Computer Science C++ Program File

Name :

Class :

Section:

INDEX

12	Program: Writing and reading data into file, using constructors	
13	Program To write data into the file, character by character	
14	Program of Writing an object into the file	
15	Program counts the number of objects already written into the file “Person.txt”. Then is reads the second object and displays the values of its data members.	
16	Program to process data from external file, Creating, opening, writing some data in fileand appending data at the end of file...	
17	Program to get rollnumbers and marks of the students of a class (get from the user) and store these details into a file called 'Marks.dat'	
18	Program to demonstrate the usage of this pointer.	
19	Program to find the even nos. and sum of all even number using I D Array	
20	Function to display the elements which are divisible by 10 of an array M[5][5]	
21	Program to find Row and column sum using 2D Array	
22	Program of Bubble sort function in Descending order	
23	Program to arrange 10 elements using selection sort	
24	Program to shift negative and positive numbers.	
25	Program search an element in an array using linear search	
26	Program of stack using array	
27	Program of stack using Linked List	
28	Program of Queue using Linked List	
29	SQL Commands using Two Tables	
30	SQL Commands using one Table	

```
//Program that illustrate the working of Random function
#include<stdlib.h>
#include<iostream.h>
void main()
```

```
{
randomize();
int NUM;
NUM=random(3)+2;
char TEXT[]="ABCDEFGHIJK";
for (int I=1;I<=NUM; I++)
{
for (int J=NUM;J<=7;J++)
cout<<TEXT[J];
cout<<endl;
}
}
```

OUTPUT
EFGH
EFGH
EFGH
EFGH

```
//Program that illustrate the working of Random function

#include<stdlib.h>
#include<iostream.h>
 int main()
{ randomize ();
int Num, Rndnum;
 Num=4;
Rndnum = random (Num) + 5;
for (int N = 1; N<=Rndnum; N++)
cout << N << " ";

}
```

OUTPUT
12345678

```
//Program to illustrate the working of function overloading
#include <iostream.h>
int operate (int a, int b)
{ return (a*b);
}
float operate (float a, float b)
{return (a/b);
}
int main ()
{ int x=5,y=2;
float n=5.0,m=2.0;
```

```
cout << operate (x,y);
cout << "\n";
cout << operate (n,m);
cout << "\n";
return 0;
}
/*
Output
10
2.5
*
/
```

```
//Program to illustrate the working of Scope resolution operator
#include<iostream.h>
#include<conio.h>
class FLIGHT
{
int Fno;
char Destination[20];
float Distance, Fuel;
void CALFUEL();
public:
void FEEDINFO();
void FEEDINFO();
void SHOWINFO();
};
void FLIGHT::CALFUEL()
{
if (Distance<=1000)
Fuel=500;
else
if (Distance<=2000)
Fuel=1100;
else
Fuel=2200;
}
void FLIGHT::FEEDINFO()
{
cout<<"Flight No :";cin>>Fno;
cout<<"Destination :";gets(Destination);
cout<<"Distance :";cin>>Distance;
CALFUEL();
}
void FLIGHT::SHOWINFO()
{
cout<<"Flight No :"<<Fno<<endl;
cout<<"Destination :"<<Destination<<endl;
cout<<"Distance :"<<Distance<<endl;;
cout<<"Fuel :"<<Fuel<<endl;;
```

```
}
void main()
{clrscr();
FLIGHT F;
F.FEEDINFO();
F.SHOWINFO();
}

/*OUTPUT
Flight No :1
Destination :delhi
Distance :200

Flight No :1
Destination :delhi
Distance :200
Fuel :500
*/
```

```
/*Use of a Class in a Program
Define the following in a sequence to use a class in a program
1. Class Definition
2. Class Method
3. In main( ) Create Object
*/

#include<iostream.h>
#include<conio.h>
#include<stdio.h>
class stud
{private: //private members
int sno;
char sname[80];
int cls;
public: //public members
void input(void);
void output(void);
};
void stud :: input(void)
{
cout<<"Enter Sno : ";
cin>>sno;
cout<<"Enter Sname : ";
gets(sname);
```

```
cout<<"\nEnter Class :";
cin>>cls;
}
void stud :: output(void)
{
cout<<"\n Sno : "<<sno;
cout<<"\n Sname : "<<sname;
cout<<"\n Class : "<<cls;
}
void main()
{ clrscr();
stud s;
s.input();
s.output();
getch();
}
/*

OUTPUT

Enter Sno : 1

Enter Sname : ashok



Enter Class :4

 Sno : 1

 Sname : ashok

 Class : 4

*/


// Program of default constructor
// The class is as :
#include <iostream.h>
#include <conio.h>
#include <string.h>
#include <stdio.h>
class TravelPlan
{
        long PlanCode;
        char Place[20];
        int Number_of_travellers;
        int Number_of_buses;
        public :
                TravelPlan();
                void NewPlan();
                void ShowPlan();
};
TravelPlan::TravelPlan()
{
        PlanCode = 1001;
```

```
        strcpy(Place, "Agra");
        Number_of_travellers = 5;
        Number_of_buses = 1;
}
void TravelPlan::NewPlan()
{
        cin >> PlanCode;
        gets(Place);
        cin >> Number_of_travellers;
        if (Number_of_travellers < 20)
                Number_of_buses = 1;
        else if (Number_of_travellers < 40)
                Number_of_buses = 2;
        else
                Number_of_buses = 3;
};
void TravelPlan::ShowPlan()
{
        cout << PlanCode << endl
        << Place << endl
        << Number_of_travellers << endl
        << Number_of_buses << endl;
}
void main()
{
        clrscr();
        TravelPlan TP;
        TP.NewPlan();
        TP.ShowPlan();
}

/*
OUTPUT
Enter PlanCode1
Enter Placedelhi
Enter no of Travellers23
1
delhi
23
2
*/
```

```
//Program show the working of Copy constructor

#include<iostream.h>

#include<conio.h>

class player

{
```

```
int health;
int age;
public:
player() { health=7; age=17; } //Constructor1
player(int h, int a) //Constructor2
{health =h;
age = a ; }
player( player &p)        //Constructor3
{health=p.health;
age=p.age;
 }
~player() { cout<<"Memory Free"; } //Destructor
};
void main()
{clrscr();
player p1(9,26);
player p3 = p1;
}

/* OUTPUT
Memory Free

*/

//Program of Parameterized constructor
#include<iostream.h>
#include<conio.h>
class stud
{
int sno; char sname[40];
Public :
stud(int s,char n[ ]) // Paramerized Constructor
{
sno=s;
strcpy(sname,n);
}
void getinfo( )
{ cin>> sno;
```

```
gets(sname);
}
void showinfo( )
{ cout<< sno;
puts(sname);
}
};
void main()
{ clrscr();
stud obj (1, “Ashu”); // Parameterized constructor invoked
Obj.showinfo(); // displays the value of sno as 1 and sname as “Ashu”
Obj.getinfo(); // reads the user given value from the user
Obj.showinfo(); // displays the user given values
}
/* OUTPUT
1Ashu
2 dd
2dd
*/
```

```
// Program of Hybrid Inheritance

#include <iostream.h>

#include<stdio.h>

class person

{

        private:

                int code;

                char name[10];

        public :

                void read()

                {

                        cout << "Enter the code ";

                        cin >> code;

                        cout << "Enter the name ";

                        gets(name);

                }

                void display()

                {
```

```
                cout << "\nThe code " << code;
                cout << "\nThe name ";
                puts(name);
        }
};
class account : public person
{
        float pay;
        public:
                void read_acc()
                {
                        cout << "Enter the pay of account ";
                        cin >> pay;
                }
                void display_acc()
                {
                        cout << "\nThe pay of account " << pay;
                }
};
class admin : public person
{
        private :
                int experience;
        public:
                void read_adm()
                {
                        read();
                        cout << " Enter the administrative experience ";
                        cin >> experience;
                }
                void display_adm()
                {
```

```
                display();
                cout << " \nThe administrative experience is : " << experience;
        }
};
class master : public account, public admin
{
    public :
        void read_master()
        {
            read_acc();
            read_adm();
        }
        void display_master()
        {
            display_acc();
            display_adm();
        }
};
void main()
{
    master obj;
    obj.read_master();
    obj.display_master();

}
/*OUTPUT
Enter the pay of account 12.02
Enter the code 1
Enter the name abc
 Enter the administrative experience 4

The pay of account 12.02
```

```
The code 1
The name abc

The administrative experience is : 4
*/


// Program to show the multiple inheritance
# include <iostream.h>
# include <stdio.h>
# include <conio.h>
class Person
{
        char name[20];
        char sex;
        int age;
        public :
                void read_p()
                {
                        cout << "Enter the name ";
                        gets(name);
                        cout << "Enter sex ";
                        cin >> sex;
                        cout << "Enter age ";
                        cin >> age;
                }
                void display_p()
                {
                        cout << "\nName " << name;
                        cout << "\nSex " << sex;
                        cout << "\nAge " << age;
                }
};
```

```
class Hospital
{
        int bed_no;
        char illness[20];
        public:
                void read_h()
                {
                        cout << "Enter the bed number ";
                        cin >> bed_no;
                        cout << "Enter the type of illness ";
                        gets(illness);
                }
                void display_h()
                {
                        cout << "\nBed Number " << bed_no;
                        cout << "\nType of illness " << illness;
                }
};
struct date
{
        int dd;
        int mm;
        int yy;
};
class Patient : private Person, private Hospital
{
        date  dob;
        public:
                void read()
                {
                        read_p();
                        read_h();
```

```
                        cout << "Enter the date ";
                        cin >> dob.dd;
                        cin >> dob.mm;
                        cin >> dob.yy;
                }
                void display()
                {
                        display_p();
                        display_h();
                        cout << "\nDate of Admission " << dob.dd << "/" << dob.mm << "/" <<
dob.yy;
                }
};
void main()
{
        Patient obj;
        clrscr();
        cout << "Enter the data of the patient\n";
        obj.read();
        cout << "\nThe data of the patient \n";
        obj.display();
}
/*OUTPUT
Enter the data of the patient
Enter the name abc
Enter sex f
Enter age 33
Enter the bed number 12
Enter the type of illness diabetes
Enter the date 2/3/14

The data of the patient
```

```
Name abc
Sex f
Age 33
Bed Number 12
Type of illness diabetes
Date of Admission 2/5/14
                                */
```

```
// Program of Hierarchical Inheritance
#include<iostream.h>
#include<stdio.h>
#include<conio.h>
class building
{
        private:
                int floor;
                int number, footage;
        public :
                void read()
                {
                        cout << "Enter number of floors : ";
                        cin >> floor;
                        cout << "Enter number of rooms : ";
                        cin >> number;
                        cout << "Enter total square of footage : ";
                        cin >> footage;
                }
                void display()
                {
                        cout << "\nNo. of floors : " << floor;
                        cout << "\nNo. of rooms : " << number;
```

```
                cout << "\nTotal footage : " << footage;
        }
};
class house : public building
{
        int bedroom, bathroom;
        public:
                void read_house()
                {
                        cout << "\nEnter data for house..." << endl;
                        read();
                        cout << "Enter no. of bedrooms : ";
                        cin >> bedroom;
                        cout << "Enter no. of bathrooms : ";
                        cin >> bathroom;
                }
                void display_house()
                {
                        display();
                        cout << "\nNo. of bedrooms : " << bedroom;
                        cout << "\nNo. of bathrooms : " << bathroom;
                }
};
class office : public building
{
        private :
                int fire;
                int telephone;
        public:
                void read_office()
                {
                        cout << "\nEnter data for office..." << endl;
```

```
                read();
                cout << "Enter no. of fire extinguishers : ";
                cin >> fire;
                cout << "Enter no. of telephones : ";
                cin >> telephone;
        }
        void display_adm()
        {
                display();
                cout << "\nNo. of fire extinguishers : " << fire;
                cout << "\nNo. of telephones : " << telephone;
        }
};
void main()
{
        clrscr();
        house hsu;
        hsu.read_house();
        hsu.display_house();
        office off;
        off.read_office();
        off.display_adm();
}
/*OUTPUT

Enter data for house...
Enter number of floors : 3
Enter number of rooms : 4
Enter total square of footage : 200
Enter no. of bedrooms : 3
Enter no. of bathrooms : 2
```

No. of floors : 3

No. of rooms : 4

Total footage : 200

No. of bedrooms : 3

No. of bathrooms : 2

Enter data for office...

Enter number of floors : 4

Enter number of rooms : 3

Enter total square of footage : 300

Enter no. of fire extinguishers : 5

Enter no. of telephones : 3

No. of floors : 4

No. of rooms : 3

Total footage : 300

No. of fire extinguishers : 5

No. of telephones : 3

*/

//Program: Writing and reading data into file, using constructors

```
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
void main()
{
ofstream outfile("sample.txt"); // create file for output
char ch = 'a';
int i = 12;
float f = 4356.15;
char arr[ ] = "hello";
outfile << ch << endl << endl << f << endl << arr; //send the data to file
outfile.close();

ifstream infile("sample.txt");


infile >> ch >> i >> f >> arr; // read data from file

cout << ch << i << f << arr; // send data to screen
}
```

```
/*
OUTPUT

D:\TC>type sample.txt
a

4356.149902
hello
D:\TC>exit
```

```
//Program To write data into the file, character by character.

#include<fstream.h>
#include<conio.h>
#include<string.h>
void main()
{
char str[]="C++ is superset of C. It is an object-oriented /programming language.";

ofstream outfile("sample2.txt"); // Open the file in write mode

for(int i = 0; i < strlen(str); i++)
outfile.put(str[i]); // write data into the file, character by character.
}

/*OUTPUT
D:\TC>type sample2.txt
C++ is superset of C. It is an object-oriented /programming language.
D:\TC>exit




*/
```

```
//Program of Writing an object into the file


#include<fstream.h>

#include<conio.h>

class Person

{

private:
```

```
char name[40];
int age;
public:
void getData()
{
cout << "\n Enter name:"; cin >> name;
cout << "\n Enter age:"; cin >> age;
}
} ; // End of the class definition

void main()
{clrscr();
Person per ; // Define an object of Person class

per.getData(); // Enter the values to the data members of the class.

ofstream outfile("Person.txt"); // Open the file in output mode

outfile.write((char*)&per, sizeof(per)); // Write the object into the file
}
 /*
 OUTPUT

 Enter name:asha

 Enter age:23

Type EXIT to return to Turbo C++. . .Microsoft(R) Windows DOS
(C)Copyright Microsoft Corp 1990-2001.

D:\TC>type person.txt
asha >type person.txt
```

```
D:\TC>exit
      */
```

```
/*This program counts the number of objects already written into the file “Person.txt”. Then is reads the second object and displays the values of its data members.
*/
#include<iostream.h>
#include<fstream.h>
#include<conio.h>
class person
{
private:
char name[40];
int age;
public:
void showData()
{
cout << "\n Name = " << name;
cout << "\n Age = " << age;
}
};

void main()
{clrscr();
person pers; // create person object
ifstream infile; // create input file
infile.open("Person.txt"); // open the file
infile.seekg(0, ios::end); // go to end from 0 byte
int endposition = infile.tellg(); // find where we are
int n = endposition/sizeof(person); // number of persons
cout <<"\n There are "<< n << " persons in file: ";
```

```
cout << "\n Enter person number: ";
cin >> n;
int position = (n-1) * sizeof(person); // number times size
infile.seekg(position);
infile.read( (char*)&pers, sizeof(pers) );
pers.showData(); // display the person
}

/*OUTPUT

 There are 1 persons in file:
 Enter person number: 1

 Name = asha
 Age = 23

 */
```

```
// a simple processing data from external file.
// Creating, opening and writing some data in file
// and appending data at the end of file...
#include <iostream.h>
#include <fstream.h>
#include<process.h>
#include<conio.h>

void main(void)
{                clrscr();
   char filename[ ] = "C:\\testfileio2.txt";
   ofstream outputfile;
```

```
   // creating, opening and writing/appending data to file
   outputfile.open(filename, ios::out|ios::app);
   // simple error handling for file creating/opening for writing
   // test if fail to open the file, do…
   if(outputfile.fail())
   {
      cout<<"Creating and opening file "<<filename<<" for writing\n";
      cout<<"-----------------------------------------\n";
      cout<<"The "<<filename<<" file could not be created/opened!\n";
      cout<<"Possible errors:\n";
      cout<<"1.  The file does not exist.\n";
      cout<<"2.  The path was not found.\n";
      exit(1);   // just exit
      // 0-normal, non zero - some error
}
// else, if the file can be opened, do…
else
{
      cout<<"The "<<filename<<" file was created and opened successfully!\n";
      cout<<"\nDo some file writing....\n\n";
      outputfile<<"Writing some data in this file\n";
      outputfile<<"----------------------------\n";
      cout<<"Check the "<<filename<<" file contents :-)"<<endl;
      cout<<"If the file already have had data, the new data will be appended\n";
      int sampledata;
      // write some integers to the file...
      for(sampledata=0; sampledata<=10; sampledata++)
      outputfile<<sampledata<<"  ";
      outputfile<<endl;
      // close the output file
      outputfile.close();
      // test if fail to close the file, do the following...
```

```
	// simple error handling for output files closing
	if(outputfile.fail())
	{
		cout<<"The "<<filename<<" file could not be closed!\n";
		exit(1);
	}
	// test if successful to close the file, do the following...
	else
		cout<<"\nThe "<<filename<<" file was closed successfully!\n";
}
}
/*OUTPUT
The C:\testfileio2.txt file was created and opened successfully!

Do some file writing....

Check the C:\testfileio2.txt file contents :-)
If the file already have had data, the new data will be appended

The C:\testfileio2.txt file was closed successfully!
*/
```

```
/*Program to get rollnumbers and marks of the students of a class (get from the user) and store
these details into a file called 'Marks.dat' */

#include<fstream.h>
void main( )
{
   ofstream filout ;                        // stream decided and declared - steps 1 & 2
   filout.open("marks.dat", ios :: out) ;       // file linked - step 3
   char ans = 'y' ;                         // process as required - step 4 begins
```

```
    int rollno ;
    float marks ;
    while(ans == 'y' || ans == 'Y')
    {
        cout << " \n Enter Rollno. :" ;
        cin >> rollno;
        cout << " \n Enter Marks :" ;
        cin >> marks ;
        filout << rollno << " \n " << marks << " \n " ;
        cout << " \n Want to enter more records?(y/n)..." ;
        cin >> ans ;
    }
    filout.close( ) ;                    // delink the file - step 5
}
```

```
//Program to demonstrate the usage of this pointer.
#include<iostream.h>
#include<conio.h>
class Rectangle
{ float area,len,bre;
public:
void input( )
{ cout<<"\nEnter the length and breadth: ";
cin>>this->len>>this->bre;
}
void calculate( )
{ area=len*bre;
//Here Implicit 'this' pointer will be worked.
}
void output( )
{
```

```
cout<<"\nThe Area of the Rectangle: "<<this->area;
}
};
void main( )
{
Rectangle R;
clrscr( );
R.input( );
R.calculate( );
R.output( );
getch();
}
/*OUTPUT

Enter the length and breadth: 2
4
The Area of the Rectangle: 8
*/

//Program to find out the output of the following code
#include<iostream.h>
void main( )
{char *a[ ]={“DELHI”, “MUMBAI”,”VARANSAI”};
char **p;
p=a;
cout<<sizeof(a)<<”, “<<sizeof(a[0])<<”, “<<sizeof(p)<<endl;
cout<< p << “, ”<<++*p<<“, ”<<*++p<<endl;
cout<< **a << “, ” <<*(a+1)<<“, ” <<a[2]<<endl;
}
/*OUTPUT
6, 2, 2
0x8fdffff2, UMBAI, MUMBAI
```

```
D, UMBAI, VARANSAI
*/



// Function Lower_half which prints the lower half of the array
#include <iostream.h>
#include <conio.h>
void Lower_half(int A[5][5])
{
	int k = 0, j;
	for(int i=0; i<5; i++)
	{
		int j = 0;
		while(j<=k)
		{
			cout << A[i][j] << "  ";
			j++;
		}
		cout << "\n";
		k++;
	}
}
void main()
{
	clrscr();
	int X[5][5];
	int i, j;
	cout << "Enter the array elements : ";
	for (i=0; i<5; i++)
	{
		for (j=0; j<5; j++)
```

```
                                cin >> X[i][j];
            }


            Lower_half(X);
}
/*OUTPUT
Enter the array elements : 10
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
10
```

```
5  6
10  11  12
15  16  17  18
20  21  22  23  24
*/
```

```
// Program  to find the even nos. and sum of all even number using I D Array
#include <iostream.h>
#include <conio.h>
const r = 100, c = 100;
void sum(int AR[10], int n)
{
        int i, j, sum = 0;
        for( i=0; i<n; i++)
        {
                if (AR[i] % 2 == 0)
                {
                        sum = sum + AR[i];
                        cout << "\nEven Number is : " << AR[i];
                }
        }
        cout << "\nThe sum of even numbers is : " << sum;
}
void main()
{
        clrscr();
        int ar[r];
        int rr;
        int i, j;
```

```
	cout << "Enter total no. of elements in array : ";
	cin >> rr;
	cout << "Enter the array elements : ";
	for (i=0; i<rr; i++)
		cin >> ar[i];
	sum(ar, rr);
}

/*OUTPUT
Enter total no. of elements in array : 10
Enter the array elements : 1
2
3
4
5
6
7
8
9
10

Even Number is : 2
Even Number is : 4
Even Number is : 6
Even Number is : 8
Even Number is : 10
The sum of even numbers is : 30

*/
```

```
// Function to display the elements which are divisible by 10 of an array M[5][5]
#include <iostream.h>
#include <conio.h>
void Display10(int M[5][5])
{
	for(int i=0; i<5; i++)
	{
		for(int j=0; j<5; j++)
		{
			if (M[i][j]%10 == 0)
				cout << M[i][j];
		}
	}
}
void main()
{
	clrscr();
	int AR[5][5], x, y;
	cout << "Enter the array elements : ";
	for (x=0; x<5; x++)
		for (y=0; y<5; y++)
			cin >> AR[x][y];
	Display10(AR);
}

/*OUTPUT
Enter the array elements : 3
5
6
7
10
20
```

```
36
40
60
50
70
48
65
64
30
20
27
40
80
90
43
57
2
6
89

102040605070302040809O
*/
```

```
//Program to find Row and column sum using 2D Array
#include <iostream.h>
#include <conio.h>

void row_col(float ARR[5][5], int r, int c)
{
        int i, j, rsum = 0, csum = 0;
        for (i = 0; i < r; i++)
```

```
	{
		rsum = 0;
		for (j = 0; j < c; j++)
		{
			rsum = rsum + ARR[i][j];
		}
		cout << "\n\t sum of row " << (i+1) << " is " << rsum;
	}
	for (i = 0; i < c; i++)
	{
		csum = 0;
		for (j = 0; j < r; j++)
		{
			csum = csum + ARR[j][i];
		}
		cout << "\n\t sum of column " <<(i+1)<< " is " << csum;
	}
}
void main()
{
	int r = 5, c = 5, i, j;
	float A[5][5];
	clrscr();
	cout << "Enter the array elements : ";
	for (i=0; i<5; i++)
	{
		for (j=0; j<5; j++)
			cin >> A[i][j];
	}
	row_col(A, r, c);
}
/*
```

OUTPUT

```
Enter the array elements : 2
3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23

        sum of row 1 is 11
        sum of row 2 is 30
        sum of row 3 is 55
        sum of row 4 is 80
        sum of row 5 is 105
```

```
        sum of column 1 is 48
        sum of column 2 is 53
        sum of column 3 is 55
        sum of column 4 is 60
        sum of column 5 is 65

*/
```

```
// Program of Bubble sort function in  Descending order is :
#include <iostream.h>
#include <conio.h>
void bubble_sort(int a[], int N)
{
        int i, j, temp;
        for (i = 0; i < N; i++)
                for (j = 0 ; j < N-1; j++)
                        if (a[j] < a[j+1])
                        {
                                temp = a[j];
                                a[j] = a[j+1];
                                a[j+1] = temp;
                        }
}
void main()
{
        int A[10];
        clrscr();
        cout << "Enter the array elements : ";
        for (int i = 0; i<10; i++)
                cin >> A[i];
```

```
        bubble_sort(A, 10);
        cout << "Sorted elements are : ";
        for (i = 0; i<10; i++)
                cout << A[i] << "\n";

}
/* output
Enter the array elements : 1
2
3
4
5
75
43
27
87
56
Sorted elements are : 87
75
56
43
27
5
4
3
2
1

*/
```

```
// Program to arrange 10 elements using selection sort
#include <iostream.h>
#include <conio.h>
#include <stdio.h>
#include<iomanip.h>
#include<string.h>

void SelectionSort()
{
	int range[10], loc, lowest, T, N, i, j, x;
	N = 10;
	cout << "Enter the array elements : \n";
	for (i = 0; i < N; i++)
	{
		cout << "Location " << i + 1 << ". Value :  ";
		cin >> range[i];
	}
	for (i = 0; i < N - 1; i++)
	{
		lowest = range[i];
		loc = i;
		for (j = i + 1; j < N; j++)
		{
			if (lowest > range[j])
			{
				loc = j;
				lowest = range[j];
			}
		}
		T = range[i];
		range[i] = range[loc];
```

```
                range[loc] = T;
        }
        cout << "\nThe sorted list is ... \n";
        for (i = 0; i < N; i++)
                cout << range[i] << endl;
}
void main()
{
        clrscr();
        SelectionSort();
}
/* OUTPUT
Enter the array elements :
Location 1. Value :  23
Location 2. Value :  43
Location 3. Value :  42
Location 4. Value :  67
Location 5. Value :  54
Location 6. Value :  64
Location 7. Value :  89
Location 8. Value :  75
Location 9. Value :  90
Location 10. Value :  13

The sorted list is ...
13
23
42
43
54
64
67
```

```
75
89
90

*/
```

```
// Program to shift negative and positive numbers.
#include <iostream.h>
#include <conio.h>
void swap(int a[10])
{
        int i = 0, j = 0, t;
        for (i =0; i <10 ; i++)
        {
                j = i + 1;
                if (a[j] <0)
                {
                        t = a[j];
                        while (j>=0)
                        {
                                a[j] = a[j-1];
                                 j--;
                                if (a[j] < 0) break;
                        }
                        a[j+1] = t;
                }
        }
}
void main()
{
```

```
	clrscr();
	int i, ar[10];
	cout << "Enter array elements : ";
	for (i = 0; i < 10; i++)
		cin >> ar[i];
	swap(ar);
	cout << endl;
	for (i=0; i<10; i++)
		cout << ar[i] << endl;
}
/*
OUTPUT

 Enter array elements : 1
3
4
6
-5
-23
5
34
-53
67




-5
-23
-53
1
3
4
```

```
6
5
34
67

*/
```

```
// This Program search an element in an array using linear search.
// data is the searched element in array P
#include<iostream.h>
#include<conio.h>
const val = 100;
int linear(float AR[val], float data ,int n)
{
	int  i, flag = 0, pos = 1;
	i= 0;
	while ( i < n)
	{
		if (AR[i] == data)
		{
			flag = 1;
			break;
		}
		i++;
	 }
	if (flag == 1)
		return(1);
	else
		return(0);
}
```

```
main()
{
        float AR[100],data;
        int n,x;
        clrscr();
        cout << "Enter n";
        cin >>n;
        for(int i=0;i<n;i++)
        cin >>AR[i];
        cout << "Enter number to be searched ";
        cin >>data;
        x = linear(AR,data,n);
        if (x ==1)
        cout << "Number exists in the list";
        else
        cout << "Number does not exists in the list";
}
/*OUTPUT
Enter n4
2
15
5
76
Enter number to be searched 15
Number exists in the list
*/


//Program of stack using array
#include<iostream.h>
#include<conio.h>
void main()
```

```
{               clrscr();
int s[6];
int top=-1;
int n;char ch='y';
while(ch=='y'||ch=='Y')
{
cin>>n;
if(top==5)
cout<<"overflow";
else
{top++;
s[top]=n;
}
cout<<s[top]<<"<--"<<endl;
for(int i=top-1;i>=0;i--)
cout<<s[i]<<endl;
cout<<"enter more";
cin>>ch;
}
cout<<"want to delete";
cin>>ch;
if (top==-1)
cout<<"underflow";
else
{
cout<<endl<<s[top]<<"Deleted\n";
top--;
}
cout<<endl<<s[top]<<"<--"<<endl;
for(int i=top-1;i>=0;i--)
cout<<s[i]<<endl;
getch();
```

```
}
/*
OUTPUT
12
12<--
enter morey
23
23<--
12
enter morey
65
65<--
23
12
enter moren
want to deletey

65Deleted

23<--
12
*/


//Program of stack using linked list
#include<iostream.h>
#include<conio.h>
struct student
{int rollno;char name[20];student *next;
};
student *p,*temp,*top;
void main()
```

```
{clrscr();            char ch='y';
top=NULL;
while(ch=='y'||ch=='Y')
{p=new student;
cin>>p->rollno;
cin>>p->name;
p->next=NULL;
if (top==NULL)
top=p;
else
temp=top;
top=p;
p->next=temp;
cout<<"enter more";
cin>>ch;
}

temp=top;
while(temp!=NULL)
{cout<<temp->rollno<<temp->name<<"->";
temp=temp->next;
}
getch();
//system("pause");
cout<<"\nwant to delete: ";
cin>>ch;
if(ch=='y'||ch=='Y')
{
if(top==NULL)
cout<<"underflow";
else
{temp=top;
```

```
top=top->next;
delete temp;
cout<<"deleted\n";
}}
temp=top;
while(temp!=NULL)
{cout<<endl<<temp->rollno<<temp->name<<"->";
temp=temp->next;
}
getch();

}
/*OUTPUT
1
heena
enter morey
2
teena
enter morey
3
defg
enter moren
3defg->2teena->1heena->
want to delete: y
deleted

2teena->
1heena->
*/


//Program of queue using linked list
```

```
#include<iostream.h>
#include<conio.h>
struct student
{int rollno;char name[20];student *next;
};
student *p,*temp,*start,*end ;
void main()
{clrscr();              char ch='y';
start=end=NULL;
while(ch=='y'||ch=='Y')
{p=new student;
cin>>p->rollno;
cin>>p->name;
p->next=NULL;
if (start==NULL)
start=end=p;
else
{end->next=p;
end=p;}
//p->next=NULL;}
cout<<"enter more";
cin>>ch;
}
temp=start;
while(temp!=NULL)
{cout<<temp->rollno<<temp->name<<"->";
temp=temp->next;
}

//system("pause");
cout<<"\nwant to delete\n";
cin>>ch;
```

```
if(ch=='y'||ch=='Y')
{
if(start==NULL)
cout<<"underflow";
else
{temp=start;
start=start->next;
delete temp;
cout<<"deleted\n";
}}
temp=start;
while(temp!=NULL)
{cout<<temp->rollno<<temp->name<<"->";
temp=temp->next;
}
getch();



}
/*OUTPUT
1
rert
enter morey
2
fdre
enter morey
3
yyyy
enter moren
1rert->2fdre->3yyyy->
want to delete
y
```

deleted

2fdre->3yyyy->

*/

Consider the following tables Emp and Dept. Write SQL commands for the statements (i) to (iv)

Table: Emp

Empno	Empname	Job	Hiredate	Sal	Deptno
7839	King	President	23/02/02	5000	1
7688	Blake	Manager	20/01/92	2850	2
7782	Clark	Manager	10/02/02	8500	1
7844	Martin	Salesman	01/01/02	1250	3
7369	Turner	Clerk	12/01/02	1500	2

Table: Dept

Deptno	Deptname	Location
1	Account	Delhi
2	Sales	Mumbai
3	Marketing	Delhi
4	Purchase	Kolkata

(a) Find out number of employees having “Manager “ as Job
(b) Create view dept2 with empname and the salary of employees for dept is 1.
(c) To find all those employees whose job does not start with ‘M’.
(d) To display all employees working in Delhi.

Ans

(a)
```
select count(empno) from emp
    where job='Manager' group by job;
```

(b)
```
create view dept2
as select empname, sal
from emp
where deptno=1;
```

(c)
```
select * from emp  Where job not like 'M%';
```

(d)
```
select empname, sal, location
from emp, dept
where emp.deptno=dept.deptno
and dept.location='Delhi';
```

Write SQL commands for (i) to (iv) on the basis of table **CLUB**

Table: **CLUB**

Coach_id	Coachname	Age	Sports	DATOFAPP	Pay	Sex
1	Kukerja	35	Karate	27/03/96	1000	M
2	Ravina	34	Karate	20/01/98	1200	F
3	Karan	34	Squash	19/02/98	2000	M
4	Tarun	33	Basketball	01/01/98	1500	M
5	Zubin	36	Swimming	12/01/98	750	M
6	Ketaki	36	Swimming	24/02/98	800	F

(i) To show all information about the swimming coaches in the club.
(ii) To list names of all coaches with their date of appointment(DATEODAPP) in descending order)
(iii) To display a report, showing coachname, pay, age and bonus(15% of pay) for all the coaches.
(iv) To insert a new row in the CLUB table with the following data:
7, 'Prakash', 37,'Squash','25/2/98',2500,'M'

(i) select * from club where sports='Swimming';

(ii) select coachname from club order by DATEOF APP desc;

(iii) select coachname, pay, age, 0.15*pay from club;

(iv) insert into club values (7, 'Prakash', 37,'Squash','25/2/98',2500,'M');