1.

```
// Sum of odd and wvwn numbers in a 2-D Array
#include<iostream>
using namespace std;
class element {
    private:
        int p1,p2;
    public:
        void set(int m,int n){
            p1=m;
            p2=n;
    }
    void check()
    {
        int arr[p1][p2];
        int i, j, sE = 0, sO = 0;
        for (i = 0; i < p1; i++)
        {
            for (j = 0; j < p2; j++)
            {
                cout << "Enter the number : ";
                cin >> arr[i][j];
                if (arr[i][j] % 2 == 0)
                    sE = sE + arr[i][j];
                if (arr[i][j] % 2 != 0)
                    sO = sO + arr[i][j];
            }
        }
        cout<<endl<< "The sum of even numbers is : "<< sE;
        cout<<endl<< "The sum of odd numbers is :"<< sO;
    }
};
int main(){
element object;
int m,n;
cout << "Enter the 2 parameters of 2D Array :" ;
cin >> m >> n;
object.set(m,n);
object.check();

return 0;
}
```

2.

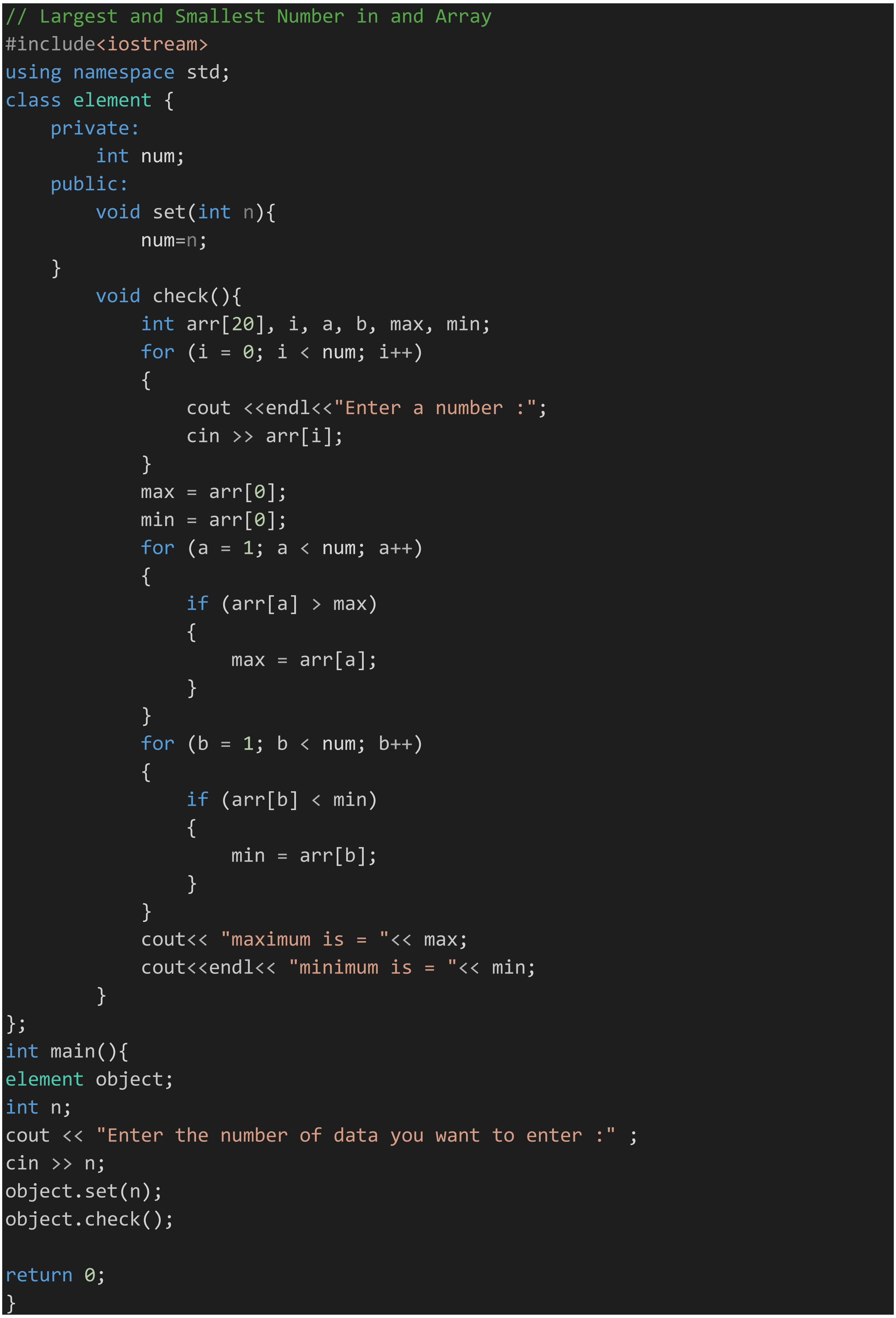
```
// 2-D Array Transpose
#include<iostream>
using namespace std;
class element {
    private:
        int row,column,arr[10][10];
    public:
        void set(int m,int n){
            row=m;
            column=n;
            arr[row][column];
    }
        void insert(){
            for (int i = 0; i < row; ++i)
            {
                for (int j = 0; j < column; ++j)
                {
                    cout << "Enter element a" << i + 1 << j + 1 << ": ";
                    cin >> arr[i][j];
                }
            }
        }
        void print(){
            cout << "\nEntered Matrix: " << endl;
            for (int i = 0; i < row; ++i)
            {
                for (int j = 0; j < column; ++j)
                {
                    cout << " " << arr[i][j];
                    if (j == column - 1)
                        cout << endl;
                }
            }
        }
        void transpose(){
            cout << "\nTranspose of Matrix: " << endl;
            for (int i = 0; i < column; ++i)
                for (int j = 0; j < row; ++j)
                {
                    cout << " " << arr[j][i];
                    if (j == row - 1)
                        cout << endl;
                }
        }
};
int main(){
```

```
element object;
int m,n;
cout <<"Enter the dimensional size of matrix :";
cin >> m >> n;
object.set(m,n);
object.insert();
object.print();
object.transpose();
return 0;
}
```

3.

```
// Armstrong Number Check
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
        int originalNum=num, remainder, result = 0;
        while (originalNum != 0) {
        remainder = originalNum % 10;
        result += remainder * remainder * remainder;
        originalNum /= 10;
    }

    if (result == num)
        cout << num << " is an Armstrong number.";
    else
        cout << num << " is not an Armstrong number.";

    }

};
int main(){
element object;
int n;
cout <<"Enter a number: ";
cin >> n;
object.set(n);
object.check();

return 0;
}
```

4.



```
// Largest and Smallest Number in and Array
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int arr[20], i, a, b, max, min;
            for (i = 0; i < num; i++)
            {
                cout <<endl<<"Enter a number :";
                cin >> arr[i];
            }
            max = arr[0];
            min = arr[0];
            for (a = 1; a < num; a++)
            {
                if (arr[a] > max)
                {
                    max = arr[a];
                }
            }
            for (b = 1; b < num; b++)
            {
                if (arr[b] < min)
                {
                    min = arr[b];
                }
            }
            cout<< "maximum is = "<< max;
            cout<<endl<< "minimum is = "<< min;
        }
};
int main(){
element object;
int n;
cout << "Enter the number of data you want to enter :" ;
cin >> n;
object.set(n);
object.check();

return 0;
}
```

5.

```cpp
// Printing The reverse of an Array
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int arr[num], i, a;
        for (i = 0; i < num; i++)
        {
            cout << "\n Enter a number :";
            cin >> arr[i];
        }
        cout << "\n Array in reverse :";
        for (a = num - 1; a >= 0; a--)
        {
            cout<<""<<arr[a];
        }
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter The number of elements of array :";
    cin >> n;
    object.set(n);
    object.check();

    return 0;
}
```

6.

```
// Even Multiple of 3 check
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            if (num%3==0 && num%2==0)
            {
                cout << "The number is an even multiple of 3";
            }
            else
            {
                cout << "The number is not an even multiple of 3";
            }


        }

};
int main(){
element object;
int n;
cout << "Enter a number :" ;
cin >> n;
object.set(n);
object.check();
return 0;
}
```

```
7.
```

```
// Even and Odd number check
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            if(num==0)
                cout << "The number is 0" ;
            else if(num%2==0)
                cout << "The number is even" ;
            else
                cout << "The number is odd" ;
        }
};
int main(){
element object;
int n;
cout << "Enter a number :" ;
cin >> n;
object.set(n);
object.check();
return 0;
}
```

8.

```cpp
// Printing of Fibonacci Series upto n
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int a = 0, b = 1, c;
        printf("The series runs as follows:");
        while (a <= num)
        {
            c = a + b;
            cout << " " << a ;
            a = b;
            b = c;
        }
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter the limit : ";
    cin >> n;
    object.set(n);
    object.check();
    return 0;
}
```

9.

```cpp
// Printing of Floyd's Triangle Pattern
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int i, j, m = 1;
            for (i = 1; i <= num; i++){
                for (j = 1; j <= i; ++j){
                    cout << " "<< m;
                    ++ m;
                }
                cout << endl ;
            }
        }
};
int main(){
element object;
int n;
cout << "Enter the number of Rows";
cin >> n;
object.set(n);
object.check();
return 0;
}
```

10.

```cpp
// Finding The largest number among 3
#include<iostream>
using namespace std;
class element {
    private:
        int a,b,c;
    public:
        void set(int m , int n , int o){
            a=m;
            b=n;
            c=o;
    }
        void largest(){
            int big = a>c?(a>b?a:b):(b>c?b:c) ;
            cout<<"The largest number is :"<< big;
        }
};
int main(){
element object;
int m,n,o;
cout << "Enter 1st number : " ;
cin >> m ;
cout << "Enter 2nd number : " ;
cin >> n ;
cout << "Enter 3rd number : " ;
cin >> o ;
object.set(m,n,o);
object.largest();

return 0;
}
```

11.

```
// Finding the largest and smallest number among n numbers
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n){
        num = n;
    }
    void check(){
        int max = 0, min = 0, no;
        for (int i = 0; i < num; i++){
            cout << "Enter the number :";
            cin >> no;
            if (i == 0){
                max = no;
                min = no;
            }
            if (no > max){
                max = no;
            }
            if (no < max){
                min = no;
            }
        }
        cout << "The largest number is :" << max << endl
             << "The Least number is :" << min;
    }
};
int main(){
    element object;
    int n;
    cout << "Enter the number of numbers :";
    cin >> n;
    object.set(n);
    object.check();
    return 0;
}
```

```
12.
```

```
// Finding The odd and even factors
#include<iostream>
#include<conio.h>
using namespace std;
class factors
{
    private:
        int n;
    public:
        void input(int num){
            n=num;
        }
        void output(){
            cout<<endl<<"The even factors are : ";
            for(int i=1;i<=n;i++){
                if(n%i==0 && i%2==0){
                    cout<<i<<" ";
                }
            }
            cout<<endl<<"The odd factors are : ";
            for(int i=1;i<=n;i++){
                if(n%i==0 && i%2!=0){
                    cout<<i<<" ";
                }
            }
        }
};
int main(){
    factors obj;
    int num;
    cout<<"Enter the number : ";
    cin>>num;
    obj.input(num);
    obj.output();
    getch();
}
```

13.

```
// Finding the sum of the odd and even digits of a number
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int cpy, sum1 = 0, sum2 = 0, r;
            cpy = num;
            while (cpy > 0)
            {
                r = cpy % 10;
                if (r % 2 == 0)
                    sum1 += r;
                else
                    sum2 += r;
                cpy = cpy / 10;
            }
            cout<< "The Even sum is : " << sum1;
            cout<< "\nThe odd sum is : " << sum2;
    }

};
int main(){
element object;
int n;
cout << "Enter a number :";
cin >> n;
object.set(n);
object.check();

return 0;
}
```

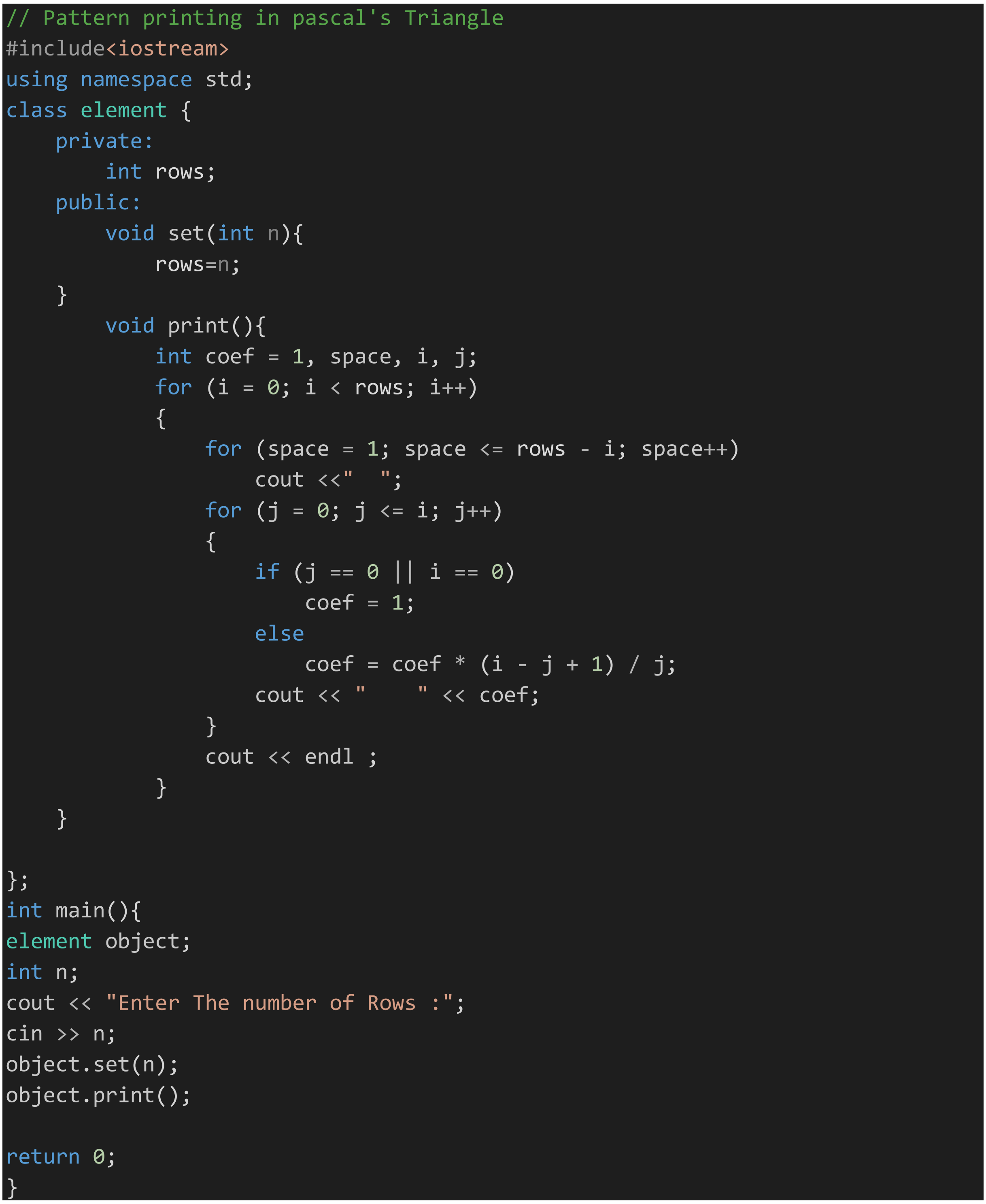
14.

```
// Finding the odd and even Sum of n numbers
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int i=0,sum1=0,sum2=0;
            while(i <= num){
            if(i%2==0)
                sum1=sum1+i;
            else
                sum2=sum2+i;
                i++;
            }
            cout << "Sum of even is : " << sum1 << "\nSum of odd is : " <<
sum2 ;
    }

};
int main(){
element object;
int l, n;
cout << "Enter the limit and the number : ";
cin >> n;
object.set(n);
object.check();
return 0;
}
```

15.

```
// Checking if a number is a palindrome
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int cpy, rev = 0, r;
            cpy = num;
            while (cpy > 0)
            {
                r = cpy % 10;
                rev = rev * 10 + r;
                cpy = cpy / 10;
            }
            if (rev == num)
                cout << "The number is a palindrome";
            else
                cout << "The number is not a palindrome";
        }
};
int main(){
element object;
int n;
cout << "Enter a number :";
cin >> n;
object.set(n);
object.check();

return 0;
}
```

16.



```cpp
// Pattern printing in pascal's Triangle
#include<iostream>
using namespace std;
class element {
    private:
        int rows;
    public:
        void set(int n){
            rows=n;
    }
        void print(){
            int coef = 1, space, i, j;
            for (i = 0; i < rows; i++)
            {
                for (space = 1; space <= rows - i; space++)
                    cout <<"  ";
                for (j = 0; j <= i; j++)
                {
                    if (j == 0 || i == 0)
                        coef = 1;
                    else
                        coef = coef * (i - j + 1) / j;
                    cout << "    " << coef;
                }
                cout << endl ;
            }
    }

};
int main(){
element object;
int n;
cout << "Enter The number of Rows :";
cin >> n;
object.set(n);
object.print();

return 0;
}
```

17.

```
//     *
//    **
//   ***
//  ****
// *****
// Above Pattern Printing
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int m = 1;
        for (int i = num; i >= 1; i--)
        {
            for (int j = 1; j <= i - 1; j++)
            {
                cout<<" ";
            }
            for (int k = 1; k <= m; k++)
            {
                cout<<"*";
            }
            cout<<endl;
            m++;
        }
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter the number of rows :";
    cin >> n;
    object.set(n);
    object.check();

    return 0;
}
```

18.

```
// *****
//  ****
//   ***
//    **
//     *
// Above pattern printing
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int m;
        m = num;
        for (int i = 1; i <= num; i++)
        {
            for (int j = 1; j < i; j++)
            {
                cout <<" ";
            }
            for (int k = 1; k <= m; k++)
            {
                cout <<"*";
            }
            m--;
            cout<<endl;
        }
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter the number of rows :" ;
    cin >> n;
    object.set(n);
    object.check();
    return 0;
}
```

```
19.
```

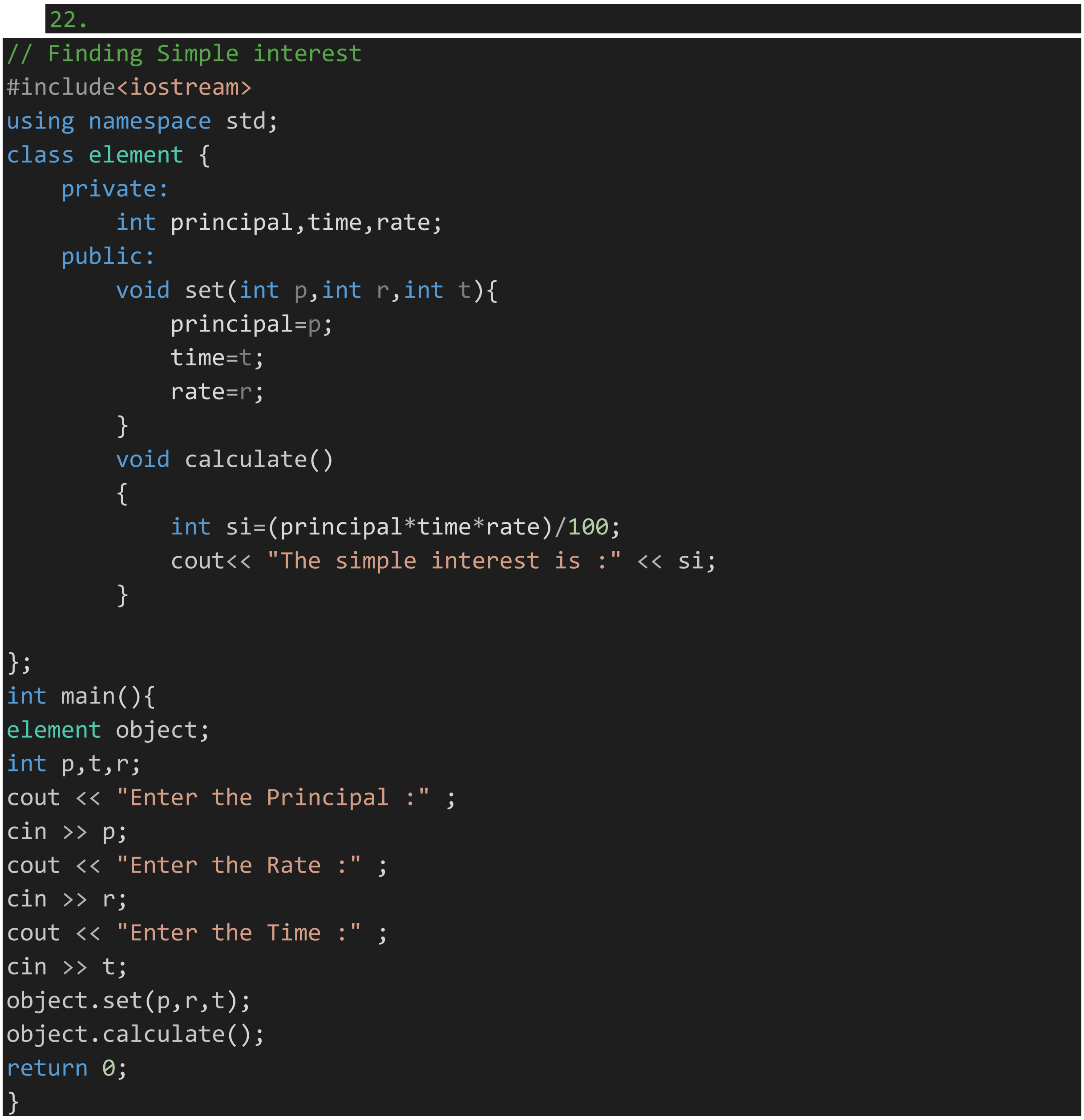
```cpp
// *********
//  *******
//   *****
//    ***
//     *
// Above Pattern Print
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int m = 1;
        for (int i = num; i >= 1; i--)
        {
            for (int j = 1; j < m; j++)
            {
                cout <<" ";
            }
            for (int k = 1; k <= 2 * i - 1; k++)
            {
                cout <<"*";
            }
            m++;

            cout <<endl;
        }
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter the number of rows :";
    cin >> n;
    object.set(n);
    object.check();
    return 0;
}
```

20.

```
// Checking if a number is a perfect number
#include<iostream>
using namespace std;
class element {
    private:
        int num;
    public:
        void set(int n){
            num=n;
    }
        void check(){
            int sum = 0;
            for (int i = 1; i < num; i++)
            {
                if (num % i == 0)
                    sum = sum + i;
            }
            if (num == sum)
                cout <<"The number is perfect";
            else
                cout <<"The number is not perfect";
        }
};
int main(){
element object;
int n;
cout << "Enter a number :";
cin >> n;
object.set(n);
object.check();
return 0;
}
```

21.

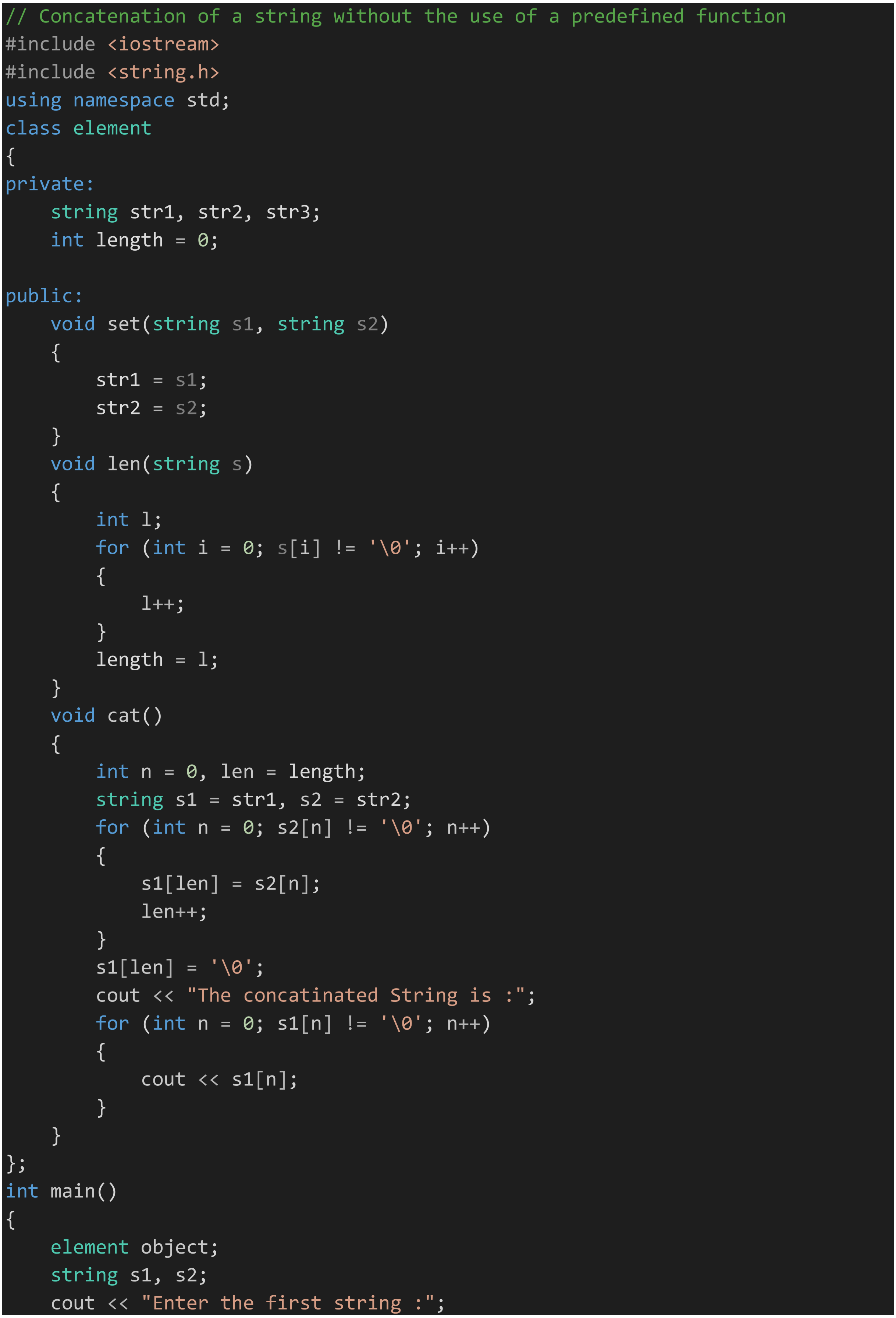
```cpp
// Checking if a number is prime
#include <iostream>
using namespace std;
class element
{
private:
    int num;

public:
    void set(int n)
    {
        num = n;
    }
    void check()
    {
        int b = 0;
        for(int i=1;i<=num;i++)
        {
            if (num % i == 0)
                b++;
        }
        if (b == 2)
            cout << "The number is prime";
        else
            cout << "The number is not prime";
    }
};
int main()
{
    element object;
    int n;
    cout << "Enter the number :";
    cin >> n;
    object.set(n);
    object.check();
    return 0;
}
```



```
22.
```

```
// Finding Simple interest
#include<iostream>
using namespace std;
class element {
    private:
        int principal,time,rate;
    public:
        void set(int p,int r,int t){
            principal=p;
            time=t;
            rate=r;
        }
        void calculate()
        {
            int si=(principal*time*rate)/100;
            cout<< "The simple interest is :" << si;
        }

};
int main(){
element object;
int p,t,r;
cout << "Enter the Principal :" ;
cin >> p;
cout << "Enter the Rate :" ;
cin >> r;
cout << "Enter the Time :" ;
cin >> t;
object.set(p,r,t);
object.calculate();
return 0;
}
```

23.



```cpp
// Concatenation of a string without the use of a predefined function
#include <iostream>
#include <string.h>
using namespace std;
class element
{
private:
    string str1, str2, str3;
    int length = 0;

public:
    void set(string s1, string s2)
    {
        str1 = s1;
        str2 = s2;
    }
    void len(string s)
    {
        int l;
        for (int i = 0; s[i] != '\0'; i++)
        {
            l++;
        }
        length = l;
    }
    void cat()
    {
        int n = 0, len = length;
        string s1 = str1, s2 = str2;
        for (int n = 0; s2[n] != '\0'; n++)
        {
            s1[len] = s2[n];
            len++;
        }
        s1[len] = '\0';
        cout << "The concatinated String is :";
        for (int n = 0; s1[n] != '\0'; n++)
        {
            cout << s1[n];
        }
    }
};
int main()
{
    element object;
    string s1, s2;
    cout << "Enter the first string :";
    cin >> s1;
    cout << "Enter the second string :";
    cin >> s2;
    object.set(s1, s2);
    object.len(s1);
    object.cat();

    return 0;
}
```

24.

```
// Finding length of a string without the use of a predefined function
#include <iostream>
#include <string>
using namespace std;
class element
{
private:
    string str;

public:
    void set(string s)
    {
        str = s;
    }
    void len()
    {
        int l;
        for (int i = 0; str[i] != '\0'; i++)
        {
            l++;
        }
        cout << "The length is : " << l;
    }
};
int main()
{
    element object;
    string s;
    cout << "Enter a string :";
    cin >> s;
    object.set(s);
    object.len();

    return 0;
}
```

25.

```
//Reversal of a string without the use of a predefined function
#include <iostream>
#include <string>
using namespace std;
class element
{
private:
    string str;
    int length;

public:
    void set(string s)
    {
        str = s;
    }
    void len()
    {
        int l;
        for (int i = 0; str[i] != '\0'; i++)
        {
            l++;
        }
        length = l;
        cout << "The length is : " << length;
    }
    void rev()
    {
        int i = 0;
        string str_rev;
        for (int a = length; a != 0; a--)
        {
            str_rev[i] = str[a - 1];
            i++;
        }
        str_rev[length] = '\0';
        cout << endl << "The reversed String is :";
        for (int n = 0; str_rev[n] != '\0'; n++)
        {
            cout << str_rev[n];
        }
    }
};
int main()
{
    element object;
    string s;
    cout << "Enter a string :";
    cin >> s;
    object.set(s);
    object.len();
    object.rev();
    return 0;
}
```

26.

```
// Sum of Two numbers
#include<iostream>
using namespace std;
class element {
    private:
        int num1;
        int num2;
    public:
        void set(int n,int m){
            num1=n;
            num2=m;
        }
        void sum(){
            int sum = num1 + num2;
            cout << "The Sum is " << sum;
        }
};
int main(){
element object;
int n,m;
cout << "Enter the first number:" ;
cin >> n;
cout << "Enter the second number:" ;
cin >> m;
object.set(n,m);
object.sum();
return 0;
}
```

27.

```
// Finding the surface area and volume of a cuboid
#include<iostream>
using namespace std;
class element {
    private:
        int l,b,h;
    public:
        void set(int m,int n,int o){
            l=m;
            b=n;
            h=o;
        }
        void area(){
            int area=l*b*h;
            cout<<"The area is " << area;
        }
        void surface_area(){
            int surface_area=(2*l*b + 2*l*h + 2*b*h);
            cout<<"The surface area is " << surface_area;
        }

};
int main(){
element object;
int m,n,o;
cout << "Enter the first side of cuboid :" ;
cin >> m;
cout << "Enter the first side of cuboid :" ;
cin >> n;
cout << "Enter the first side of cuboid :" ;
cin >> o;
object.set(m,n,o);
object.area();
object.surface_area();

return 0;
}
```

28.

```
// Identifying the type of triangle
#include<iostream>
using namespace std;
class element {
    private:
        int a,b,c;
    public:
        void set(int m, int n, int o){
            b=n;
            a=m;
            c=o;
    }
        void check(){
            if(a==b && b==c && c==a)
            {
                printf("The triangle is Equilateral");
            }
            else if((a==c && a!=b)|| (a==b && a!=c)|| (b==c && b!=a))
            {
                printf("The triangle is isosceles");
            }
            else
            {
                printf("The triangle is Scalene");
            }
        }

};
int main(){
element object;
int m,n,o;
cout << "Enter the First Second and Third side of the triangle in cm :" ;
cin >> m >> n >> o;
object.set(m,n,o);
object.check();
return 0;
}
```

an account balance, then display appropriate message and do not carry out withdraw.

```
/*Program-99: Code to define and use an account type class */
#include<iostream.h>
class account { int acc;            // Account Number declared as private
                char name[30];      // Account holder name declared as private
                float balance;      // Account balance declared as private
                public:
                  void setData()                     // setData() function defined
                    {cout << "\nEnter account number: ";    cin >> acc;
                     cout << "\nEnter Name of customer: ";
                     fflush(stdin);
                     gets(name);
                     cout << "\nEnter starting balance: ";  cin >> balance;
                    }
                  void display() const               // display() function defined
                    { cout << "\nAccount No.: " << acc;
                      cout << "\nCustomer Name: " << name;
                      cout << "\nAccount balance: Rs." << balance;
                    }
                  void deposit( float amt )          // deposit() function defined
                    { balance = balance + amt;       // balance gets incremented by amt
                      cout << "\nBalance after deposit = Rs." << balance;
                    }
                  float getBalance() const           // getBalance() function defined
                    { return balance; }
                  void withdraw( float amt )         // withdraw() function defined
                    { if (balance >= amt)            // if balance is more than amount
                          {balance = balance - amt; // balance gets decreased by amt
                           cout << "\nBalance after withdrawal = Rs." << balance;
                          }
                      else
                          cout << "\nWithdrawal not possible. Balance = Rs." << balance;
                    }
              };
int main()
 {account a;                //account type object declared
  a.setData();              //data set for account a
  a.display();              //data displayed for account a
  a.deposit(500.00);        //amount 500.00 deposited to account a
  a.deposit(300.00);        //amount 300.00 deposited to account a
  a.withdraw(600.00);       //amount 600.00 withdrawn from account a
  a.withdraw(2000.00);      //amount 2000.00 withdrawn from account a
  return 0;
 }
```

Imp

```
/*Program-100: Code to define and use an employee type class */
#include<iostream.h>
class employee {int empID;                          // Employee ID declared as private
                char name[30];                      // Employee name declared as private
                float basic;                        // basic salary declared as private
                float totalSalary()          // totalSalary() function defined as private
                   {float x, total;          // Local variables to calculate total salary
                    x = basic + 0.4*basic + 0.2*basic;  //Basic + DA + HRA added
                    total = x - 0.06*x;             //Provident fund subtracted
                    return total;                   //Total calculated salary returned
                   }
                public:
                  void setData()                    // setData() function defined
                    {cout << "\nEnter employee ID: "; cin >> empID;
                     cout << "\nEnter Name of employee: "; fflush(stdin); gets(name);
                     cout << "\nEnter basic salary: "; cin >> basic;
                    }
                  void display()                    // display() function defined
                   {  cout << "\nEmployee ID: " << empID << ", Name: " << name;
                      cout << "\nTotal monthly salary: Rs." << totalSalary();
                   }
               };
int main()
 {employee e;       //employee type object declared
  e.setData();      //employee data of employee e set
  e.display();      //employee data of employee e displayed
  return 0;
 }
```