EXP 1.1 Operator Precedence

```
#include <iostream.h>
Int main() {
  int a = 20, b = 10, c = 15,d = 5, e;
   e = (a + b) * c / d;
  cout << "Value of (a + b) * c / d is :" << e << endl ;
  e = ((a + b) * c) / d;
  cout << "Value of ((a + b) * c) / d is  :" << e << endl ;
  e = (a + b) * (c / d);
  cout << "Value of (a + b) * (c / d) is  :" << e << endl ;
  e = a + (b * c) / d;
  cout << "Value of a + (b * c) / d is  :" << e << endl ;
  return 0;
}
```

Output

```
Value of (a + b) * c / d is :90
Value of ((a + b) * c) / d is  :90
Value of (a + b) * (c / d) is  :90
Value of a + (b * c) / d is  :50
```

EXP 1.2 Roots of a Quadratic Equation

```
#include <iostream.h>
#include <math.h>
int main()
```

```
{
   float a, b, c, x1, x2;
   cout << "Enter coefficients a, b and c: ";
   cin >> a >> b >> c;
      x1 = (-b + sqrt((b*b)-(4*a*c))) / (2*a);
      x2 = (-b - sqrt((b*b)-(4*a*c))) / (2*a);
      cout << "Roots are real and different." << endl;
      cout << "x1 = " << x1 << endl;
      cout << "x2 = " << x2 << endl;
   return 0;
}
```

Output

```
Enter coefficients a, b and c: 4
5
1
Roots are real and different.
x1 = -0.25
x2 = -1
```

EXP 2.1 Check if a year is leap year or not

```
#include <iostream.h>
int main()
{
   int year;
```

```
    cout << "Enter a year: ";
    cin >> year;
    if (year % 4 == 0)
    {
        if (year % 100 == 0)
        {
            if (year % 400 == 0)
                cout << year << " is a leap year.";
            else
                cout << year << " is not a leap year.";
        }
        else
            cout << year << " is a leap year.";
    }
    else
        cout << year << " is not a leap year.";

    return 0;
}
```

Output

Enter a year: 2014

2014 is not a leap year.

EXP 2.2 Display Armstrong Numbers Between two Integers

```
#include <iostream.h>
int main()
{
  int num1, num2, i, num, digit, sum;
  cout << "Enter first number: ";
  cin >> num1;

  cout << "Enter second number: ";
  cin >> num2;

  cout << "Armstrong numbers between " << num1 << " and " << num2 << " are: " << endl;
  for(i = num1; i <= num2; i++)
  {
      sum = 0;
      num = i;

      while(num > 0)
      {
         digit = num % 10;
         sum = sum + digit * digit * digit;
             num=num/10;
      }

      if(sum == i)
```

```
      {
         cout << i << endl;
      }
  }

  return 0;
}
```

Output

```
Enter first number: 100
Enter second number: 400
Armstrong numbers between 100 and 400 are:
153
370
371
```

EXP 2.3 Check Palindrome Number

```
#include <iostream.h>
int main()
{
    int n, num, digit, rev = 0;
    cout << "Enter a positive number: ";
    cin >> num;

    n = num;
```

```
    do
    {
       digit = num % 10;
       rev = (rev * 10) + digit;
       num = num / 10;
    } while (num != 0);

    cout << " The reverse of the number is: " << rev << endl;

    if (n == rev)
       cout << " The number is a palindrome";
    else
       cout << " The number is not a palindrome";

   return 0;
}
```

Output

```
Enter a positive number: 12321
The reverse of the number is: 12321
The number is a palindrome
```

EXP 3.1 Display Multiplication table up to 10

```
#include <iostream.h>
```

```
int main()
{
   int n;

   cout << "Enter a positive integer: ";
   cin >> n;

   for (int i = 1; i <= 10; ++i) {
      cout << n << " * " << i << " = " << n * i << endl;
   }

   return 0;
}
```

Output

```
Enter an integer: 5
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
5 * 4 = 20
5 * 5 = 25
5 * 6 = 30
5 * 7 = 35
5 * 8 = 40
5 * 9 = 45
```

5 * 10 = 50

EXP 3.2 Displaying numbers 100 to 1

```
#include <iostream.h>
int main()
{
   int n;

   for (int i = 100; i >= 1; i--) {
      cout << i << endl;
   }

   return 0;
}
```

EXP 3.3 Print Floyd's Triangle.

1

2 3

4 5 6

7 8 9 10

```
#include <iostream.h>
int main()
{
```

```
    int rows, number = 1;

    cout << "Enter number of rows: ";
    cin >> rows;

    for(int i = 1; i <= rows; i++)
    {
        for(int j = 1; j <= i; j++)
        {
            cout << number << " ";
            number++;
        }

        cout << endl;
    }

    return 0;
}
```

EXP 3.4 Program to print full pyramid using *

```
     *
    * *
   * * *
  * * * *
* * * * *
```

```
#include<iostream.h>
#include<conio.h>
int main()
{
	int r,c,k,rows;
	clrscr();
	cout<<"Enter number of rows: \n";
	cin>>rows;

	cout<<endl;
	for(r=1;r<=rows;r++)
	{

	    for(k=1;k<=rows-r;k++)
	    {
			cout<<"  ";
	    }
	    for(c=1;c<=r;c++)
	    {
			cout<<"  * ";
	    }
	    cout<<endl;
	}
	getch();
	return 0;
```

}

EXP 4.1 program to find median of two sorted arrays

```
#include <iostream.h>
#include<conio.h>
   void main()
   {

         int a1[50], a2[50], a3[100], m, n, i, j, k = 0;
         clrscr();
         cout<<"\n Enter size of array Array 1: ";
         cin>>m;

         cout<<"\n Enter sorted elements of array 1: \n";
         for (i = 0; i < m; i++)
         {
            cin>>a1[i];
         }

         cout<<"\n Enter size of array 2: ";
         cin>>n;

         cout<<"\n Enter sorted elements of array 2: \n";
         for (i = 0; i < n; i++)
         {
```

```
    cin>>a2[i];
}

i = 0;
j = 0;

while (i < m && j < n)
{
    if (a1[i] < a2[j])
    {
            a3[k] = a1[i];
            i++;
    }

    else
    {
            a3[k] = a2[j];
            j++;
    }
    k++;
}

if (i >= m)
{
    while (j < n)
```

```
    {
        a3[k] = a2[j];
        j++;
        k++;
    }
}

if (j >= n)
{
    while (i < m)
    {
        a3[k] = a1[i];
        i++;
        k++;
    }
}

cout<<"\n After merging: \n";
for (i = 0; i < m + n; i++)
{
    cout<<" "<<a3[i];
}

int len=m+n;
int mid = len/ 2;
```

```
        float median;
   if (len % 2)
   {
        median = a3[mid] ;
        cout << "\nThe median is: " << median << endl;
   }

   else
   {
        median = (a3[mid]+a3[mid-1])/2.0;
        cout << "\nThe median is: " << median << endl;
   }
   getch();

   }
```

EXP 4.2 program to find two repeating elements in a given array

```
#include<iostream.h>
#include<conio.h>

 void main()
 {
  int i,arr[20],j,no;
  clrscr();
  cout<<"Enter Size of array: ";
```

```
 cin>>no;
 cout<<"Enter any "<<no<<" num in array: ";
 for(i=0;i<no;i++)
 {
  cin>>arr[i];
 }
 cout<<"Dublicate Values are: ";
 for(i=0; i<no; i++)
  {
   for(j=i+1;j<no;j++)
   {
   if(arr[i]==arr[j])
   {
   cout<<"\n"<<arr[i];
   }
  }
  }
 getch();
}
```

Output

Enter Size of Array : 5

Enter any 5 elements in Array:

5 4 5 2 3

Duplicate Elements are:

EXP 4.3 program to find smallest and second smallest element in a given array

```
#include <iostream.h>
int main()
{
        int arr[50], n;
        cout<<"\n Enter number of elements in array: ";
         cin>>n;
cout<<"\n Enter array elements: ";
for(i=0; i<n; i++)
                cin>>arr[i];
        int small,next_small=INT_MAX;
        small = arr[0];

        for(int i=1;i<len;i++)
        {
                if(arr[i] < small)
                {
                        next_small = small;
                        small = arr[i];
                }
                else if(arr[i] < next_small and arr[i] > small)
                        next_small = arr[i];
        }
```

```
        cout << "Smallest and the second smallest numbers are respectively "<< small
<< " and " << next_small <<endl;


        return 0;
}
```

EXP 4.4 program to find missing number from array

```
#include <iostream.h>
int main()
{
  int n=5,i;
  int arr[4];
  cout<<"Enter sequential array elements: ";
  for(i=0; i<n; i++)
  {
    cin>>arr[i];
  }

  int temp;
  temp  = ((n+1)*(n+2))/2;  //sum of n elements is n(n+1)/2,
  for (i = 0; i<n; i++)    //here one no. is missing so we use (n+1)(n+2)/2
    temp -= arr[i];

  int missingNo =temp;
```

```
  cout<<"Missing Number is: "<<missingNo;
  return 0;
}
```

EXP6.1 Write a c++ program to declare a class book having data members book_name, author and price. Accept and display data for book having maximum price.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class Book
{
        int no_of_pages;
        char book_name[50];
        public:
        float price;
        void getdata()
        {

                        cout<<"Enter Book name:-> ";
                        cin>>book_name;
                        cout<<"Enter Book Price:->";
                        cin>>price;
                        cout<<"Enter No of pages:-> ";
                        cin>>no_of_pages;
```

```
        }
        void display()
        {

                cout<<"\nBook name:-> "<<book_name;
                cout<<"\nBook Price:->"<<price;
                cout<<"\nNo of pages:-> "<<no_of_pages;
        }

};

void main()
{
        Book b1,b2;
        clrscr();
        b1.getdata();
        b2.getdata();
        if(b1.price>b2.price)
                b1.display();
        else
                b2.display();

getch();

}
```

EXP 6.2 Write a c++ program to declare a class staff having data members name, salary, DA, HRA and calculate gross salary. Accept and display data for one staff.

Where DA=74.5% of basic, HRA=30% of basic, gross_salary= basic+DA+HRA

```
#include<iostream.h>
#include<conio.h>
class staff
{
        float basic,gross_sal;
    public:
        void accept()
        {
            cout<<"Enter Basic Salary of Staff: ";
            cin>>basic;
        }
        float display ();
};
float staff :: display()
{  float DA=74.5, HRA=30;
  gross_sal=(basic*DA/100)+(basic*HRA/100)+basic;
  return(gross_sal);
}
void main()
{
    clrscr();
```

```
        staff s1;
        s1.accept();
        cout<<"Gross Salary of Employee "<<s1.display();
        getch();
}
```

EXP7.1 Employee details having salary greater than 25000/-

```
#include<iostream.h>
#include<conio.h>
class Employee
{
        int emp_id;
        char emp_name[20];

public:
        float emp_salary;
        void accept()
        {
                cout<<"\nEnter Employee id=> ";
                cin>>emp_id;
                cout<<"Enter employee Name=> ";
                cin>>emp_name;
                cout<<"Enter employee salary=> ";
                cin>>emp_salary;
        }
```

```
        void display()
        {
                cout<<endl<<"Employee id=> "<<emp_id<<endl;
                cout<<"Employee NAme=> "<<emp_name<<endl;
                cout<<"Employee Salary=> "<<emp_salary;
        }

};

void main()
{
        int i;
        clrscr();
        Employee e[5];
        for(i=0;i<5;i++)
        {
                e[i].accept();
        }
        cout<<"Employees having Salary greater than 25000 are:\n";
        for(i=0;i<5;i++)
        {
                if(e[i].emp_salary>25000)
                        e[i].display();
        }
        getch();
```

```
}
```

EXP7.2 City and their population

```
#include<iostream.h>
#include<conio.h>
class City
{
        long int population;
        char city_name[20];

public:
        void accept()
        {
                cout<<"\nEnter City Name=> ";
                cin>>city_name;
                cout<<"Enter Population of the City=> ";
                cin>>population;
        }
        void display()
        {
                cout<<endl<<"City Name=> "<<city_name<<endl;
                cout<<"Population=> "<<population<<endl;
        }
```

```
};

void main()
{
        int i;
        clrscr();
        City c[10];
        for(i=0;i<10;i++)
        {
                c[i].accept();
        }
        cout<<"\nCity details are:\n";
        for(i=0;i<10;i++)
        {
                c[i].display();
        }
        getch();
}
```

EXP 8.1 Swap variables using friend function

```
#include<iostream.h>
#include<conio.h>
class swap
{
                int a,b;
```

```
        public:
                void getab()
                {
                        cout<<"\n\tEnter Value  of a:->";
                        cin>>a;
                        cout<<"Enter value of b:->";
                        cin>>b;

                }
                friend void swapping(swap s1);
};

void swapping(swap s1)
{  int t;
        t=s1.a;
        s1.a=s1.b;
        s1.b=t;
        cout<<"After Swapping a= "<<s1.a<<"\t b= "<<s1.b;
}
void main()
{
        clrscr();
        swap s;
        s.getab();
        swapping(s);
```

```
getch();


}
```

EXP 8.2

```
#include<iostream.h>
#include<conio.h>
class DM
{
        float distm;    //distance in meters
    public:
        DM()
        {
            cout<<"\n\tEnter ditance in meters:->";
            cin>>distm;
        }
        friend void totaldist(DM d1, DM d2);
};
void totaldist(DM d1, DB d2)
{
  float dist =d1.distm + d2.distm;
  cout<<"\nTotal Distance in Meters => "<<dist;
}
void main()
{
    clrscr();
```

```
        DM d1;
        DM d2
        totaldist(d1,d2);
getch();


}
```

EXP 10.1 Arithmetic operations on two numbers using constructor

```
#include<iostream.h>
#include<conio.h>
class Number
{
        int x,y;
public:
        Number()
        {
                x=15;
                y=5;
        }
        Number(int a,int b)
        {
                x=a;
                y=b;
        }
        void display()
        {
```

```
        cout<<"\nx= " <<x<<"\t"<<"y= "<<y;
        cout<<endl<<"Addition= "<<x+y<<endl;
        cout<<"Subtraction= "<<x-y<<endl;
        cout<<"Multiplication= "<<x*y<<endl;
        cout<<"Division= "<<x/y<<endl;
    }
    ~Number()
    {
        cout<<"\nMemory released";
    }
};

void main()
{
    clrscr();
    {
        Number n1;
        Number n2(45,15);
        cout<<"\nFor Object 1:\n";
        n1.display();
        cout<<"\nFor Object 2:\n";
        n2.display();
    }
    getch();
}
```

EXP10.2 Product Details using Overloaded Constructors

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
class Product
{
        int prod_id;
        float prod_price;
        char prod_name[25];
public:
        Product()
        {
                prod_id=001;
                prod_price=500;
                strcpy(prod_name,"Photo Frame");
        }
        Product(int i,float p,char n[25])
        {
                prod_id=i;
                prod_price=p;
                strcpy(prod_name,n);
        }
        Product(Product &pc)
        {
                prod_id=pc.prod_id;
```

```
            prod_price=pc.prod_price;
            strcpy(prod_name,pc.prod_name);
        }
        void displayprod()
        {
            cout<<"\nProduct ID= " <<prod_id<<"\n"<<"Product Price=
"<<prod_price;
            cout<<"\n"<<"Product Name= "<<prod_name<<endl;
        }
        ~Product()
        {
            cout<<"\nMemory released";
        }
};

void main()
{
        clrscr();
        {
            Product p1;
            Product p2(002,1000,"Fan");
            cout<<"\nFor Product 1:";
            p1.displayprod();
            cout<<"\nFor Product 2:";
            p2.displayprod();
```

```
		cout<<"\nFor Product 3:";
		Product p3(p1);
		p3.displayprod();
	}
	getch();
}
```

Exp11.1

```
#include<iostream.h>
#include<conio.h>
class student
{
	protected :
		int rollno;
		char name[25];
	public:
	  void get();
	  void put();
};
void student :: get()
{
	cout<<"\nEnter rollno: ";
	cin>>rollno;
	cout<<"Enter Name: ";
```

```
        cin>>name;
}
void student :: put()
{
        cout<<"\nRoll No: "<<rollno;
        cout<<"\nName: "<<name;
}
 class marks : public student
{
        protected:
                int m1,m2,m3,total;
                float average;
        public:
                void getmarks();
                void putmarks();
};
void marks :: getmarks()
{
        cout<<"Enter Marks of sub1: ";
        cin>>m1;
        cout<<"Enter Marks of sub2: ";
        cin>>m2;
        cout<<"Enter Marks of sub3: ";
        cin>>m3;
```

```
}
void marks :: putmarks()
{
        put();
        cout<<"\nMark obtained in sub1: "<<m1;
        cout<<"\nMarks obtained in sub2: "<<m2;
        cout<<"\nMarks obtained in Sub3: "<<m3;
        total=m1+m2+m3;
        average=total/3;
        cout<<"\nTotal Marks: "<<total;
        cout<<"\nAverage Marks: "<<average;

}
void main()
{
        marks S1;
        clrscr();

        S1.get();
        S1.getmarks();
        S1.putmarks();
        getch();

}
```

Exp11.2

```
#include<iostream.h>
#include<conio.h>
class Employee
{
	int emp_id;
	char emp_designation[20];
	char emp_name[25];
public:
	void accept()
	{
		cout<<"Enter Employee id: ";
		cin>>emp_id;
		cout<<"Enter employee Name: ";
		cin>>emp_name;
		cout<<"Enter employee Designation: ";
		cin>>emp_designation;
	}
	void display()
	{
		cout<<"\nEmployee id: "<<emp_id<<endl;
		cout<<"Employee Name: "<<emp_name<<endl;
		cout<<"Employee Designation: "<<emp_designation<<endl;
	}
};
```

```
class Salary:public Employee
{
        float basic,gross_sal;
    public:
        void accept_salary()
        {
            accept();
            cout<<"Enter Basic Salary of Employee: ";
            cin>>basic;
        }
        void display_salary();
};
void Salary :: display_salary()
{
    float DA=74.5, HRA=30;
    gross_sal=(basic*DA/100)+(basic*HRA/100)+basic;

    display();
    cout<<"Gross Salary: "<<gross_sal;
}
void main()
{
    clrscr();
    Salary s1;
```

```
        s1.accept_salary();
        cout<<"\nDetails of Employee "<<endl;
        s1.display_salary();
        getch();
}
```

//Exp 12.1

```
#include<iostream.h>
#include<conio.h>
class Person
{
        protected :
                int age;
                char name[25];
                char gender;
        public:
                void get_info();
                void put_info();
};
void Person :: get_info()
{
        cout<<"Enter Name:-\t";
        cin>>name;
        cout<<"Enter age:-\t";
        cin>>age;
```

```
        cout<<"Enter Gender:-\t";
        cin>>gender;
}
void Person :: put_info()
{
        cout<<"\nName:-\t"<<name;
        cout<<"\nAge:-\t"<<age;
        cout<<"\nGender:-\t"<<gender;
}
 class Employee : public Person
{
        protected:
                int emp_id;
                float salary;
                char company[25];
        public:
                void get_emp_details();
                void put_emp_details();
};
void Employee :: get_emp_details()
{
        cout<<"Enter Employee ID:-\t";
        cin>>emp_id;
        cout<<"Enter Company Name:-\t";
        cin>>company;
```

```
	cout<<"Enter Salary:-\t";
	cin>>salary;
}
void Employee :: put_emp_details()
{
	cout<<"\nEmployee Id:-\t"<<emp_id;
	cout<<"\nCompany Name:-\t"<<company;
	cout<<"\nSalary:-\t"<<salary;
}

class Programmer : public Employee
{
		int no_of_prog_lang_known;
	public:
		void accept()
		{
			cout<<"Enter no. of programming languages known:-\t";
			cin>>no_of_prog_lang_known;
		}
		void display();
};
void Programmer :: display()
{
	put_info();
	put_emp_details();
```

```
	cout<<"\nNo. of programming languages known:-
\t"<<no_of_prog_lang_known;
}

void main()
{
	clrscr();
	Programmer r1;
	r1.get_info();
	r1.get_emp_details();
	r1.accept();
	cout<<"\nDetails of programmer:\n";
	r1.display();
	getch();
}
```

//Exp 12.2

```
#include<iostream.h>
#include<conio.h>
class Car
{
	protected :
		char car_type[25];
	public:
		void get_car_type();
```

```
                void put_car_type();
};
void Car :: get_car_type()
{
        cout<<"Enter car_type:-\t";
        cin>>car_type;
}
void Car :: put_car_type()
{
        cout<<"\nCar Type:-\t"<<car_type;
}
 class Brand : public Car
{
        protected:
                int speed;
                char brand_name[25];
        public:
                void get_brand_details();
                void put_brand_details();
};
void Brand :: get_brand_details()
{
        cout<<"Enter Brand Name of car:-\t";
        cin>>brand_name;
        cout<<"Enter Speed:-\t";
```

```
	cin>>speed;
}
void Brand :: put_brand_details()
{
	cout<<"\nSpeed:-\t\t"<<speed;
	cout<<"\nBrand Name :-\t"<<brand_name;
}

class Model : public Brand
{
		char model_name[10];
		long float price;
	public:
		void get_model_details()
		{
			cout<<"Enter Model Name:-\t";
			cin>>model_name;
			cout<<"Enter Price:-\t";
			cin>>price;
		}
		void put_model_details();
};
void Model :: put_model_details()
{
	put_car_type();
```

```
        put_brand_details();
        cout<<"\nModel Name:-\t"<<model_name;
        cout<<"\nPrice:-\t"<<price;
}

void main()
{
        clrscr();
        Model r1;
        r1.get_car_type();
        r1.get_brand_details();
        r1.get_model_details();
        cout<<"\nCar Details:\n";
        r1.put_model_details();
        getch();

}
```

//Exp 13.1

```
#include <iostream.h>
#include<conio.h>
class Area
{
  public:
    float area(float l,float b)
    {
```

```
        return l*b;
    }
};

class Perimeter
{
  public:
    float perimeter(float l,float b)
    {
        return 2*(l+b);
    }
};

class Rectangle : public Area, public Perimeter
{
    private:
        float length, breadth;
    public:
      void get_data( )
      {
          cout<<"Enter length: ";
          cin>>length;
          cout<<"Enter breadth: ";
          cin>>breadth;
      }
```

```
    void show()
    {
        cout<<"Area = "<<area(length,breadth);
        cout<<"\nPerimeter = "<<perimeter(length,breadth);
    }
};

int main()
{
   clrscr();
   Rectangle r;
   r.get_data();
   cout<<"\nArea & Perimeter of Rectangle:\n";
   r.show();
   getch();
   return 0;
}
```

//Exp 13.2

```
#include<iostream.h>
#include<conio.h>

class Cricketer
{
   char name[15];
```

```
    int no_of_matches;
public:
    void getinfo()
    {
            cout << "Enter Name of cricketer:";
            cin>>name;
            cout << "Enter No. of Matches played:";
            cin>>no_of_matches;
    }

    void putinfo()
    {
            cout << "\n\nName of cricketer:" <<name << "\n";
            cout << "No. of Matches played:"<<no_of_matches;
    }
};

class Batsman : virtual public Cricketer
{
   long int no_of_runs;
  public:
    void getruns()
    {
            cout << "Enter Total No.of Runs:";
            cin>>no_of_runs;
```

```
    }

    void putruns()
    {
        cout << "\nTotal No.of Runs:" << no_of_runs;
    }
};

class Bowler : public virtual Cricketer
{
    int no_of_wickets;
  public:
    void getno_of_wickets()
    {
        cout << "Enter Total No.of Wickets:";
        cin>>no_of_wickets;
    }

    void putno_of_wickets()
    {
        cout << "\nTotal no_of_wickets:" << no_of_wickets;
    }
};

class allrounder : public Batsman, public Bowler
```

```
{
   int total;
  public:

   void display()
   {
         putinfo();
         putruns();
         putno_of_wickets();
   }
};

void main()
{
   allrounder obj;
   clrscr();
   obj.getinfo();
   obj.getruns();
   obj.getno_of_wickets();
   cout<<"\nDetails of Allrounder Cricketer:\n";
   obj.display();
   getch();
}
```

//EXP14.1

```
#include<iostream.h>
```

```
#include<conio.h>
#include<string.h>
class Book
{
	char author[10];
	char book_name[10];
	float price;
	public:

	void getdata()
	{

			cout<<"Enter Book name:-> ";
			cin>>book_name;
			cout<<"Enter Book Price:->";
			cin>>price;
			cout<<"Enter Author Name:-> ";
			cin>>author;
	}
	void display()
	{

			cout<<"\nBook name:-> "<<book_name;
			cout<<"\nBook Price:->"<<price;
			cout<<"\nAuthor:-> "<<author;
	}

};

void main()
{
	Book b1,*bptr;
	clrscr();
	bptr=&b1;
	bptr->getdata();
	(*bptr).display();
  getch();
```

```
}
```

//EXP142.2

```
#include<iostream.h>
#include<conio.h>
class box
{
 float height,width,breadth;
 public:
 void get()
 {
  cout<<"\n (1).Enter height  =";
  cin>>height;
  cout<<"\n (2).Enter width   =";
  cin>>width;
  cout<<"\n (3).Enter breadth =";
  cin>>breadth;
 }
 void area();
 void volume();
};
 void box::area()
 {
  cout<<endl<<"\n Area of box   = "<<(width*height)<<"cm^2";

 }
 void box::volume()
 {
  cout<<endl<< "\nVolume of box ="<<(width*breadth*height)<<"cm^3";
 }
 int main()
 {
  clrscr();
  cout<<"*** calculation of volume and area of box  ***"<<endl;
  box b;
  box *ptr;
  ptr=&b;
```

```
 ptr->get();
 (*ptr).area();
 (*ptr).volume();
 getch();
 return 0;
}
```

//EXP14.3

```
#include<iostream.h>
#include<conio.h>
class birthday
{
 char name[25];
 int day;
 int month;
 int year;
public:
 void get()
 {
 cout<<"\nenter the name      : ";
 cin>>name;
 cout<<"\nenter the birth day : ";
 cin>>day;
 cout<<"\nenter the birth month : ";
 cin>>month;
 cout<<"\nenter the birth year : ";
 cin>>year;
 }
 void display()
 {
 cout<<"\n NAME      ="<<name;
 cout<<"\n BIRTH DATE ="<<day<<"/"<<month<<"/"<<year;
 }
};
int main()
{
 clrscr();
```

```
 int i=0;
 birthday b[5];
 birthday *B;
 B=&b[0];


 for(i=0;i<5;i++)
 {
 B->get();
 B++;
 }
 cout<<" * * * BIRTHDATE INFORMATION * * * "<<endl;
 B=&b[0];
 for(i=0;i<5;i++)
 {
  B->display();
  B++;
 }
 getch();
 return 0;
}
```

EXP 15.1

```
#include<iostream.h>
#include<conio.h>
class polygon
{
        int width,height;
        public:
        void get()
        {
                cout<<"\n Enter width & height:";
                cin>>width>>height;
        }
        int wid()
        {
                return width;
```

```
        }
        int heig()
        {
                return height;
        }
};
class triangle:public polygon
{
        public:
        void area()
        {
                float a;
                int w=wid();
                int h=heig();
                a=0.5*w*h;
                cout<<"\n Area of triangle: "<<a;
        }
};
class rectangle:public polygon
{
        public:
        void area()
        {
                int b=wid();
                int l=heig();
                float a=l*b;
                cout<<"Area of rectangle: "<<a;
        }
};
int main()
{
        triangle t, *tptr;
        rectangle r, *rptr;
        tptr=&t;
        tptr->get();
        tptr->area();
        rptr=&r;
        rptr->get();
```

```
        rptr->area();
        getch();
        return 0;
}
```

EXP 16.1

```
#include <iostream.h>
#include<conio.h>
class Check
{
  private:
    int i;
  public:
    Check()
    {
        i=0;
    }

    Check operator ++ ()
    {
        Check temp;
        temp.i = i++;
        return temp;
    }

    Check operator -- ()
```

```
    {
        Check temp;
        temp.i = i--;
        return temp;
    }

    void Display()
    { cout << "i = "<< i <<endl; }
};

int main()
{
    clrscr();
    Check obj, obj1;
    obj.Display();
    obj1.Display();

    cout<<"Postincrement \n";
    obj++;
    obj.Display();

    cout<<"Postdecrement \n";
    obj--;
    obj.Display();
```

```
   getch();
   return 0;
}
```

//EXP 17.1 Complex Number addition operator overloading

```
#include<iostream.h>
class complex
{
   public:
   int real;
   int img;
   complex()
   {
      real=img=0;
   }
   complex(int x,int y)
   {
      real=x;
      img=y;
   }
   void show()
   {
      cout<<"\n"<<real<<"+"<<img<<"i";
   }
   friend complex operator+(complex c,complex d);
};

//operator+ is not part of complex class so have 2 args for + operator overload.
complex operator+(complex c, complex f)
{
   complex ans;
   ans.real=c.real+f.real;
   ans.img=c.img+f.img;
   return(ans);
}
```

```
int main()
{
   complex x(1,2), y(0,7);
   complex c=x+y;     //overloaded + is called here
   c.show();
}
```

//EXP 17.2 PROGRAM OF OVERLOADING ARITHMETIC OPERATORS

```
#include<iostream.h>
#include<conio.h>

class BINARY
{
   float no;
   public:
   BINARY(){}
   void getdata()
   {
      cout<<"\n ENTER AN FLOATING NUMBER :";
      cin>>no;
    }
    void putdata()
    {
      cout<<"\n\nANSWER IS                :"<<no;
    }
    BINARY operator+(BINARY);
    BINARY operator*(BINARY);
    BINARY operator-(BINARY);
    BINARY operator/(BINARY);
};
BINARY BINARY::operator+(BINARY a)
{
   BINARY temp;
   temp.no=no+a.no;
   return temp;
}
BINARY BINARY::operator*(BINARY b)
```

```
{
   BINARY temp;
   temp.no=no*b.no;
   return temp;
}
BINARY BINARY::operator-(BINARY b)
{
   BINARY temp;
   temp.no=no-b.no;
   return temp;
}
BINARY BINARY::operator/(BINARY b)
{
   BINARY temp;
   temp.no=no/b.no;
   return temp;
}


main()
{
  clrscr();
  Binary a,b,c;
  a.getdata();
  b.getdata();

cout<<"\n\nAFTER ADDITION OF TWO OBJECTS";
  c=a+b;
  c.putdata();
  cout<<"\n\nAFTER MULTIPLICATION OF TWO OBJECTS";
  c=a*b;
  c.putdata();
  cout<<"\n\nAFTER SUBSTRACTION OF TWO OBJECTS";
  c=a-b;
  c.putdata();
  cout<<"\n\nAFTER DIVISION OF TWO OBJECTS";
  c=a/b;
  c.putdata();
```

```
 getch();
}
```

//EXP 17.3 Definesa Class String and Overload == Operator to Compare Two Strings

```
#include<iostream.h>
#include<stdio.h>
#include<string.h>
class String
{
      char str[20];
      public:
      void getdata()                    //function to read the string
      {
           gets(str);
      }
      int operator==(String s)
      {
         if(!strcmp(str,s.str))
            return 1;
         else
            return 0;
      }
};
void main()
{
      String s1,s2;
     cout<<"Enter first string:";
      s1.getdata();
     cout<<"Enter second string:";
      s2.getdata();

     if(s1==s2)              //here the operator function will be called
      {
           cout<<"\nStrigs are Equal\n";
      }
      else
      {
```

```
        cout<<"\nStrings are Not Equal\n";
    }
    return 0;
}
```

//EXP 18.1 Program to interchange the values of two int , float and char using function overloading

```
#include<iostream.h>
#include<conio.h>
void swap(int &,int &);
void swap(float &,float &);
void swap(char &,char &);
main()
  {
    clrscr();
    int I1, I2;
    float F1, F2;
    char C1, C2;
    cout<<"\n Enter the value of I1 & I2 = ";
    cin>>I1>>I2;
    cout<<"\n Enter the value of F1 & F2 = ";
    cin>>F1>>F2;
    cout<<"\n Enter the value of C1 & C2 = ";
    cin>>C1>>C2;
    cout<<"\n ********* Before Swap()  **********"<<endl;
    cout<<"\n Values of int :"<<endl;
    cout<<"\t I1 = "<<I1<<"\t I2 = "<<I2<<endl;
    cout<<" Values of float :"<<endl;
    cout<<"\t F1 = "<<F1<<"\t F2 = "<<F2<<endl;
    cout<<" Values of  char :"<<endl;
    cout<<"\t Value of C1 = "<<C1<<"\t C2 = "<<C2<<endl;
    cout<<"\n ********* After Swap()  **********"<<endl;
    cout<<"\n Values of int :"<<endl;
    swap(I1,I2);
    cout<<"\t I1 = "<<I1<<"\t I2 = "<<I2<<endl;
    cout<<" Values of float :"<<endl;
    swap(F1,F2);
```

```
    cout<<"\t F1 = "<<F1<<"\t F2 = "<<F2<<endl;
    cout<<" Values of  char :"<<endl;
    swap(C1,C2);
    cout<<"\t Value of C1 = "<<C1<<"\t C2 = "<<C2<<endl;
    getch();
    return 0;
  }
void swap(int &v1,int &v2)
  {
    int temp;
    temp=v2;
    v2=v1;
    v1=temp;
  }
void swap(float &v1,float &v2)
  {
    float temp;
    temp=v2;
    v2=v1;
    v1=temp;
  }
void swap(char &v1,char &v2)
  {
    char temp;
    temp=v2;
    v2=v1;
    v1=temp;
  }
```

//EXP 18.3 C++ program to find area of square,rectangle,circle and triangle by using function overloadingC++

```
#include<iostream.h>
int area(int);
int area(int,int);
float area(float);
float area(float,float);
int main()
```

```
{
        int s,l,b;
        float r,bs,ht;
        cout<<"Enter side of a square:";
        cin>>s;
        cout<<"Enter length and breadth of rectangle:";
        cin>>l>>b;
        cout<<"Enter radius of circle:";
        cin>>r;
        cout<<"Enter base and height of triangle:";
        cin>>bs>>ht;
        cout<<"Area of square is"<<area(s);
        cout<<"\nArea of rectangle is "<<area(l,b);
        cout<<"\nArea of circle is "<<area(r);
        cout<<"\nArea of triangle is "<<area(bs,ht);
}
int area(int s)
{
   return(s*s);
}
int area(int l,int b)
{
   return(l*b);
}
float area(float r)
{
   return(3.14*r*r);
}
float area(float bs,float ht)
{
  return((bs*ht)/2);
}
```

//EXP 19.2 Program to merge contents of two files into a third file

```
#include <iostream.h>
#include <stdlib.h>   // For exit()
```

```
#include <fstream.h>
#include <conio.h>
int main()
{
  clrscr();
  ifstream inf1("file1.txt",ios::in);
  ifstream inf2;
  inf2.open("file2.txt",ios::in);

  ofstream outf;
  outf.open("file3.txt",ios::out);
  char c;

  if (inf1.fail() || inf2.fail() || outf.fail())
  {
        cout<<"Could not open files";
        exit(0);
  }

  while (!inf1.eof())
  {
    inf1.get(c);
    outf.put(c);
  }

  while (!inf2.eof())
  {
   inf2.get(c);
   outf.put(c);
  }
  outf.close();
  cout<<"Merged file1.txt and file2.txt into file3.txt\n\n";

  inf1.close();
  inf2.close();
  outf.close();
  getch();
  return 0;
```

```
}
```

//EXP 19.3 Program to print contents of file

```
#include <iostream.h>
#include <stdlib.h> // For exit()
#include <fstream.h>
#include <conio.h>
int main()
{
   clrscr();
   ifstream inf;

   char filename[100], c;

   cout<<"Enter the filename to open \n";
   cin>> filename;

   inf.open(filename, ios::in);
   if (inf.fail())
   {
         cout<<"Cannot open file \n";
         exit(0);
   }

   while (!inf.eof())
   {
         inf.get(c);
         cout<<c;
   }

   inf.close();
   getch();
   return 0;
}
```