 = "; p3.display();
24. }
25.
```

Slip 4 Q1.cpp

```
1. #include <iostream>
2. #include <unordered_map>
3. using namespace std;
4. int main() {
5.     unordered_map<int, string> m;
6.     int roll;
7.     string name;
8.     for (int i = 0; i < 3; i++) {
9.         cout << "Enter roll no: ";
10.        cin >> roll;
11.        cout << "Enter name: ";
12.        cin >> name;
13.        m[roll] = name;
14.    }
15.    cout << "\nStudent Data:\n";
16.    for (auto &p : m) {
17.        cout << "Roll No: " << p.first << " Name: " << p.second << endl;
18.    }
19. }
20.
```

Slip 4 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Person {
4.     int pid;
5.     string name, nat, gen, dob, doi, doe;
6. public:
7.     void input() {
8.         cout << "Enter Passport Id: ";
9.         cin >> pid;
```

```

10.     cout << "Enter Name: ";
11.     cin >> name;
12.     cout << "Enter Nationality: ";
13.     cin >> nat;
14.     cout << "Enter Gender: ";
15.     cin >> gen;
16.     cout << "Enter Date of Birth: ";
17.     cin >> dob;
18.     cout << "Enter Date of Issue: ";
19.     cin >> doi;
20.     cout << "Enter Date of Expiry: ";
21.     cin >> doe;
22. }
23. void display() {
24.     cout << pid << " " << name << " " << nat << " " << gen << " " << dob << " " << doi << " " <<
25.     doe << endl;
26. }
27. void display(int id) {
28.     if (pid == id) display();
29. }
30. int main() {
31.     int n;
32.     cout << "Enter number of persons: ";
33.     cin >> n;
34.     Person p[50];
35.     for (int i = 0; i < n; i++) p[i].input();
36.     int ch;
37.     do {
38.         cout << "\n1. Display one person\n2. Display all persons\n3. Exit\nEnter choice: ";
39.         cin >> ch;
40.         if (ch == 1) {
41.             int id;
42.             cout << "Enter Passport Id: ";
43.             cin >> id;
44.             for (int i = 0; i < n; i++) p[i].display(id);
45.         } else if (ch == 2) {
46.             for (int i = 0; i < n; i++) p[i].display();
47.         }
48.     } while (ch != 3);
49. }
50.

```

Slip 5 Q1.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Cube {
4.     double l, w, d;
5. public:
6.     void input() {
7.         cout << "Enter length: ";
8.         cin >> l;
9.         cout << "Enter width: ";
10.        cin >> w;
11.        cout << "Enter depth: ";
12.        cin >> d;
13.    }
14.    double volume() {
15.        return l * w * d;
16.    }
17. };
18. int main() {
19.     Cube c;
20.     c.input();
21.     cout << "Volume = " << c.volume();
22. }
23.

```

Slip 5 Q2.cpp

```

1. #include <iostream>
2. using namespace std;
3. int main() {
4.     double a, b;
5.     char op;
6.     int ch;
7.     do {
8.         cout << "\nMenu:\n1. Calculate\n2. Exit\nEnter choice: ";
9.         cin >> ch;
10.        if (ch == 1) {
11.            cout << "Enter 2 numbers: ";
12.            cin >> a >> b;
13.            cout << "Enter operator (+,-,*,/): ";
14.            cin >> op;
15.            try {
16.                switch(op) {
17.                    case '+': cout << "Result = " << a + b; break;
18.                    case '-': cout << "Result = " << a - b; break;
19.                    case '*': cout << "Result = " << a * b; break;
20.                    case '/':
21.                        if (b == 0) throw "Division by zero";
22.                        cout << "Result = " << a / b; break;
23.                    default: throw "Invalid operator";
24.                }
25.            } catch (const char* msg) {
26.                cout << "Exception: " << msg;
27.            }
28.        }
29.    } while(ch != 2);
30. }
31.

```

Slip 6 Q1.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Interest {
4.     double principal, rate, time;
5. public:
6.     Interest(double p, double t, double r = 5) {
7.         principal = p;
8.         time = t;
9.         rate = r;
10.    }
11.    double simpleInterest() {
12.        return (principal * rate * time) / 100;
13.    }
14. };
15. int main() {
16.     double p, t, r;
17.     cout << "Enter principal: ";
18.     cin >> p;
19.     cout << "Enter time (years): ";
20.     cin >> t;
21.     cout << "Enter rate (default 5%): ";
22.     cin >> r;
23.     Interest in(p, t, r);
24.     cout << "Simple Interest = " << in.simpleInterest();
25. }
26.

```

Slip 6 Q2.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Date {
4.     int dd, mm, yyyy;
5. public:
6.     friend istream& operator>>(istream &in, Date &d) {
7.         cout << "Enter day: "; in >> d.dd;
8.         cout << "Enter month: "; in >> d.mm;

```

```

9.         cout << "Enter year: "; in >> d.yyyy;
10.        return in;
11.    }
12.    friend ostream& operator<<(ostream &out, const Date &d) {
13.        out << d.dd << "/" << d.mm << "/" << d.yyyy;
14.        return out;
15.    }
16.};
17. int main() {
18.     Date d;
19.     cin >> d;
20.     cout << "Date = " << d;
21. }
22.

```

Slip 7 Q1.cpp

```

1. #include <iostream>
2. #include <iomanip>
3. using namespace std;
4. class Product {
5. protected:
6.     int id;
7.     string name;
8.     double price;
9. public:
10.    void input() {
11.        cout << "Enter Product Id: ";
12.        cin >> id;
13.        cout << "Enter Product Name: ";
14.        cin >> name;
15.        cout << "Enter Price: ";
16.        cin >> price;
17.    }
18.    double getPrice() {
19.        return price;
20.    }
21.    string getName() {
22.        return name;
23.    }
24. };
25. class Discount : public Product {
26.     double discount;
27. public:
28.    void inputDiscount() {
29.        input();
30.        cout << "Enter Discount (%): ";
31.        cin >> discount;
32.    }
33.    double discountedPrice() {
34.        return price - (price * discount / 100);
35.    }
36. };
37. int main() {
38.     int n;
39.     cout << "Enter number of products: ";
40.     cin >> n;
41.     Discount p[50];
42.     double total = 0, totalDisc = 0;
43.     for (int i = 0; i < n; i++) {
44.         p[i].inputDiscount();
45.         total += p[i].getPrice();
46.         totalDisc += (p[i].getPrice() - p[i].discountedPrice());
47.     }
48.     cout << "\nBill:\n";
49.     cout << setw(10) << "Product" << setw(15) << "Price" << setw(15) << "Discounted Price" << endl;
50.     for (int i = 0; i < n; i++) {
51.         cout << setw(10) << p[i].getName()
52.             << setw(15) << p[i].getPrice()
53.             << setw(15) << p[i].discountedPrice() << endl;
54.     }
55.     cout << "\nTotal Price = " << total;
56.     cout << "\nTotal Discount = " << totalDisc;

```

```
57. }
58.
```

Slip 7 Q2.cpp

```
1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int main() {
5.     ifstream fin("file1.txt");
6.     ofstream fout("file2.txt", ios::app);
7.     if (!fin || !fout) {
8.         cout << "File error";
9.         return 0;
10.    }
11.    string line;
12.    while (getline(fin, line)) {
13.        fout << line << endl;
14.    }
15.    fin.close();
16.    fout.close();
17.    cout << "Contents appended successfully";
18. }
19.
```

Slip 10 Q1.cpp

```
1. #include <iostream>
2. #include <iomanip>
3. using namespace std;
4. class Mobile {
5.     int id;
6.     string name;
7.     double price;
8. public:
9.     Mobile(int i, string n, double p) {
10.         id = i;
11.         name = n;
12.         price = p;
13.     }
14.     void display() {
15.         cout << "Mobile Id: " << id << " Name: " << name
16.             << " Price: " << setw(8) << fixed << setprecision(2) << price << endl;
17.     }
18. };
19. int main() {
20.     Mobile m(100, "One Plus", 21000.99);
21.     m.display();
22. }
23.
```

Slip 10 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Shape {
4. public:
5.     virtual double area() = 0;
6. };
7. class Circle : public Shape {
8.     double r;
9. public:
10.     Circle(double rad) { r = rad; }
11.     double area() { return 3.1416 * r * r; }
12. };
13. class Rectangle : public Shape {
14.     double l, b;
15. public:
16.     Rectangle(double length, double breadth) { l = length; b = breadth; }
```

```

17.     double area() { return l * b; }
18. };
19. class Triangle : public Shape {
20.     double base, height;
21. public:
22.     Triangle(double b, double h) { base = b; height = h; }
23.     double area() { return 0.5 * base * height; }
24. };
25. int main() {
26.     Circle c(5);
27.     Rectangle r(4, 6);
28.     Triangle t(3, 7);
29.     cout << "Circle Area = " << c.area() << endl;
30.     cout << "Rectangle Area = " << r.area() << endl;
31.     cout << "Triangle Area = " << t.area() << endl;
32. }
33.

```

Slip 11 Q1.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Person {
4. protected:
5.     string name;
6.     int age;
7. public:
8.     void inputPerson() {
9.         cout << "Enter name: ";
10.        cin >> name;
11.        cout << "Enter age: ";
12.        cin >> age;
13.    }
14. };
15. class Student : public Person {
16.     int roll_no;
17.     float marks;
18. public:
19.     void inputStudent() {
20.         inputPerson();
21.         cout << "Enter roll number: ";
22.         cin >> roll_no;
23.         cout << "Enter marks: ";
24.         cin >> marks;
25.     }
26.     void display() {
27.         cout << "Name: " << name << " Age: " << age
28.             << " Roll No: " << roll_no << " Marks: " << marks << endl;
29.     }
30. };
31. int main() {
32.     Student s;
33.     s.inputStudent();
34.     s.display();
35. }
36.

```

Slip 11 Q2.cpp

```

1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. class Item {
5.     int no, qty;
6.     string name;
7.     double price;
8. public:
9.     void input() {
10.         cout << "Enter Item No: ";
11.         cin >> no;
12.         cout << "Enter Item Name: ";

```

```

13.     cin >> name;
14.     cout << "Enter Price: ";
15.     cin >> price;
16.     cout << "Enter Quantity: ";
17.     cin >> qty;
18. }
19. void writeToFile(ofstream &fout) {
20.     fout << no << " " << name << " " << price << " " << qty << endl;
21. }
22. };
23. int main() {
24.     int n;
25.     cout << "Enter number of items: ";
26.     cin >> n;
27.     Item items[50];
28.     ofstream fout("Items.txt");
29.     for (int i = 0; i < n; i++) {
30.         items[i].input();
31.         items[i].writeToFile(fout);
32.     }
33.     fout.close();
34.     cout << "Item information written to file successfully.";
35. }
36.

```

Slip 12 Q1.cpp

```

1. #include <iostream>
2. using namespace std;
3. inline double circleArea(double r) {
4.     return 3.14 * r * r;
5. }
6. inline double squareArea(double s) {
7.     return s * s;
8. }
9. int main() {
10.     double r, s;
11.     cout << "Enter radius of circle: ";
12.     cin >> r;
13.     cout << "Enter side of square: ";
14.     cin >> s;
15.     cout << "Area of circle = " << circleArea(r) << endl;
16.     cout << "Area of square = " << squareArea(s);
17. }
18.

```

Slip 12 Q2.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Date {
4.     int dd, mm, yyyy;
5.     string months[12] = {"Jan", "Feb", "Mar", "Apr", "May", "Jun", "Jul", "Aug", "Sep", "Oct", "Nov", "Dec"};
6. public:
7.     Date(int d, int m, int y) {
8.         dd = d;
9.         if(m >= 1 && m <= 12) mm = m;
10.        else {
11.            cout << "Invalid month, setting to 1\n";
12.            mm = 1;
13.        }
14.        yyyy = y;
15.    }
16.    void display() {
17.        cout << dd << "-" << months[mm-1] << "-" << yyyy;
18.    }
19. };
20. int main() {
21.     int d, m, y;
22.     cout << "Enter date (dd mm yyyy): ";
23.     cin >> d >> m >> y;

```

```
24.     Date dt(d, m, y);
25.     cout << "Formatted date: ";
26.     dt.display();
27. }
28.
```

Slip 13 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. double area(double r) {
4.     return 3.14 * r * r;
5. }
6. double area(double r, double h) {
7.     return 3.14 * r * (r + h);
8. }
9. double area(double r, char c) {
10.    return 4 * 3.14 * r * r;
11. }
12. int main() {
13.     double r, h;
14.     char ch = 's';
15.     cout << "Enter radius for circle and sphere: ";
16.     cin >> r;
17.     cout << "Enter height for cone: ";
18.     cin >> h;
19.     cout << "Area of circle = " << area(r) << endl;
20.     cout << "Area of cone = " << area(r, h) << endl;
21.     cout << "Area of sphere = " << area(r, ch);
22. }
23.
```

Slip 13 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. int main() {
4.     string name;
5.     int marks;
6.     cout << "Enter student name: ";
7.     cin >> name;
8.     cout << "Enter marks: ";
9.     cin >> marks;
10.    try {
11.        if(name.empty())
12.            throw "Name cannot be empty";
13.        if(marks < 0 || marks > 100)
14.            throw "Marks should be between 0 and 100";
15.        cout << "Student: " << name << " Marks: " << marks << endl;
16.    } catch(const char* msg) {
17.        cout << "Exception: " << msg;
18.    }
19. }
20.
```

Slip 14 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Sum {
4.     int n, total;
5. public:
6.     Sum(int x) {
7.         n = x;
8.         total = 0;
9.         for(int i = 1; i <= n; i++) total += i;
10.    }
11.    void display() {
12.        cout << "Sum = " << total;
13.    }

```

```
14. };
15. int main() {
16.     int n;
17.     cout << "Enter n: ";
18.     cin >> n;
19.     Sum s(n);
20.     s.display();
21. }
22.
```

Slip 14 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class College {
4.     int id, year;
5.     string name, uni;
6. public:
7.     void input() {
8.         cout << "Enter College Id: ";
9.         cin >> id;
10.        cout << "Enter College Name: ";
11.        cin >> name;
12.        cout << "Enter Establishment Year: ";
13.        cin >> year;
14.        cout << "Enter University Name: ";
15.        cin >> uni;
16.    }
17.    void display() {
18.        cout << id << " " << name << " " << year << " " << uni << endl;
19.    }
20.    void display(string u) {
21.        if(uni == u) display();
22.    }
23.    void display(int y) {
24.        if(year == y) display();
25.    }
26. };
27. int main() {
28.     int n;
29.     cout << "Enter number of colleges: ";
30.     cin >> n;
31.     College c[50];
32.     for(int i=0; i<n; i++) c[i].input();
33.     string u;
34.     cout << "Enter University Name to search: ";
35.     cin >> u;
36.     for(int i=0; i<n; i++) c[i].display(u);
37.     int y;
38.     cout << "Enter Establishment Year to search: ";
39.     cin >> y;
40.     for(int i=0; i<n; i++) c[i].display(y);
41. }
42.
```

Slip 16 Q1.cpp

```
1. #include <iostream>
2. #include <unordered_map>
3. using namespace std;
4. int main() {
5.     string str;
6.     cout << "Enter a string: ";
7.     cin.ignore();
8.     getline(cin, str);
9.     unordered_map<char, int> freq;
10.    for(char c : str)
11.        freq[c]++;
12.    cout << "Character frequencies:\n";
13.    for(auto &p : freq)
14.        cout << p.first << " : " << p.second << endl;
```

```
15. }
16.
```

Slip 16 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Employee {
4.     int E_no;
5.     string E_Name, Designation;
6.     double Salary;
7. public:
8.     void input() {
9.         cout << "Enter Employee No: ";
10.        cin >> E_no;
11.        cout << "Enter Name: ";
12.        cin >> E_Name;
13.        cout << "Enter Designation: ";
14.        cin >> Designation;
15.        cout << "Enter Salary: ";
16.        cin >> Salary;
17.    }
18.    void display() {
19.        cout << "No: " << E_no << " Name: " << E_Name
20.                      << " Designation: " << Designation << " Salary: " << Salary << endl;
21.    }
22.    Employee* maxSalary(Employee *e) {
23.        return (Salary > e->Salary) ? this : e;
24.    }
25. };
26. int main() {
27.     Employee e1, e2;
28.     e1.input();
29.     e2.input();
30.     cout << "Employee with maximum salary:\n";
31.     e1.maxSalary(&e2)->display();
32. }
33.
```

Slip 18 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Product {
4.     int productCode;
5.     string productName;
6.     double weight;
7. public:
8.     void input() {
9.         cout << "Enter Product Code: ";
10.        cin >> productCode;
11.        cout << "Enter Product Name: ";
12.        cin >> productName;
13.        cout << "Enter Weight: ";
14.        cin >> weight;
15.    }
16.    void display() {
17.        cout << "Code: " << productCode << " Name: " << productName << " Weight: " << weight << endl;
18.    }
19.    void checkWeight() {
20.        if(weight > 100) throw "Weight exceeds 100";
21.    }
22. };
23. int main() {
24.     Product p;
25.     p.input();
26.     try {
27.         p.checkWeight();
28.         p.display();
29.     } catch(const char* msg) {
30.         cout << "Exception: " << msg;
31.     }
}
```

```
32. }
33.
```

Slip 18 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Distance {
4.     int feet, inches;
5. public:
6.     void input() {
7.         cout << "Enter feet: ";
8.         cin >> feet;
9.         cout << "Enter inches: ";
10.        cin >> inches;
11.    }
12.    void display() {
13.        cout << feet << " feet " << inches << " inches" << endl;
14.    }
15.    friend Distance add(Distance d1, Distance d2);
16. };
17. Distance add(Distance d1, Distance d2) {
18.     Distance d3;
19.     d3.inches = d1.inches + d2.inches;
20.     d3.feet = d1.feet + d2.feet + d3.inches/12;
21.     d3.inches %= 12;
22.     return d3;
23. }
24. int main() {
25.     Distance d1, d2, d3;
26.     d1.input();
27.     d2.input();
28.     d3 = add(d1, d2);
29.     cout << "Sum of distances: ";
30.     d3.display();
31. }
32.
```

Slip 19 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Integer {
4.     int num;
5. public:
6.     void input() {
7.         cout << "Enter an integer: ";
8.         cin >> num;
9.     }
10.    void display() {
11.        cout << "Value = " << num << endl;
12.    }
13.    Integer operator--() {
14.        --num;
15.        return *this;
16.    }
17.    Integer operator--(int) {
18.        Integer temp = *this;
19.        num--;
20.        return temp;
21.    }
22. };
23. int main() {
24.     Integer n;
25.     n.input();
26.     cout << "Pre-decrement: ";
27.     (--n).display();
28.     cout << "Post-decrement: ";
29.     (n--).display();
30.     cout << "After Post-decrement: ";
31.     n.display();
32.
```

```
32. }
33.
```

Slip 19 Q2.cpp

```
1. #include <iostream>
2. #include <unordered_map>
3. using namespace std;
4. int main() {
5.     unordered_map<string, string> dict;
6.     int ch;
7.     do {
8.         cout << "\n1. Add Entry\n2. Search Capital\n3. Display All\n4. Exit\nEnter choice: ";
9.         cin >> ch;
10.        if(ch==1) {
11.            string country, capital;
12.            cout << "Enter country: ";
13.            cin >> country;
14.            cout << "Enter capital: ";
15.            cin >> capital;
16.            dict[country] = capital;
17.        }
18.        else if(ch==2) {
19.            string country;
20.            cout << "Enter country to search: ";
21.            cin >> country;
22.            if(dict.find(country) != dict.end()) cout << "Capital: " << dict[country] << endl;
23.            else cout << "Country not found\n";
24.        }
25.        else if(ch==3) {
26.            cout << "Country-Capital pairs:\n";
27.            for(auto &p : dict) cout << p.first << " : " << p.second << endl;
28.        }
29.    } while(ch != 4);
30. }
31.
```

Slip 20 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3.
4. int main() {
5.     int a, b;
6.     cout << "Enter two integers: ";
7.     cin >> a >> b;
8.     try {
9.         if(b == 0)
10.             throw "Division by zero error";
11.         cout << "Result = " << a / b;
12.     } catch(const char* msg) {
13.         cout << "Exception: " << msg;
14.     }
15. }
16.
```

Slip 20 Q2.cpp

```
1. #include <iostream>
2. #include <iomanip>
3. using namespace std;
4. class Train {
5. protected:
6.     int Train_no;
7.     string Train_Name;
8. public:
9.     void inputTrain() {
10.         cout << "Enter Train No: ";
11.         cin >> Train_no;
```

```

12.         cout << "Enter Train Name: ";
13.         cin >> Train_Name;
14.     }
15. }
16. class Route : public Train {
17. protected:
18.     int Route_id;
19.     string Source, Destination;
20. public:
21.     void inputRoute() {
22.         inputTrain();
23.         cout << "Enter Route Id: ";
24.         cin >> Route_id;
25.         cout << "Enter Source: ";
26.         cin >> Source;
27.         cout << "Enter Destination: ";
28.         cin >> Destination;
29.     }
30. }
31. class Reservation : public Route {
32.     int Seats;
33.     string Train_Class;
34.     double Fare;
35.     string Date;
36. public:
37.     void inputReservation() {
38.         inputRoute();
39.         cout << "Enter Number of Seats: ";
40.         cin >> Seats;
41.         cout << "Enter Train Class: ";
42.         cin >> Train_Class;
43.         cout << "Enter Fare: ";
44.         cin >> Fare;
45.         cout << "Enter Travel Date: ";
46.         cin >> Date;
47.     }
48.     void display() {
49.         cout << setw(5) << Train_no << setw(12) << Train_Name
50.             << setw(5) << Route_id << setw(10) << Source << setw(12) << Destination
51.                 << setw(6) << Seats << setw(10) << Train_Class
52.                     << setw(8) << Fare << setw(12) << Date << endl;
53.     }
54.     void displayClass(string cls) {
55.         if(Train_Class == cls) display();
56.     }
57. };
58. int main() {
59.     int n, ch;
60.     cout << "Enter number of reservations: ";
61.     cin >> n;
62.     Reservation r[50];
63.     for(int i=0;i<n;i++) r[i].inputReservation();
64.     do {
65.         cout << "\n1. Display all reservations\n2. Display by Train Class\n3. Exit\nEnter choice: ";
66.         cin >> ch;
67.         if(ch==1) {
68.             cout << setw(5) << "TNo" << setw(12) << "TName" << setw(5) << "RId"
69.                 << setw(10) << "Source" << setw(12) << "Dest" << setw(6) << "Seats"
70.                     << setw(10) << "Class" << setw(8) << "Fare" << setw(12) << "Date" << endl;
71.             for(int i=0;i<n;i++) r[i].display();
72.         }
73.         else if(ch==2) {
74.             string cls;
75.             cout << "Enter Train Class: ";
76.             cin >> cls;
77.             cout << setw(5) << "TNo" << setw(12) << "TName" << setw(5) << "RId"
78.                 << setw(10) << "Source" << setw(12) << "Dest" << setw(6) << "Seats"
79.                     << setw(10) << "Class" << setw(8) << "Fare" << setw(12) << "Date" << endl;
80.             for(int i=0;i<n;i++) r[i].displayClass(cls);
81.         }
82.     } while(ch!=3);
83. }
84.

```

Slip 21 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Integer {
4.     int num;
5. public:
6.     void input() {
7.         cout << "Enter integer: ";
8.         cin >> num;
9.     }
10.    void display() {
11.        cout << num << endl;
12.    }
13.    Integer operator+(Integer &i) {
14.        Integer temp;
15.        temp.num = num + i.num;
16.        return temp;
17.    }
18.    Integer operator-(Integer &i) {
19.        Integer temp;
20.        temp.num = num - i.num;
21.        return temp;
22.    }
23.    Integer operator*(Integer &i) {
24.        Integer temp;
25.        temp.num = num * i.num;
26.        return temp;
27.    }
28.    Integer operator/(Integer &i) {
29.        Integer temp;
30.        temp.num = num / i.num;
31.        return temp;
32.    }
33. };
34. int main() {
35.     Integer a, b, c;
36.     a.input();
37.     b.input();
38.     c = a + b;
39.     cout << "Sum: "; c.display();
40.     c = a - b;
41.     cout << "Difference: "; c.display();
42.     c = a * b;
43.     cout << "Product: "; c.display();
44.     c = a / b;
45.     cout << "Quotient: "; c.display();
46. }
47.
```

Slip 21 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Media {
4. protected:
5.     string title;
6.     double price;
7. public:
8.     void input() {
9.         cout << "Enter title: ";
10.        cin >> title;
11.        cout << "Enter price: ";
12.        cin >> price;
13.    }
14.    virtual void display() {
15.        cout << "Title: " << title << " Price: " << price << endl;
16.    }
17. };
18. class Book : public Media {
19.     int pages;
20. public:
```

```

21.     void inputBook() {
22.         input();
23.         cout << "Enter number of pages: ";
24.         cin >> pages;
25.     }
26.     void display() {
27.         cout << "Book - Title: " << title << " Price: " << price << " Pages: " << pages << endl;
28.     }
29. };
30. class Tape : public Media {
31.     double duration;
32. public:
33.     void inputTape() {
34.         input();
35.         cout << "Enter duration: ";
36.         cin >> duration;
37.     }
38.     void display() {
39.         cout << "Tape - Title: " << title << " Price: " << price << " Duration: " << duration << "
mins" << endl;
40.     }
41. };
42. int main() {
43.     Book b;
44.     Tape t;
45.     b.inputBook();
46.     t.inputTape();
47.     Media* m1 = &b;
48.     Media* m2 = &t;
49.     m1->display();
50.     m2->display();
51. }
52.

```

Slip 22 Q1.cpp

```

1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int main() {
5.     ifstream fin("input.txt");
6.     if(!fin) {
7.         cout << "File not found";
8.         return 0;
9.     }
10.    char ch;
11.    int upper=0, lower=0, digit=0, space=0;
12.    while(fin.get(ch)) {
13.        if(ch >= 'A' && ch <= 'Z') upper++;
14.        else if(ch >= 'a' && ch <= 'z') lower++;
15.        else if(ch >= '0' && ch <= '9') digit++;
16.        else if(ch == ' ') space++;
17.    }
18.    fin.close();
19.    cout << "Uppercase letters: " << upper << endl;
20.    cout << "Lowercase letters: " << lower << endl;
21.    cout << "Digits: " << digit << endl;
22.    cout << "Spaces: " << space;
23. }
24.

```

Slip 22 Q2.cpp

```

1. #include <iostream>
2. #include <cmath>
3. using namespace std;
4. class Point {
5.     double x, y;
6. public:
7.     void input() {
8.         cout << "Enter x and y: ";

```

```

9.         cin >> x >> y;
10.    }
11.    void display() {
12.        cout << "(" << x << "," << y << ")";
13.    }
14.    friend double operator-(Point p1, Point p2);
15. };
16. double operator-(Point p1, Point p2) {
17.     return sqrt((p1.x - p2.x)*(p1.x - p2.x) + (p1.y - p2.y)*(p1.y - p2.y));
18. }
19. int main() {
20.     Point a, b;
21.     a.input();
22.     b.input();
23.     cout << "Distance between points: " << (a - b);
24. }
25.

```

Slip 23 Q1.cpp

```

1. #include <iostream>
2. #include <fstream>
3. using namespace std;
4. int countLines() {
5.     ifstream fin("input.txt");
6.     if(!fin)
7.         return -1;
8.     string line;
9.     int count = 0;
10.    while(getline(fin, line)) {
11.        if(!line.empty() && line[0] != 'C')
12.            count++;
13.    }
14.    fin.close();
15.    return count;
16. }
17.
18. int main() {
19.     int c = countLines();
20.     if(c == -1)
21.         cout << "File not found";
22.     else cout << "Number of lines not starting with 'C': " << c;
23. }
24.

```

Slip 23 Q2.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Complex {
4.     double real, imag;
5. public:
6.     void input() {
7.         cout << "Enter real and imaginary parts: ";
8.         cin >> real >> imag;
9.     }
10.    void display() {
11.        cout << real << " + " << imag << "i" << endl;
12.    }
13.    friend Complex add(Complex c1, Complex c2);
14. };
15. Complex add(Complex c1, Complex c2) {
16.     Complex temp;
17.     temp.real = c1.real + c2.real;
18.     temp.imag = c1.imag + c2.imag;
19.     return temp;
20. }
21. int main() {
22.     Complex c1, c2, sum;
23.     c1.input();
24.     c2.input();

```

```
25.     sum = add(c1, c2);
26.     cout << "Sum: ";
27.     sum.display();
28. }
29.
```

Slip 24 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Number {
4.     int a, b;
5. public:
6.     Number() {
7.         a = 5; b = 0;
8.     }
9.     Number(int x, int y) {
10.        a = x; b = y;
11.    }
12.     void displayMax() {
13.         cout << "Maximum: " << (a > b ? a : b) << endl;
14.     }
15. };
16. int main() {
17.     Number n1;
18.     Number n2(10, 20);
19.     n1.displayMax();
20.     n2.displayMax();
21. }
22.
```

Slip 24 Q2.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Employee {
4.     string name;
5.     double salary;
6. public:
7.     void input() {
8.         cout << "Enter name: ";
9.         cin >> name;
10.        cout << "Enter salary: ";
11.        cin >> salary;
12.    }
13.     friend double averageSalary(Employee e[], int n);
14. };
15. double averageSalary(Employee e[], int n) {
16.     double sum = 0;
17.     for(int i=0;i<n;i++) sum += e[i].salary;
18.     return sum / n;
19. }
20. int main() {
21.     int n;
22.     cout << "Enter number of employees: ";
23.     cin >> n;
24.     Employee e[50];
25.     for(int i=0;i<n;i++) e[i].input();
26.     cout << "Average salary: " << averageSalary(e, n);
27. }
28.
```

Slip 25 Q1.cpp

```
1. #include <iostream>
2. using namespace std;
3. class Employee {
4.     int emp_id;
5.     string emp_name;
```

```

6.     double basic_salary;
7. public:
8.     void input() {
9.         cout << "Enter Employee Id: ";
10.        cin >> emp_id;
11.        cout << "Enter Employee Name: ";
12.        cin >> emp_name;
13.        cout << "Enter Basic Salary: ";
14.        cin >> basic_salary;
15.    }
16.    void display() {
17.        cout << "Id: " << emp_id << " Name: " << emp_name << " Salary: " << basic_salary << endl;
18.    }
19.    double getSalary() {
20.        return basic_salary;
21.    }
22.};
23. int main() {
24.     Employee e[5];
25.     for(int i=0;i<5;i++) e[i].input();
26.     cout << "Employees with salary > 5000:\n";
27.     for(int i=0;i<5;i++) {
28.         if(e[i].getSalary() > 5000) e[i].display();
29.     }
30. }
31.

```

Slip 25 Q2.cpp

```

1. #include <iostream>
2. using namespace std;
3. class Student {
4.     int roll_no;
5.     string name;
6.     float marks;
7. public:
8.     void input() {
9.         cout << "Enter Roll No: ";
10.        cin >> roll_no;
11.        cout << "Enter Name: ";
12.        cin >> name;
13.        cout << "Enter Marks: ";
14.        cin >> marks;
15.    }
16.    void display() {
17.        cout << "Roll No: " << roll_no << " Name: " << name << " Marks: " << marks << endl;
18.    }
19.    friend void highestMarks(Student s1, Student s2);
20. };
21. void highestMarks(Student s1, Student s2) {
22.     if(s1.marks >= s2.marks) s1.display();
23.     else s2.display();
24. }
25. int main() {
26.     Student s1, s2;
27.     s1.input();
28.     s2.input();
29.     cout << "Student with highest marks:\n";
30.     highestMarks(s1, s2);
31. }
32.
33.

```