**#include**<iostream>

**#include**<string.h>

**using** **namespace** std;

//class declaration

**class** Student\_class

{

**private**:

//data members

**struct** student

{

**int** roll\_no;

string name;

**int** credit[5];

**int** grade[5];

**int** sgpa;

}s[20];

**public**:

//method declaration

**void** **input**(**int**);

**bool** **name\_validation**(string);//name validation

**bool** **rollNo\_validation**(**int**);//roll number validation

**void** **display**(**int**);//display method

**void** **sort\_rollNo**(**int**);//bubble sort

**void** **sort\_aplhabetically**(**int**);//insertion sort

**void** **sort\_sgpa**(**int**,**int**);//quick sort

**void** **display\_topper**(**int**);//to display toppers

**void** **search\_SGPA**(**int**);//linear search

**void** **search\_name**(**int**);//binary search

};

//definition of input method

**void** **Student\_class::input**(**int** n)

{

**for**(**int** i=0;i<n;i++)

{

**bool** valid;

cout<<**endl**<<"-----Student"<<i+1<<"----"<<**endl**;

**do**

{

cout<<"Roll\_no.:";

cin>>s[i].roll\_no;

valid=rollNo\_validation(i);

}**while**(!valid);

cin.ignore();

**do**

{

cout<<"Name:\n";

**getline**(cin,s[i].name);

valid=name\_validation(s[i].name); //validate name

}**while**(!valid);

cout<<"Enter marks of 5 subjects:"<<**endl**;

**double** sum\_of\_product=0;

**int** total\_credit=0;

**for**(**int** j=0;j<5;j++)

{

cout<<"\*\*Subject "<<j+1<<":\*\*"<<**endl**;

**do**

{

cout<<"Credit:";

cin>>s[i].credit[j];

**if**(s[i].credit[j]>5||s[i].credit[j]<1)

cout<<"Credit should be in range of 1 to 5"<<**endl**;

}**while**(s[i].credit[j]>5||s[i].credit[j]<1);

total\_credit+=s[i].credit[j];

**do**

{

cout<<"Grade:"<<**endl**;

cin>>s[i].grade[j];

**if**(s[i].grade[j]>10||s[i].grade[j]<1)

cout<<"Grade should be in range 1 to 10"<<**endl**;

}**while**(s[i].grade[j]>10||s[i].grade[j]<1);

sum\_of\_product+=(s[i].credit[j]\*s[i].grade[j]);

}

s[i].sgpa=sum\_of\_product/total\_credit;

}

}

//definition of method to validate roll number

**bool** **Student\_class::rollNo\_validation**(**int** i)

{

**for**(**int** j=i-1;j>=0;j--)

{

**if**(s[i].roll\_no==s[j].roll\_no)

{

cout<<"Roll number should be unique"<<**endl**;

**return** **false**;

}

}

**return** **true**;

}

//definition of method to validate name

**bool** **Student\_class::name\_validation**(string name)

{

**int** i=0,count=0;

**while**(name[i]!='\0')

{

**if**(**isspace**(name[i]))

count++;

i++;

}

**if**(count==2)

{

i=0;

**while**(name[i]!='\0')

{

**if**(**isalpha**(name[i])||**isspace**(name[i]))

i++;

**else**

**break**;

}

**if**(name[i]=='\0')

**return** **true**;

}

cout<<"Enter valid name"<<**endl**;

**return** **false**;

}

//definition of display

**void** **Student\_class::display**(**int** i)

{

cout<<i+1<<"\t"<<s[i].roll\_no<<"\t"<<s[i].name<<"\t"<<s[i].sgpa<<**endl**;

}

//definition of sort according to roll number

**void** **Student\_class::sort\_rollNo**(**int** n)

{

**bool** swapped;

student temp;

**for**(**int** i=0;i<n;i++)

{

swapped=**false**;

**for**(**int** j=0;j<n-i-1;j++)

{

**if**(s[j].roll\_no>s[j+1].roll\_no)

{

temp=s[j];

s[j]=s[j+1];

s[j+1]=temp;

swapped=**true**;

}

}

cout<<"Pass "<<i+1<<":"<<**endl**;

**for**(**int** k=0;k<n;k++)

display(k);

**if**(!swapped)

**break**;

}

cout<<"List sorted successfully"<<**endl**;

}

//definition of sort alphabetically

**void** **Student\_class::sort\_aplhabetically**(**int** n)

{

**int** i,j;

student temp1;

**for**(i=1;i<n;i++)

{

temp1=s[i];

j=i-1;

**while**(j>=0&&s[j].name>temp1.name){

s[j+1]=s[j];

j--;

}

s[j+1]=temp1;

cout<<"Pass "<<i<<":"<<**endl**;

**for**(**int** k=0;k<n;k++)

display(k);

}

}

//definition of search using SGPA

**void** **Student\_class::search\_SGPA**(**int** n)

{

**double** key;

**bool** found=**false**;

cout<<"Enter SGPA to be search:";

cin>>key;

**for**(**int** i=0;i<n;i++)

{

**if**(s[i].sgpa==key)

{

display(i);

found=**true**;

}

}

**if**(!found)

cout<<"No student with SGPA "<<key<<" found"<<**endl**;

}

//definition of search according to name

**void** **Student\_class::search\_name**(**int** n)

{

string key;

cin.ignore();

cout<<"Enter search key:";

**getline**(cin,key);

**int** low=0,high=n-1,mid;

**bool** found=**false**;

**while**(low<=high)

{

mid=low+(high-low)/2;

**int** x=s[mid].name.compare(key);

**if**(x==0)

{

found=**true**;

display(mid);

**break**;

}

**else** **if**(x>0)

high=mid-1;

**else**

low=mid+1;

}

**if**(!found)

cout<<"Student with name '"<<key<<"' not found"<<**endl**;

}

//definition of sort with SGPA method

**void** **Student\_class::sort\_sgpa**(**int** left,**int** right)

{

**static** **int** pass=0;

**static** **int** n=right+1;

**if**(left>=right)

**return**;

**int** i=left;

**int** j=right+1;

student pivot=s[left];

**while**(1)

{

**do**{

i++;

}**while**(s[i].sgpa<pivot.sgpa);

**do**{

j--;

}**while**(s[j].sgpa>pivot.sgpa);

**if**(i>=j)

**break**;

**else**{

student temp=s[i];

s[i]=s[j];

s[j]=temp;

}

}

s[left]=s[j];

s[j]=pivot;

cout<<"Pass "<<pass<<":"<<**endl**;

**for**(**int** k=0;k<n;k++)

display(k);

sort\_sgpa(left,j-1);

sort\_sgpa(j+1,right);

}

//definition of function to display topper

**void** **Student\_class::display\_topper**(**int** n)

{

**int** top\_num;

sort\_sgpa(0,n-1);

cout<<"Enter number of toppers to be display:";

cin>>top\_num;

**if**(top\_num>n)

cout<<"only "<<n<<" records available"<<**endl**;

**else**

**for**(**int** i=n-1;i>=n-top\_num;i--)

display(i);

}

//driver function

**int** **main**()

{

Student\_class obj;

**int** n,choice;

cout<<"Enter number of students:";

cin>>n;

obj.input(n);

**do**

{

cout<<"------------------------------------------------------------"<<**endl**;

cout<<"1:Display\n2:Sort list according to roll number\n3:Sort list alphabetically\n4:Sort with SGPA\n5:Search student with SGPA\n6:Search student according to name\n7:Display toppers\n8:Exit"<<**endl**;

cout<<"Enter choice:";

cin>>choice;//enter choice of user

cout<<"-------------------------------------------------------------"<<**endl**;

**switch**(choice)

{

**case** 1:

cout<<"SrNo\tRoll No\t\tName\tSGPA"<<**endl**;

**for**(**int** i=0;i<n;i++)

obj.display(i);

**break**;

**case** 2:

obj.sort\_rollNo(n);

**break**;

**case** 3:

obj.sort\_aplhabetically(n);

cout<<"Sorted Successfully"<<**endl**;

**break**;

**case** 4:

obj.sort\_sgpa(0,n-1);

cout<<"Sorted Successfully"<<**endl**;

**break**;

**case** 5:

obj.search\_SGPA(n);

**break**;

**case** 6:

obj.sort\_aplhabetically(n);

obj.search\_name(n);

**break**;

**case** 7:

obj.display\_topper(n);

**break**;

**case** 8:

cout<<"Thank You"<<**endl**;

**break**;

**default**:

cout<<"Enter valid choice"<<**endl**;

}

}**while**(choice!=8);

**return** 0;

}