No 1 covert temp in farenheit

```
#include <iostream>
using namespace std;
int main()
{
	float far,cent;
	cout<<"Enter temperature in Farenheit: ";
	cin>>far;
	cent=(5.0/9.0)*(far-32);
	cout<<cent<<" degree Celsius";
	return 0;
}
```

No 2 check whether postive or neg using ternary

```
#include <iostream>
using namespace std;
int main()
{
	int num;
	cout<<"Enter the number: ";
	cin>>num;
	num>=0?cout<<"Positive number":cout<<"Negative number";
	return 0;
}
```

No 3 trigle is valid or note

```
#include <iostream>
using namespace std;
int main()
{
	int A1,A2,A3;
	cout<<"Enter degree of Angles: ";
	cin>>A1>>A2>>A3;
	if((A1+A2+A3)==180)
		cout<<"The triangle is valid";
	else
		cout<<"The triangle is invalid";
	return 0;
}
```

No 4 determine type input character

```
#include <iostream>
using namespace std;
int main()
{
	char ch;
	cout<<"Enter the character: ";
	cin>>ch;
```

```
	if(int(ch)>=65&&int(ch)<=90)
		cout<<"Capital Alphabet";
	else if(int(ch)>=97&&int(ch)<=122)
		cout<<"Small Alphabet";
	else if(int(ch)>=48&&int(ch)<=57)
		cout<<"Number";
	else if(int(ch)>=0&&int(ch)<=47||int(ch)>=58&&int(ch)<=64||int(ch)>=91&&int(ch)<=96||
int(ch)>=123&&int(ch)<=127)
		cout<<"Special Character";
	else
		cout<<"Not in ASCII range";
	return 0;
}
```

No 5 area of triagle

```
#include <iostream>
#include <cmath>
using namespace std;
int main()
{
	int a,b,c,s,A;
	cout<<"Enter the length of 3 sides of triangle: ";
	cin>>a>>b>>c;
	s=(a+b+c)/2;
	A=sqrt(s*(s-a)*(s-b)*(s-c));
	cout<<"Area of the triangle is: "<<A;
	return 0;
}
```

No 6 generate prime number

```
#include <iostream>
using namespace std;
void prime(int a, int b)
{
	int i,j,flag;
	for(i=a;i<=b;i++)
	{
		flag=1;
		for(j=2;j<=i/2;j++)
		{
			if(i%j==0)
			{
				flag=0;
				break;
			}
		}
		if(i==1)
			flag=0;
		if(flag==1)
			cout<<i<<"\t";
```

```
        }
}
int main()
{
        int a,b;
        cout<<"Enter range: ";
        cin>>a>>b;
        cout<<"Prime numbers in the range "<<a<<" to "<<b<<"\n";
        prime(a,b);
        return 0;
}
```

No 7 powe of a number

```
#include <iostream>
using namespace std;
double power(double n, int p=2)
{
        double pow=1;
        int i=1;
        while(i<=p)
        {
                pow=pow*n;
                i++;
        }
        return pow;
}
int main()
{
        double pow,n;
        int p;
        cout<<"Enter the number: ";
        cin>>n;
        cout<<"Enter the power: ";
        cin>>p;
        pow=power(n,p);
        cout<<"Result: "<<pow;
        return 0;
}
```

No 8 biggest and smallest in an array

```
#include <iostream>
using namespace std;
int main()
{
        int a[100],n,i,big,small;
        cout<<"Enter the number of elements: ";
        cin>>n;
        cout<<"Enter the elements: ";
```

```
	for(i=0;i<n;i++)
		cin>>a[i];
	big=a[0];
	small=a[0];
	for(i=0;i<n;i++)
	{
		if(a[i]>big)
			big=a[i];
		if(a[i]<small)
			small=a[i];
	}
	cout<<"Biggest number is: "<<big;
	cout<<"\nSmallest number is "<<small;
	return 0;
}
```

No 9 multipy two MATRIX

```
#include <iostream>
using namespace std;
int main()
{
	int a[10][10],b[10][10],i,j,k,s,m1,n1,m2,n2,mul[10][10];
	cout<<"Enter the order of matrix 1: ";
	cin>>m1>>n1;
	cout<<"Enter the order of matrix 2: ";
	cin>>m2>>n2;
	if(n1!=m2)
		cout<<"Matrix multiplication not possible!";
	else
	{
		cout<<"Enter the elements of matrix 1: ";
		for(i=0;i<m1;i++)
			for(j=0;j<n1;j++)
				cin>>a[i][j];
		cout<<"Enter the elements of matrix 2: ";
		for(i=0;i<m2;i++)
			for(j=0;j<n2;j++)
				cin>>b[i][j];
		for(i=0;i<m1;i++)
			for(k=0;k<n2;k++)
			{
				s=0;
				for(j=0;j<n1;j++)
				{
					s=s+(a[i][j]*b[j][k]);
					mul[i][k]=s;
				}
			}
		cout<<"Matrix 1\n";
		for(i=0;i<m1;i++)
		{
```

```
                        for(j=0;j<n1;j++)
                                cout<<a[i][j]<<"\t";
                        cout<<"\n";
                }
                cout<<"Matrix 2\n";
                for(i=0;i<m2;i++)
                {
                        for(j=0;j<n2;j++)
                                cout<<b[i][j]<<"\t";
                        cout<<"\n";
                }
                cout<<"Multiplied Matrix\n";
                for(i=0;i<m1;i++)
                {
                        for(k=0;k<n2;k++)
                                cout<<mul[i][k]<<"\t";
                        cout<<"\n";
                }
        }
        return 0;
}
```

No 10 passed student storing the details

```
#include <iostream>
using namespace std;
struct student
{
        int rno,fail;
        char name[30];
        float m1,m2,m3;
};
int main()
{
        student s[25];
        int i,n;
        cout<<"Enter number of student details: ";
        cin>>n;
        for(i=0;i<n;i++)
        {
                cout<<"Student detail "<<i+1<<"\n";
                cout<<"Enter roll no: ";
                cin>>s[i].rno;
                cout<<"Enter name: ";
                cin>>s[i].name;
                cout<<"Enter marks for 3 subjects: ";
                cin>>s[i].m1>>s[i].m2>>s[i].m3;
                s[i].fail=((s[i].m1<40?1:0)+(s[i].m2<40?1:0)+(s[i].m3<40?1:0));
        }
        cout<<"List of failed students\n";
        for(i=0;i<n;i++)
                if(s[i].fail>1)
```

```
                cout<<s[i].name<<"\n";
        return 0;
}
```

No 11 check whether area of rectangle are same

```
#include <iostream>
using namespace std;
class rectangle
{
        float length, width;
    public:
        void setlength(float l)
        {
                length=l;
        }
        void setwidth(float w)
        {
                width=w;
        }
        float perimeter()
        {
                return (length+width)*2;
        }
        float area()
        {
                return length*width;
        }
        void show()
        {
                cout<<"Length: "<<length<<endl;
                cout<<"Width: "<<width<<endl;
        }
        int samearea(rectangle r2)
        {
                float A1=area();
                float A2=r2.area();
                if(A1==A2)
                        return 1;
                else
                        return 0;
        }
};
int main()
{
        rectangle r1;
        rectangle r2;
        r1.setlength(5);
        r1.setwidth(2.5);
        r2.setlength(5);
        r2.setwidth(18.9);
```

```
	r1.show();
	r2.show();
	if(r1.samearea(r2)==1)
		cout<<"\nThey have same area";
	else
		cout<<"\nThey do not have same area";
	r1.setlength(15);
	r2.setwidth(6.3);
	if(r1.samearea(r2)==1)
		cout<<"\nThey have same area";
	else
		cout<<"\nThey do not have same area";
	return 0;
}
```

No 12 ADD to complete number using class

```
#include <iostream>
using namespace std;
class complex
{
		float real,imag;
	public:
		void set(float a, float b)
		{
			real=a;
			imag=b;
		}
		void disp()
		{
			cout<<real<<" + i"<<imag<<endl;
		}
		complex sum(complex c)
		{
			complex s;
			s.real=real+c.real;
			s.imag=imag+c.imag;
			return s;
		}
};
int main()
{
	complex c1,c2,c3;
	c1.set(1,5);
	c2.set(4,7);
	c3=c1.sum(c2);
	cout<<"Complex Number 1: ";
	c1.disp();
	cout<<"Complex Number 2: ";
	c2.disp();
	cout<<"Complex Number Sum: ";
```

```
	c3.disp();
	return 0;
}
```

No 13 MAINTAIN and UPDATE cashregister

```
#include <iostream>
using namespace std;
class cashRegister
{
	int cashOnHand;
	public:
		cashRegister()
		{
			cashOnHand=500;
		}
		cashRegister(int cash)
		{
			cashOnHand=cash;
		}
		int getCurrentBalance()
		{
			return cashOnHand;
		}
		void acceptAmount(int cashIn)
		{
			cashOnHand=cashOnHand+cashIn;
			cout<<"Cash Accepted\nCurrent balance: "<<getCurrentBalance();
		}
};
int main()
{
	int cashIn;
	cashRegister csh1;
	cout<<"Current balance: ";
	cout<<csh1.getCurrentBalance()<<endl;
	cout<<"Enter amount to deposit: ";
	cin>>cashIn;
	csh1.acceptAmount(cashIn);
	return 0;
}
```

No 14 calc rectangle , TRIANGLE SPHERE USING FUN

```
#include <iostream>
#include <cmath>
using namespace std;
float area(float ra, float rb)
{
	return ra*rb;
}
float area(float ta, float tb, float tc)
```

```
{
        float s;
        s=(ta+tb+tc)/2;
        return sqrt(s*(s-ta)*(s-tb)*(s-tc));
}
float area(float cr)
{
        return 4.0*3.14*cr*cr;
}
int main()
{
        float ra,rb,ta,tb,tc,cr;
        cout<<"Enter the length and breadth of the rectangle: ";
        cin>>ra>>rb;
        cout<<"Area is "<<area(ra,rb);
        cout<<"\nEnter the sides of the triangle: ";
        cin>>ta>>tb>>tc;
        cout<<"Area is "<<area(ta,tb,tc);
        cout<<"\nEnter the radius of the sphere: ";
        cin>>cr;
        cout<<"Area is "<<area(cr);
        return 0;
}
```

No 15 CAL cube of a number using inline

```
#include <iostream>
using namespace std;
inline float cubecalc(float c)
{
        return c*c*c;
}
int main()
{
        int n;
        cout<<"Enter the number: ";
        cin>>n;
        cout<<"Cube of "<<n<<" is "<<cubecalc(n);
        return 0;
}
```

No 16 add two complex num using friend fun

```
#include <iostream>
using namespace std;
class complex
{
        float real;
        float imag;
        public:
                void input()
                {
```

```
                cout<<"Enter the real part: ";
                cin>>real;
                cout<<"Enter the imaginary part: ";
                cin>>imag;
        }
        void output()
        {
                cout<<real<<" + i"<<imag;
        }
        friend complex comsum(complex,complex);
};
complex comsum(complex c1, complex c2)
{
        complex result;
        result.real=c1.real+c2.real;
        result.imag=c1.imag+c2.imag;
        return result;
}
int main()
{
        complex c1,c2,c3;
        cout<<"Enter 1st complex number\n";
        c1.input();
        c1.output();
        cout<<"\nEnter 2nd complex number\n";
        c2.input();
        c2.output();
        c3=comsum(c1,c2);
        cout<<"\nsum of the complex numbers: ";
        c3.output();
        return 0;
}
```

No 17 count the nummber of object created

```
#include <iostream>
using namespace std;
class Student
{
        protected:
           char name[50];
           int rollNumber;
        public:
           void setStudentInfo()
                {
              cout << "Enter Student Name: ";
              cin >> name;
              cout << "Enter Roll Number: ";
              cin >> rollNumber;
           }
};
class Mark : public Student
```

```
{
        private:
            int marks;
        public:
            void setMark()
                {
                cout << "Enter Marks: ";
                cin >> marks;
            }
            char calculateGrade()
                {
                if (marks >= 90)
                    return 'A';
                else if (marks >= 80)
                    return 'B';
                else if (marks >= 70)
                    return 'C';
                else if (marks >= 60)
                    return 'D';
                else
                    return 'F';
            }
            void displayStudentInfo()
                {
                cout << "Student Name: " << name << endl;
                cout << "Roll Number: " << rollNumber << endl;
                cout << "Marks: " << marks << endl;
                cout << "Grade: " << calculateGrade() << endl;
            }
};
int main()
{
    Mark studentMark;
    studentMark.setStudentInfo();
    studentMark.setMark();
    studentMark.displayStudentInfo();
    return 0;
}
```

No 18 CALCU THE GRADE OF STUDENT

```
#include <iostream>
using namespace std;
class Student
{
        protected:
            char name[50];
            int rollNumber;
        public:
            void setStudentInfo()
                {
                cout << "Enter Student Name: ";
```

```
            cin >> name;
            cout << "Enter Roll Number: ";
            cin >> rollNumber;
        }
};
class Mark : public Student
{
    private:
        int marks;
    public:
        void setMark()
            {
            cout << "Enter Marks: ";
            cin >> marks;
        }
        char calculateGrade()
            {
            if (marks >= 90)
                return 'A';
            else if (marks >= 80)
                return 'B';
            else if (marks >= 70)
                return 'C';
            else if (marks >= 60)
                return 'D';
            else
                return 'F';
        }
        void displayStudentInfo()
            {
            cout << "Student Name: " << name << endl;
            cout << "Roll Number: " << rollNumber << endl;
            cout << "Marks: " << marks << endl;
            cout << "Grade: " << calculateGrade() << endl;
        }
};
int main()
{
   Mark studentMark;
   studentMark.setStudentInfo();
   studentMark.setMark();
   studentMark.displayStudentInfo();
   return 0;
}
```

No 19

```
#include <iostream>
using namespace std;
class student
{
    protected:
```

```
    int roll_no;
  public:
    void get_roll_no()
        {
      cout << "Enter Roll Number: ";
      cin >> roll_no;
    }
};
class test : public student
{
  protected:
    float subject1_mark;
    float subject2_mark;
  public:
    void get_marks()
        {
      cout << "Enter Marks for Subject 1: ";
      cin >> subject1_mark;
      cout << "Enter Marks for Subject 2: ";
      cin >> subject2_mark;
    }
};
class result : public test
{
    float total_marks;
  public:
    void calculate_total()
        {
      total_marks = subject1_mark + subject2_mark;
    }
    void display_details()
        {
      cout << "\nStudent Details:\n";
      cout << "Roll Number: " << roll_no << endl;
      cout << "Marks in Subject 1: " << subject1_mark << endl;
      cout << "Marks in Subject 2: " << subject2_mark << endl;
      cout << "Total Marks: " << total_marks << endl;
    }
};
int main()
{
   result student_result;
   student_result.get_roll_no();
   student_result.get_marks();
   student_result.calculate_total();
   student_result.display_details();
   return 0;
}
```

MAMMALS

No 20

```
#include <iostream>
```

```
using namespace std;
class Mammals
{
        public:
           void displayMammal()
                {
              cout << "I am a mammal." << endl;
           }
};
class MarineAnimals
{
        public:
           void displayMarineAnimal()
                {
              cout << "I am a marine animal." << endl;
           }
};
class BlueWhale : public Mammals, public MarineAnimals
{
        public:
           void displayBlueWhale()
                {
              cout << "I belong to both categories: Mammals as well as Marine Animals." << endl;
           }
};
int main()
{
   Mammals mammalObj;
   MarineAnimals marineAnimalObj;
   BlueWhale blueWhaleObj;
   cout << "Calling function of Mammals by the object of Mammals:" << endl;
   mammalObj.displayMammal();
   cout << endl;
   cout << "Calling function of MarineAnimal by the object of MarineAnimal:" << endl;
   marineAnimalObj.displayMarineAnimal();
   cout << endl;
   cout << "Calling function of BlueWhale by the object of BlueWhale:" << endl;
   blueWhaleObj.displayBlueWhale();
   cout << endl;
   cout << "Calling function of each parent by the object of BlueWhale:" << endl;
   blueWhaleObj.displayMammal();
   blueWhaleObj.displayMarineAnimal();
   return 0;
}
```

No 21

```
#include <iostream>
using namespace std;
class Employee
{
        protected:
```

```
        int empno;
        char empname[50];
    public:
        void inputEmployeeDetails()
            {
            cout << "Enter Employee Number: ";
            cin >> empno;
            cout << "Enter Employee Name: ";
            cin >> empname;
        }
        void outputEmployeeDetails()
            {
            cout << "Employee Number: " << empno << endl;
            cout << "Employee Name: " << empname << endl;
        }
};
class Manager : public Employee
{
    protected:
        char position[50];
        double dues;
    public:
        void inputManagerDetails()
            {
            inputEmployeeDetails();
            cout << "Enter Manager Position: ";
            cin >> position;
            cout << "Enter Dues: ";
            cin >> dues;
        }
        void outputManagerDetails()
            {
            outputEmployeeDetails();
            cout << "Manager Position: " << position << endl;
            cout << "Dues: " << dues << endl;
        }
};
class Laborer : public Employee
{
    protected:
        double allowances;
    public:
        void inputLaborerDetails()
            {
            inputEmployeeDetails();
            cout << "Enter Allowances: ";
            cin >> allowances;
        }
        void outputLaborerDetails()
            {
            outputEmployeeDetails();
            cout << "Allowances: " << allowances << endl;
```

```
        }
};
int main()
{
   Manager managerObj;
   Laborer laborerObj;
   cout << "Enter Manager Details:" << endl;
   managerObj.inputManagerDetails();
   cout << "\nManager Details:" << endl;
   managerObj.outputManagerDetails();
   cout << "\n\nEnter Laborer Details:" << endl;
   laborerObj.inputLaborerDetails();
   cout << "\nLaborer Details:" << endl;
   laborerObj.outputLaborerDetails();
   return 0;
}
```

No 22

```
#include <iostream>
using namespace std;
class student
{
        protected:
           int roll_no;
        public:
           void get_roll_no()
                {
              cout << "Enter Roll Number: ";
              cin >> roll_no;
           }
};
class test : public student
{
        protected:
           float subject1_mark;
           float subject2_mark;
        public:
           void get_marks()
                {
              cout << "Enter Marks for Subject 1: ";
              cin >> subject1_mark;

              cout << "Enter Marks for Subject 2: ";
              cin >> subject2_mark;
           }
};
class sports
{
        protected:
           float sports_weightage;
```

```
        public:
            void get_sports_weightage()
                {
                cout << "Enter Sports Weightage: ";
                cin >> sports_weightage;
            }
};

class result : public test, public sports
{
            float total_marks;
        public:
            void calculate_total()
                {
                total_marks = subject1_mark + subject2_mark + sports_weightage;
            }
            void display_details()
                {
                cout << "\nStudent Details:\n";
                cout << "Roll Number: " << roll_no << endl;
                cout << "Marks in Subject 1: " << subject1_mark << endl;
                cout << "Marks in Subject 2: " << subject2_mark << endl;
                cout << "Sports Weightage: " << sports_weightage << endl;
                cout << "Total Marks: " << total_marks << endl;
            }
};
int main() {
    result student_result;
    student_result.get_roll_no();
    student_result.get_marks();
    student_result.get_sports_weightage();
    student_result.calculate_total();
    student_result.display_details();
    return 0;
}
```

No 23

```
#include <iostream>
using namespace std;
class student
{
        protected:
            int roll_no;
            char name[50];
        public:
            void get_student_details()
                {
                cout << "Enter Roll Number: ";
                cin >> roll_no;
                cout << "Enter Name: ";
                cin >> name;
```

```
        }
};
class test : virtual public student
{
        protected:
          float subject1_mark;
          float subject2_mark;
        public:
          void get_test_details()
                {
            cout << "Enter Marks for Subject 1: ";
            cin >> subject1_mark;
            cout << "Enter Marks for Subject 2: ";
            cin >> subject2_mark;
          }
};
class sports : virtual public student
{
        protected:
          float sports_weightage;
        public:
          void get_sports_details()
                {
            cout << "Enter Sports Weightage: ";
            cin >> sports_weightage;
          }
};
class result : public test, public sports
{
          float total_marks;
        public:
          void calculate_total()
                {
            total_marks = subject1_mark + subject2_mark + sports_weightage;
          }
          void display_details()
                {
            cout << "\nStudent Details:\n";
            cout << "Roll Number: " << roll_no << endl;
            cout << "Name: " << name << endl;
            cout << "Marks in Subject 1: " << subject1_mark << endl;
            cout << "Marks in Subject 2: " << subject2_mark << endl;
            cout << "Sports Weightage: " << sports_weightage << endl;
            cout << "Total Marks: " << total_marks << endl;
          }
};
int main()
{
   result student_result;
   student_result.get_student_details();
   student_result.get_test_details();
   student_result.get_sports_details();
```

```
   student_result.calculate_total();
   student_result.display_details();
   return 0;
}
```

No 24

```
#include<iostream>
using namespace std;
int main()
{
	int a,b;
	cout<<"Enter a and b: ";
	cin>>a>>b;
	int x = a-b;
	try
	{
		if(x!=0)
		{
			cout<<"Result(a/x) = "<<a/x<<"\n";
		}
		else
		{
			cout<<"\n Division by zero is not possible";
			throw(x);
		}
	}
	catch(int i)
	{
		cout<<"\nException caught "<<i;
	}
	return 0;
}
```

No 25

```
#include <iostream>
using namespace std;
class shape
{
	protected:
	   double dimension1, dimension2;
	public:
	   void get_data()
		{
	      cout << "Enter Dimension 1: ";
	      cin >> dimension1;
	      cout << "Enter Dimension 2: ";
	      cin >> dimension2;
	   }
	   virtual void display_area()
		{
```

```
            cout << "Area: " << dimension1 << " * " << dimension2 << " = " << (dimension1 *
dimension2) << endl;
        }
};
class triangle : public shape
{
    public:
        void display_area()
            {
            cout << "Area of triangle: 0.5 * " << dimension1 << " * " << dimension2 << " = " <<
(0.5 * dimension1 * dimension2) << endl;
        }
};
class rectangle : public shape
{
    public:
        void display_area()
            {
            cout << "Area of rectangle: " << dimension1 << " * " << dimension2 << " = " <<
(dimension1 * dimension2) << endl;
        }
};
int main()
{
   triangle triangle_obj;
   rectangle rectangle_obj;
   cout << "Enter dimensions for triangle:" << endl;
   triangle_obj.get_data();
   triangle_obj.display_area();
   cout << "\nEnter dimensions for rectangle:" << endl;
   rectangle_obj.get_data();
   rectangle_obj.display_area();
   return 0;
}
```