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

```
#include <iostream>
using namespace std;
class ClassB;
class ClassA {
private:
   int a;
public:
   ClassA(int num) {
      a = num;
   }
   friend int subtract(ClassA objA, ClassB objB);
};
class ClassB {
private:
   int b;
public:
   ClassB(int num) {
      b = num;
   }
   friend int subtract(ClassA objA, ClassB objB);
};
int subtract(ClassA objA, ClassB objB) {
   return objA.a - objB.b;
}
int main() {
    ClassA objA(10);

    ClassB objB(5);

    int result = subtract(objA, objB);

    cout << "Result: " << result << 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>
using namespace std;
int main() {
    ifstream file("sample.txt");
    char c;
    int upper = 0, lower = 0, digits = 0, spaces = 0;
    while (file.get(c)) {
        if (isupper(c)) {
            upper++;
        }
        else if (islower(c)) {
            lower++;
        }
        else if (isdigit(c)) {
            digits++;
        }
        else if (isspace(c)) {
            spaces++;
        }
}
    file.close();
    cout << "Number of uppercase alphabets: " << upper << endl;

    cout << "Number of lowercase alphabets: " << lower << endl;

    cout << "Number of digits: " << digits << endl;

    cout << "Number of spaces: " << spaces << endl;

    return 0;

}
```

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

```
#include <iostream>
using namespace std;
const double PI = 3.14159;
double volume(double side) {
   return side * side * side;
}
double volume(double radius, double height) {
   return PI * radius * radius * height;
}
double volume(double radius) {
   return 4.0 / 3.0 * PI * radius * radius * radius;
}
int main() {
   double cubeSide = 3.0;
   double cylinderRadius = 2.0;
   double cylinderHeight = 5.0;
   double sphereRadius = 4.0;
   double cubeVolume = volume(cubeSide);
   double cylinderVolume = volume(cylinderRadius, cylinderHeight);
   double sphereVolume = volume(sphereRadius);
   cout << "Volume of cube with side " << cubeSide << " is " << cubeVolume << endl;
   cout << "Volume of cylinder with radius " << cylinderRadius << " and height " << cylinderHeight << " is " << cylinderVolume << endl;
   cout << "Volume of sphere with radius " << sphereRadius << " is " << sphereVolume << endl;

   return 0;

}
```

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

```
#include <iostream>
using namespace std;
const double PI = 3.14159;
inline double cylinderArea(double radius, double height) {
    return 2 * PI * radius * height + 2 * PI * radius * radius; // area of cylinder
}
inline double cylinderVolume(double radius, double height) {
    return PI * radius * radius * height; // volume of cylinder
}
int main() {
    double radius = 2.0;
    double height = 5.0;
    double area = cylinderArea(radius, height);
    double volume = cylinderVolume(radius, height);
    cout << "Area of cylinder with radius " << radius << " and height " << height << " is " << area << endl;
    cout << "Volume of cylinder with radius " << radius << " and height " << height << " 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 calculate(int length, int width, int& area, int& perimeter) {

   area = length * width;

   perimeter = 2 * (length + width);

}

int main() {

   int length, width, area, perimeter;

   cout << "Enter length: ";

   cin >> length;

   cout << "Enter width: ";

   cin >> width;

   calculate(length, width, area, perimeter);

   cout << "Area: " << area << endl;

   cout << "Perimeter: " << perimeter << 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(ifstream& file) {

    string line;

    int count = 0;

    while (getline(file, line)) {

        if (line[0] != 'A' && line[0] != 'a') {

            count++;

        }

    }

    return count;

}

int main() {

    ifstream file("sample.txt");

    if (!file) {

        cout << "Error opening file!" << endl;

        return 1;

    }

    int count = countLinesNotStartingWithA(file);

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

    file.close();

    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(ifstream& file) {
    string line;
    int count = 0;
    while (getline(file, line)) {
        if (line[0] != 'A' && line[0] != 'a') {
            count++;
        }
    }
    return count;
}
int main() {
    ifstream file("sample.txt");
    if (!file) {
        cout << "Error opening file!" << endl;
        return 1;
    }
    int count = countLinesNotStartingWithA(file);
    cout << "Number of lines not starting with A: " << count << endl;
    file.close();
    return 0;
}
```

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

```
#include <iostream>

template <typename T>

void swap(T& a, T& b)

{

    T temp = a;

    a = b;

    b = temp;

}

int main()

{

    int num1 = 5, num2 = 10;

    float f1 = 3.14, f2 = 2.71;

    std::cout << "Before swapping:" << std::endl;

    std::cout << "num1 = " << num1 << ", num2 = " << num2 << std::endl;

    std::cout << "f1 = " << f1 << ", f2 = " << f2 << std::endl;

    swap(num1, num2);

    swap(f1, f2);

    std::cout << "After swapping:" << std::endl;

    std::cout << "num1 = " << num1 << ", num2 = " << num2 << std::endl;

    std::cout << "f1 = " << f1 << ", f2 = " << f2 << std::endl;

    return 0;

}
```

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

```cpp
#include <iostream>
#include <cmath>
const double PI = 3.14159265358979323846;
double area(double radius)
{
   return PI * radius * radius;
}
double area(double radius, double height)
{
   return PI * radius * sqrt(radius * radius + height * height);
}
double area(double radius, double height, double slantHeight)
{
   return PI * radius * (radius + slantHeight);
}
int main()
{
   double radius, height, slantHeight;
   std::cout << "Enter the radius of the circle: ";
   std::cin >> radius;
   std::cout << "Area of the circle: " << area(radius) << std::endl;
   std::cout << "Enter the radius and height of the cone: ";
   std::cin >> radius >> height;
   std::cout << "Area of the cone: " << area(radius, height) << std::endl;
```

```
    std::cout << "Enter the radius, height, and slant height of the sphere: ";
    std::cin >> radius >> height >> slantHeight;
    std::cout << "Area of the sphere: " << area(radius, height, slantHeight) << std::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 Number {
private:
   int num1, num2;
public:
   void accept() {
      cout << "Enter two numbers: ";
      cin >> num1 >> num2;
   }
   void display() {
      cout << "Numbers: " << num1 << " and " << num2 << endl;
   }
   void swap() {
      int temp = num1;
      num1 = num2;
      num2 = temp;
   }
};
int main() {
   Number n;
   n.accept();
   n.display();
```

```
    n.swap();
    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>
using namespace std;
class Student {
private:
   int Roll_Number;
   string Stud_Name;
   float Percentage;
public:
   void accept() {
      cout << "Enter Roll Number: ";
      cin >> Roll_Number;
      cout << "Enter Student Name: ";
      cin >> Stud_Name;
      cout << "Enter Percentage: ";
      cin >> Percentage;
   }
   void display() {
      cout << "Roll Number: " << Roll_Number << endl;
      cout << "Student Name: " << Stud_Name << endl;
      cout << "Percentage: " << Percentage << endl;
      if (Percentage >= 60)
         cout << "Class Obtained: First Class" << endl;
      else if (Percentage >= 50)
```

```
            cout << "Class Obtained: Second Class" << endl;
        else if (Percentage >= 40)
            cout << "Class Obtained: Pass Class" << endl;
        else
            cout << "Class Obtained: Fail" << endl;
    }
};
int main() {
    int n;
    cout << "Enter number of students: ";
    cin >> n;
    Student *s = new Student[n];
    for (int i = 0; i < n; i++) {
        cout << "Enter details of Student " << i+1 << ":" << endl;
        s[i].accept();
    }
    cout << endl << "Details of all Students:" << endl;
    for (int i = 0; i < n; i++) {
        cout << endl << "Details of Student " << i+1 << ":" << endl;
        s[i].display();
    }
    delete [] s;
    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:
   Item() {
      count++;
   }
   void accept() {
      cout << "Enter Item Code: ";
      cin >> Item_Code;
      cout << "Enter Item Name: ";
      cin >> Item_Name;
      cout << "Enter Item Price: ";
      cin >> Item_Price;
   }
   void display() {
      cout << "Item Code: " << Item_Code << endl;
```

```
        cout << "Item Name: " << Item_Name << endl;
        cout << "Item Price: " << Item_Price << endl;
    }
    static void displayCount() {
        cout << "Number of Items created: " << count << endl;
    }
};
int Item::count = 0;
int main() {
    Item i1, i2, i3;
    cout << "Enter details of Item 1:" << endl;
    i1.accept();
    cout << "Enter details of Item 2:" << endl;
    i2.accept();
    cout << "Enter details of Item 3:" << endl;
    i3.accept();
    cout << endl << "Details of all Items:" << endl;
    i1.display();
    i2.display();
    i3.display();
    Item::displayCount();
    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, imag;

public:

   Complex(double r = 0, double i = 0) {

      real = r;

      imag = i;

   }

   Complex operator+ (Complex const &obj) {

      Complex res;

      res.real = real + obj.real;

      res.imag = imag + obj.imag;

      return res;

   }

   void print() {

      cout << real << "+" << imag << "i" << endl;

   }

};

int main() {

   Complex c1(1, 2);

   Complex c2(3, 4);
```

```
    Complex c3 = c1 + c2;

    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()

{

   char op;

   double num1, num2, result;

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

   cin >> op;

   cout << "Enter two numbers: ";

   cin >> num1 >> num2;

   switch(op)

   {

      case '+':

         result = num1 + num2;

         break;

      case '-':

         result = num1 - num2;

         break;

      case '*':

         result = num1 * num2;

         break;

      case '/':

         result = num1 / num2;
```

```
            break;
        default:
            cout << "Invalid operator" << endl;
            return 1;
    }
    cout << num1 << " " << op << " " << num2 << " = " << result << 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 countLines()

{

  ifstream file("STORY.TXT");

  string line;

  int count = 0;

  if(file.is_open())

  {

    while(getline(file, line))

    {

      if(line[0] != 'A' && line[0] != 'a')

      {

        count++;

      }

    }

    file.close();

  }

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

}
```

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>

#include <string>

using namespace std;

class Student {

  private:

    int rollNo;

    string name;

    int numSubjects;

    int *marks;

    int totalMarks;

    float percentage;

  public:

    Student(int roll, string n, int num) {

        rollNo = roll;

        name = n;

        numSubjects = num;

        marks = new int[numSubjects];

        totalMarks = 0;

        percentage = 0.0;

    }

    void acceptMarks() {

        cout << "Enter marks of " << name << ":\n";
        for (int i = 0; i < numSubjects; i++) {
            cout << "Subject " << i+1 << ": ";
            cin >> marks[i];
            totalMarks += marks[i];
        }
        percentage = (float)totalMarks / (numSubjects * 100) * 100;
    }
    void displayInfo() {
        cout << "Roll No: " << rollNo << endl;
        cout << "Name: " << name << endl;
        cout << "Number of Subjects: " << numSubjects << endl;
        cout << "Marks: ";
        for (int i = 0; i < numSubjects; i++) {
            cout << marks[i] << " ";
        }
        cout << endl;
        cout << "Total Marks: " << totalMarks << endl;
        cout << "Percentage: " << setprecision(2) << fixed << percentage << "%" << endl;
    }
    void displayMarkList() {
        cout << setw(5) << "Roll No" << setw(10) << "Name" << setw(20) << "Marks" << setw(10)
<< "Total" << setw(12) << "Percentage" << endl;
        cout << setw(5) << rollNo << setw(10) << name << " ";
        for (int i = 0; i < numSubjects; i++) {
            cout << setw(5) << marks[i] << " ";
        }
        cout << setw(10) << totalMarks << setw(12) << setprecision(2) << fixed << percentage <<
"%" << endl;
```

```
    }
    ~Student() {
        delete[] marks;
    }
};
int main() {
    Student s(1, "John", 3);
    s.acceptMarks();
    s.displayInfo();
    s.displayMarkList();
    return 0;
}
```