1. Write a C++ program to subtract two integer numbers of two different classes using friend function.

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

class Number2; // forward declaration

class Number1 {
private:
   int num1;
public:
   Number1(int n1) : num1(n1) {}
   friend int subtract(Number1 n1, Number2 n2);
};

class Number2 {
private:
   int num2;
public:
   Number2(int n2) : num2(n2) {}
   friend int subtract(Number1 n1, Number2 n2);
};

int subtract(Number1 n1, Number2 n2) {
   return n1.num1 - n2.num2;
}

int main() {
```

```
    int n1, n2;
    cout << "Enter first integer: ";
    cin >> n1;
    cout << "Enter second integer: ";
    cin >> n2;
    Number1 num1(n1);
    Number2 num2(n2);
    cout << "Subtraction result: " << subtract(num1, num2) << endl;
    return 0;
}
```

==========

2. Write a C++ program to read a text file and count number of Upper-case Alphabets, Lowercase Alphabets Digits and Spaces using File Handling

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int main() {
    string filename;
```

```
cout << "Enter the name of the input file: ";
cin >> filename;

ifstream input_file(filename);
if (!input_file.is_open()) {
   cout << "Unable to open file " << filename << endl;
   return 1;
}

int upper_count = 0;
int lower_count = 0;
int digit_count = 0;
int space_count = 0;
char ch;

while (input_file.get(ch)) {
   if (isupper(ch)) {
      upper_count++;
   } else if (islower(ch)) {
      lower_count++;
   } else if (isdigit(ch)) {
      digit_count++;
   } else if (isspace(ch)) {
      space_count++;
   }
}

input_file.close();

string output_filename;
cout << "Enter the name of the output file: ";
```

```
    cin >> output_filename;

    ofstream output_file(output_filename);
    if (!output_file.is_open()) {
        cout << "Unable to create output file " << output_filename << endl;
        return 1;
    }

    output_file << "Number of uppercase alphabets: " << upper_count << endl;
    output_file << "Number of lowercase alphabets: " << lower_count << endl;
    output_file << "Number of digits: " << digit_count << endl;
    output_file << "Number of spaces: " << space_count << endl;

    output_file.close();

    return 0;
}
```

==========

3. Write a C++ program to overload function volume and find volume of cube, cylinder and sphere.

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

// Function to find the volume of a cube
double volume(double side)
{
```

```
    return side * side * side;
}

// Function to find the volume of a cylinder
double volume(double radius, double height)
{
    return 3.14 * radius * radius * height;
}

// Function to find the volume of a sphere
double Volume(double radius)
{
    return (4.0 / 3.0) * 3.14 * radius * radius * radius;
}

int main()
{
    double s, r, h;

    // Input for cube
    cout << "Enter the length of a side of the cube: ";
    cin >> s;
    cout << "The volume of the cube is " << volume(s) << endl;

    // Input for cylinder
    cout << "Enter the radius of the cylinder: ";
    cin >> r;
    cout << "Enter the height of the cylinder: ";
    cin >> h;
    cout << "The volume of the cylinder is " << volume(r, h) << endl;
```

```
    // Input for sphere
    cout << "Enter the radius of the sphere: ";
    cin >> r;
    cout << "The volume of the sphere is " << Volume(r) << endl;

    return 0;
}
```

==========

4. Write a C++ program to find area and volume of cylinder using Inline function.

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

inline double cylinderArea(double radius, double height) {
    return 2 * 3.14 * radius * height + 2 * 3.14 * radius * radius;
}

inline double cylinderVolume(double radius, double height) {
    return 3.14 * radius * radius * height;
}

int main() {
    double radius, height;
    cout << "Enter the radius of the cylinder: ";
    cin >> radius;
```

```
    cout << "Enter the height of the cylinder: ";
    cin >> height;

    double area = cylinderArea(radius, height);
    double volume = cylinderVolume(radius, height);

    cout << "The area of the cylinder is: " << area << endl;
    cout << "The volume of the cylinder is: " << volume << endl;

    return 0;
}
```

==========

5. Write a C++ program to accept length and width of a rectangle. Calculate and display perimeter as well as area of a rectangle by using Inline function.

```
#include<iostream>

using namespace std;

inline void calculatePerimeterAndArea(float length, float width, float& perimeter, float& area) {
    perimeter = 2 * (length + width);
    area = length * width;
}

int main() {
    float length, width, perimeter, area;
```

```
    cout << "Enter length of rectangle: ";
    cin >> length;
    cout << "Enter width of rectangle: ";
    cin >> width;

    calculatePerimeterAndArea(length, width, perimeter, area);

    cout << "Perimeter of rectangle is " << perimeter << endl;
    cout << "Area of rectangle is " << area << endl;

    return 0;
}
```

==========

6. Write a C++ program using function to count and display the number of lines not starting with alphabet “A” in a text file.

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int countLinesNotStartingWithA(string fileName) {
```

```
    ifstream inputFile;
    inputFile.open(fileName);

    if (!inputFile) {
        cout << "Error opening file " << fileName << endl;
        return -1;
    }

    int count = 0;
    string line;
    while (getline(inputFile, line)) {
        if (line[0] != 'A' && line[0] != 'a') {
            count++;
        }
    }

    inputFile.close();
    return count;
}

int main() {
    string fileName;
    cout << "Enter file name: ";
    cin >> fileName;

    int count = countLinesNotStartingWithA(fileName);

    if (count == -1) {
        return 1;
    }
```

```
    cout << "Number of lines not starting with 'A': " << count << endl;

    return 0;
}
```

==========

7. Write a C++ program using function to count and display the number of lines not starting with alphabet “A” in a text file.

```
#include <iostream>
#include <fstream>
#include <string>

using namespace std;

int countLinesNotStartingWithA(string fileName) {
    ifstream inputFile;
    inputFile.open(fileName);

    if (!inputFile) {
        cout << "Error opening file " << fileName << endl;
        return -1;
    }

    int count = 0;
    string line;
```

```
    while (getline(inputFile, line)) {
        if (line[0] != 'A' && line[0] != 'a') {
            count++;
        }
    }

    inputFile.close();
    return count;
}

int main() {
    string fileName;
    cout << "Enter file name: ";
    cin >> fileName;

    int count = countLinesNotStartingWithA(fileName);

    if (count == -1) {
        return 1;
    }

    cout << "Number of lines not starting with 'A': " << count << endl;

    return 0;
}
```

==========

8. Write a C++ program to swap two integer values and two float values by using function template.

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

template<typename T>
void swap_values(T &a, T &b) {
    T temp = a;
    a = b;
    b = temp;
}

int main() {
    int int1, int2;
    float float1, float2;

    cout << "Enter two integer values: ";
    cin >> int1 >> int2;

    cout << "Enter two float values: ";
    cin >> float1 >> float2;

    cout << "Before swapping:" << endl;
    cout << "Integer values: " << int1 << " " << int2 << endl;
    cout << "Float values: " << float1 << " " << float2 << endl;

    swap_values(int1, int2);
    swap_values(float1, float2);
```

```
    cout << "After swapping:" << endl;
    cout << "Integer values: " << int1 << " " << int2 << endl;
    cout << "Float values: " << float1 << " " << float2 << endl;

    return 0;
}
```

==========

9. Write a C++ program to calculate area of cone, sphere and circle by using function overloading.

```
#include <iostream>
#include <cmath>

using namespace std;

const double PI = 3.14159265;

double area(double radius) {
    return PI * pow(radius, 2);
}

double area(double radius, double height) {
    double slantHeight = sqrt(pow(radius, 2) + pow(height, 2));
    return PI * radius * slantHeight + PI * pow(radius, 2);
}
```

```
double area(double radius, double height, double sideLength) {
    double baseArea = PI * pow(radius, 2);
    double lateralArea = PI * radius * sideLength;
    double totalArea = baseArea + lateralArea;
    return totalArea;
}

int main() {
    double radius, height, sideLength;

    cout << "Enter radius of circle: ";
    cin >> radius;
    cout << "Area of circle: " << area(radius) << endl;

    cout << "Enter radius and height of cone: ";
    cin >> radius >> height;
    cout << "Area of cone: " << area(radius, height) << endl;

    cout << "Enter radius, height, and side length of sphere: ";
    cin >> radius >> height >> sideLength;
    cout << "Area of sphere: " << area(radius, height, sideLength) << endl;

    return 0;
}
```

==========

10. Write a C++ program to create a class which contains two data members. Write member functions to accept display and swap two entered numbers using call by reference.

```
#include <iostream>

using namespace std;

class Numbers {
   private:
      int num1, num2;

   public:
      void accept() {
         cout << "Enter two numbers: ";
         cin >> num1 >> num2;
      }

      void display() {
         cout << "Num1 = " << num1 << ", Num2 = " << num2 << endl;
      }

      void swap() {
         int temp = num1;
         num1 = num2;
         num2 = temp;
      }
};

int main() {
   Numbers n;
```

```
    n.accept();
    cout << "Before swapping:" << endl;
    n.display();
    n.swap();
    cout << "After swapping:" << endl;
    n.display();

    return 0;
}
```

==========

11. Write a C++ program to create a class Student which contains data members as Roll_Number, Stud_Name, Percentage. Write member functions to accept Student information. Display all details of student along with a class obtained depending on percentage. (Use array of objects).

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

class Student {
    private:
        int Roll_Number;
        string Stud_Name;
```

```
    float Percentage;
  public:
    void accept_student();
    void display_student();
    string get_class();
};

void Student::accept_student() {
  cout << "Enter Roll Number: ";
  cin >> Roll_Number;
  cout << "Enter Student Name: ";
  cin.ignore();
  getline(cin, Stud_Name);
  cout << "Enter Percentage: ";
  cin >> Percentage;
}

void Student::display_student() {
  cout << "Roll Number: " << Roll_Number << endl;
  cout << "Student Name: " << Stud_Name << endl;
  cout << "Percentage: " << Percentage << "%" << endl;
}

string Student::get_class() {
  if (Percentage >= 90) {
    return "First Class with Distinction";
  } else if (Percentage >= 75) {
    return "First Class";
  } else if (Percentage >= 60) {
    return "Second Class";
  } else if (Percentage >= 50) {
```

```
        return "Pass Class";
    } else {
        return "Fail";
    }
}

int main() {
    const int num_students = 3;
    Student students[num_students];

    for (int i = 0; i < num_students; i++) {
        cout << "Enter details of student " << i+1 << ":" << endl;
        students[i].accept_student();
        cout << endl;
    }

    for (int i = 0; i < num_students; i++) {
        cout << "Details of student " << i+1 << ":" << endl;
        students[i].display_student();
        cout << "Class Obtained: " << students[i].get_class() << endl;
        cout << endl;
    }

    return 0;
}
```

==========

12. Write a C++ program to create a class Item with data members Item_Code, Item_Name, Item_Price. Write member functions to accept and display Item information also display number of objects created for a class. (Use Static data member and Static member function).

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

class Item {
   private:
      int Item_Code;
      string Item_Name;
      float Item_Price;
      static int count;
   public:
      void accept_item();
      void display_item();
      static int get_count();
};

int Item::count = 0;

void Item::accept_item() {
   cout << "Enter item code: ";
   cin >> Item_Code;
   cout << "Enter item name: ";
   cin.ignore();
   getline(cin, Item_Name);
   cout << "Enter item price: ";
```

```
    cin >> Item_Price;
    count++;
}

void Item::display_item() {
    cout << "Item Code: " << Item_Code << endl;
    cout << "Item Name: " << Item_Name << endl;
    cout << "Item Price: " << Item_Price << endl;
}

int Item::get_count() {
    return count;
}

int main() {
    Item item1, item2, item3;
    item1.accept_item();
    item2.accept_item();
    item3.accept_item();

    cout << "Item 1 Details: " << endl;
    item1.display_item();
    cout << endl;

    cout << "Item 2 Details: " << endl;
    item2.display_item();
    cout << endl;

    cout << "Item 3 Details: " << endl;
    item3.display_item();
    cout << endl;
```

```
    cout << "Number of objects created: " << Item::get_count() << endl;
    return 0;
}
```

==========

13. Implement a class Complex which represents the Complex Number data type. Implement the following operations: 1. Constructor (including a default constructor which creates the complex number0+0i). 2. Overloaded operator+ to add two complex numbers.

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

class Complex {
private:
    double real;
    double imaginary;
public:
    Complex() {
        real = 0;
        imaginary = 0;
    }
    Complex(double r, double i) {
        real = r;
        imaginary = i;
    }
```

```
    Complex operator+(Complex const &obj) {
        Complex res;
        res.real = real + obj.real;
        res.imaginary = imaginary + obj.imaginary;
        return res;
    }
    void print() {
        cout << real << " + " << imaginary << "i" << endl;
    }
};

int main() {
    double r1, i1, r2, i2;
    cout << "Enter real and imaginary part of first complex number: ";
    cin >> r1 >> i1;
    cout << "Enter real and imaginary part of second complex number: ";
    cin >> r2 >> i2;

    Complex c1(r1, i1);
    Complex c2(r2, i2);
    Complex c3 = c1 + c2;

    cout << "Sum of two complex numbers: ";
    c3.print();

    return 0;
}
```

==========

14. Write a C++ program create a calculator for an arithmetic operator (+, -, *, /). The program should take two operands from user and performs the operation on those two operands depending upon the operator entered by user. Use a switch statement to select the operation.

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

int main() {
   float operand1, operand2;
   char op;

   cout << "Enter operator (+, -, *, /): ";
   cin >> op;

   cout << "Enter Frist Number: ";
   cin >> operand1 ;
   cout << "Enter Second Number: ";
          cin>> operand2;

   switch (op) {
      case '+':
         cout << "Result: " << operand1 + operand2 << endl;
         break;
      case '-':
         cout << "Result: " << operand1 - operand2 << endl;
         break;
```

```
        case '*':
            cout << "Result: " << operand1 * operand2 << endl;
            break;
        case '/':
            if (operand2 == 0) {
                cout << "Error: division by zero" << endl;
            } else {
                cout << "Result: " << operand1 / operand2 << endl;
            }
            break;
        default:
            cout << "Error: invalid operator" << endl;
    }

    return 0;
}
```

==========

15. Write a function in C++ to count and display the number of lines not starting with alphabet 'A' present in a text file "STORY.TXT". Example: If the file "STORY.TXT" contains the following lines, The roses are red. A girl is playing there. There is a playground. An aeroplane is in the sky. Numbers are not allowed in the password. The function should display the output as 3.

```
#include <iostream>
#include <fstream>
#include <string>
```

```
using namespace std;

void count_lines() {
    ifstream file("STORY.TXT");
    if (!file.is_open()) {
        cout << "Error: file not found" << endl;
        return;
    }

    int count = 0;
    string line;
    while (getline(file, line)) {
        if (line.empty() || line[0] == 'A' || line[0] == 'a') {
            continue;
        }
        count++;
    }

    cout << "Number of lines not starting with 'A': " << count << endl;
}

int main() {
    count_lines();
    return 0;
}
```

==========

16. Create a C++ class for a student object with the following attributes—roll no, name, number of subjects, marks of subjects. Write member function for

accepting marks and display all information of student along with total and Percentage. Display marklist with Use of manipulators.

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

class Student {
private:
    int roll_no;
    string name;
    int num_subjects;
    int* marks;

public:
    Student(int r, string n, int num) {
        roll_no = r;
        name = n;
        num_subjects = num;
        marks = new int[num_subjects];
    }

    void accept_marks() {
        for (int i = 0; i < num_subjects; i++) {
            cout << "Enter marks for subject " << i + 1 << ": ";
            cin >> marks[i];
        }
    }

    void display_info() {
```

```
    cout << "Roll no: " << roll_no << endl;
    cout << "Name: " << name << endl;
    cout << "Number of subjects: " << num_subjects << endl;
    cout << "Marks: ";
    for (int i = 0; i < num_subjects; i++) {
        cout << marks[i] << " ";
    }
    cout << endl;

    int total_marks = 0;
    for (int i = 0; i < num_subjects; i++) {
        total_marks += marks[i];
    }
    double percentage = (double)total_marks / num_subjects;

    cout << "Total marks: " << total_marks << endl;
    cout << "Percentage: " << fixed << setprecision(2) << percentage << "%" << endl;
}

void display_marklist() {
    cout << "Roll no: " << setw(10) << "Name: " << setw(20);
    for (int i = 0; i < num_subjects; i++) {
        cout << "Subject " << i + 1 << setw(10);
    }
    cout << "Total" << endl;

    cout << roll_no << setw(10) << name << setw(20);
    int total_marks = 0;
    for (int i = 0; i < num_subjects; i++) {
        cout << marks[i] << setw(10);
        total_marks += marks[i];
```

```
        }
        cout << total_marks << endl;
    }

    ~Student() {
        delete[] marks;
    }
};

int main() {
    int roll_no, num_subjects;
    string name;

    cout << "Enter roll no: ";
    cin >> roll_no;

    cout << "Enter name: ";
    cin.ignore();
    getline(cin, name);

    cout << "Enter number of subjects: ";
    cin >> num_subjects;

    Student student(roll_no, name, num_subjects);
    student.accept_marks();

    cout << "\nStudent information:\n";
    student.display_info();

    cout << "\nMark list:\n";
    student.display_marklist();
```

```
    return 0;
}
```

===