**ASSIGNMENT 1**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<iostream.h>

using namespace std;

class weather

{ int high,low,snow,rain;

public:

friend void avg(weather \*k[]);

void displayed()

{ cout<<"\nenter the HIGHEST TEMPREATURE - ";

cout<<high;

cout<<"\nenter the lowest tempreature - ";

cout<<low;

cout<<"\nenter the amount of snow fall - ";

cout<<snow;

cout<<"\nenter the amount of rain fall - ";

cout<<rain; }

void input()

{ cout<<"\nenter the HIGHEST TEMPREATURE - ";

cin>>high;

cout<<"\nenter the lowest tempreature - ";

cin>>low;

cout<<"\nenter the amount of snow fall - ";

cin>>snow;

cout<<"\nenter the amount of rain fall - ";

cin>>rain; }

weather(void)

{ high=0; low=0; snow=0; rain=0;}

~weather()

{ cout<<"deleted SUCCESS-FULLY";}

};

weather \*a[30];

int b=0;

int create(void)

{ if(b==29)

{ cout<<" SORRY! RECORD OVERFLOW " ;

return 0;}

else

{ a[b]=new weather;

a[b]->input();++b; }

return 0; }

int modify()

{ int g

cout<<"enter the date of modfiction - ";

cin>>g;

a[g-1]->input();

return 0; }

int deletes()

{ if(b==0)

{ cout<<"UNDERFLOW! NO FURTHER DELETE poSSIBLE ";

getch();

return 0;}

int g;

cin>>g;

if(g>b)

{cout<<" data filled upto date"<<b+1; }

for(int i=g-1;i<b-1;i++)

a[i]=a[i+1];

delete a[b-1]; --b;

return 0; }

void display()

{ cout<<"\nenter date to display particular record else enter 100 for all";

int t;cin>>t;

if(t==100)

for(int i=0;i<b;i++)

{ cout<<endl<<"DATE - "<<i+1;

a[i]->displayed();}

else

a[t-1]->displayed();}

void avg(weather \*k[])

{ float e=0,f=0,g=0,h=0;

for(int i=0;i<b;i++)

{ e=e+k[i]->high; f=f+k[i]->low; g=g+k[i]->snow; h=h+k[i]->rain; }

cout<<endl<<endl<<"the average of high tempreature - "<<e;

cout<<endl<<"the average of low tempreature - "<<f;

cout<<endl<<"the average of snow fall - "<<g;

cout<<endl<<"the average of rain fall - "<<h; }

int main()

{ cout<<"MENU\n" <<"PRESS 1 to create new record "<<"\nPRESS 2 to modify the record ";

cout<<"\nPRESS 3 to DELETE record "<<"\nPRESS 4 to find average "<<"\nPRESS 5 to DISPLAY RECORD "<<"\n\n..::ENTER ANY OTHER NUMBER TO EXIT::.." ;

int c; cin>>c;

do

{ switch(c)

{case 1: create();break;

case 2: modify();break;

case 3: deletes();break;

case 4: avg(a);break;

case 5: display();break;

}}while(c<=5);

}

**ASSIGNMENT 2**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

using namespace std;

class bookshop

{ string title, author, publisher;

double price; int available;

public:

static int transaction;

static int untransaction;

bookshop (){}

bookshop(int iop)

{ bookshop p; int i=-1; char a; int aw;

cout<<" ENTER THE NUMBER OF DATA YOU WANT TO ENTER - ";

cin>>aw;

bookshop \*\*qw=new bookshop\*[aw];

printf(" \n\n\n\t\t\t<< MENU >>\n\n1.CREATE AND INSERT\n2.DISPALY ALL \n3.DISPLAY PARTICULAR\n4.MODIFY\n5.DELETE\n6.ACCOUNT SECTION\t\t");

a=getch();

string h;

int r;

switch(a)

{case '1': ++i;

qw[i]=p.create();

for(int o=0;o<i;o++)

{if(qw[o]->title==qw[i]->title && qw[o]->author==qw[i]->author && qw[o]->publisher==qw[i]->publisher)

{

cout<<"\n\n\n\t\t\t DATA ALREADY EXIST NOT CREATED NEW DATA available stock updated ";

qw[o]->available+=qw[i]->available;

qw[o]->price=qw[i]->price;

delete qw[i--]; }}break;

case '2':

for(int j=0;j<=i;j++)

qw[j]->display();

case '3':

cout<<"ENTER DISPLAY BY TITLE - ";

cin>>h;

for(int iq=0;iq<=i;iq++)

{ if(qw[iq]->searches(h))

{qw[iq]->display();break;}}

case '4':

cout<<"ENTER DATA TO BE MODIFIED BY TITLE - ";

cin>>h;

for(int iq=0;iq<=i;iq++)

{ if(qw[iq]->searches(h))

{ delete qw[iq];

qw[iq]=qw[iq]->create();

qw[iq]->display();}

break;}

case '5':

cout<<"ENTER DATA TO BE DELETED BY TITLE - ";

cin>>h; int r;

for( int iq=0;iq<=i;iq++)

{ if(qw[iq]->searches(h))

{ for(r=iq;r<=i;r++)

qw[r]=qw[r+1];

delete qw[i--];

break;}}

case '6':

accounts(qw,i);}}

bookshop \* create()

{ bookshop \*abc=new bookshop;

cout<<"\n\n\nENTER THE TITLE - ";

cin>>abc->title;

cout<<"\nENTER THE AUTHOR ";

cin>>abc->author;

cout<<"\nENTER THE PUBLISHER ";

cin>>abc->publisher;

cout<<"\nENTER THE PRICE ";

cin>>abc->price;

cout<<"\nENTER THE AVAILABLE STOCK ";

cin>>abc->available;

return abc;}

int searches(string a)

{if(a==title)

return 1; return 0; }

void display()

{ cout<<"\n\n\nTITLE - ";

cout<<title; cout<<"\nAUTHOR ";

cout<<author<<"\nPUBLISHER "<<publisher<<"\nPRICE "<<price<<"\nAVAILABLE STOCK "<<available;}

double cost(int n)

{ if(available<n)

return 0;

if(available>=n)

{available-=n;}

return price\*n;}};

int bookshop::transaction=0;

int bookshop::untransaction=0;

void accounts(bookshop \*\*a,int i)

{ cout<<"\n\n\nENTER THE BOOK TITLE TO TO BE PURCHASED ";

string fg;cin>>fg; int q=0;

for(q=0;q<=i;q++)

{ if(a[q]->searches(fg))

a[q]->display();}

cout<<"\n\n\nENTER THE NUMBER OF COPIES YOU WANT TO PURCHASE - ";

int n; cin>>n;

double r=a[q]->cost(n);

if(r==0)

{cout<< ++bookshop::untransaction<<" TRANSACTION FAILED NOT SUFFICIENT STOCK ";}

else

{ bookshop ::transaction++;

cout<<" TRANSACTION SUCCESSFULL \nTRANSACTION NUMBER : "<<bookshop::transaction<<endl;

cout<<" YOU HAVE TO PAY "<<r<<" RUPEES ";}}

int main()

{bookshop programme\_starter\_obcect(1);

return 0;

}

**ASSIGNMENT 3**

#include<iostream>

#include<string>

#include<stdio.h>

#include<conio.h>

using namespace std;

class complexes

{

int real,comp;

public:

friend istream &operator >>(istream &a, complexes &b);

friend ostream &operator <<(ostream &a, complexes &b);

friend void operator +(complexes a,complexes b);

friend void operator -(complexes a,complexes b);

void operator <(complexes a)

{ if(a.real<a.comp)

cout<<"its true REAL IS LESS THAN IMAGINARY ";

else

cout<<" FALSE ";}

void operator >(complexes a)

{ if(a.real>a.comp)

cout<<"its true REAL IS GREATER THAN IMAGINARY ";

else

cout<<" FALSE ";}

void operator ==(complexes a)

{ if(a.real==a.comp)

cout<<"its true REAL IS EQUAL TO IMAGINARY ";

else

cout<<" FALSE ";}

void operator !=(complexes a)

{ if(a.real!=a.comp)

cout<<"its true REAL IS NOT EQUAL IMAGINARY ";

else

cout<<" FALSE ";}

void operator -()

{ cout<<-real;

if(-comp>0)

cout<<"+"<<-comp;

else

cout<<-comp;}

void operator \* (complexes a)

{ cout<<(real\*a.real-comp\*a.comp)<<" + "<<real\*a.comp+comp\*a.real<<" i ";}

void operator / (complexes a)

{ a.comp=-a.comp;

cout<<(real\*a.real-comp\*a.comp)<<" + "<<real\*a.comp+comp\*a.real<<" i ";}};

istream &operator >>(istream &a, complexes &b)

{ a>>b.real;

a>>b.comp;}

ostream &operator <<(ostream &a, complexes &b)

{ cout<<"\n\n";

a<<b.real;

cout<<" + ";

a<<b.comp;}

void operator -(complexes a,complexes b)

{ cout<<" YOUR ANSWER IS ";

cout<<a.real-b.real<<" + "<<a.comp-b.comp;}

void operator +(complexes a,complexes b)

{ cout<<" YOUR ANSWER IS ";

cout<<a.real+b.real<<" + "<<a.comp+b.comp;}

void unary()

{ complexes a;

cout<<"\n\n\n ENTER THE DATA REAL AND THE IMAGINARY PART RESPECTIVELY : " ;

cin>>a;

cout<<"\n\n\nYOUR DATA IS ";

cout<<a;

cout<<" \n\nNEW DATA IS ";|}

void binary()

{ complexes a,b;

cout<<"\n\n\n ENTER THE DATA REAL AND THE IMAGINARY PART RESPECTIVELY : ";

cin>>a;

cout<<"\n\n\nYOUR DATA 1 IS ";

cout<<a;

cout<<"\n\n\n ENTER THE DATA REAL AND THE IMAGINARY PART RESPECTIVELY : ";

cin>>b;

cout<<"\n\n\nYOUR DATA 2 IS ";

cout<<b;

cout<<"\n\n\n 1. TO ADD \n2. TO SUBTRACT \n3. TO MULTIPLY \n4. TO DIVIDE\n\n\n";

char s;

s=getch();

switch(s)

{

case '1':

a+b;

break;

case '2':

operator - (a,b);

break;

case '3':

a\*b;

break;

case '4':

a/b;

break;}}

void relational()

{ complexes a,b;

cout<<"\n\n\n ENTER THE DATA REAL AND THE IMAGINARY PART RESPECTIVELY : ";

cin>>a;

cout<<"\n\n\nYOUR DATA IS “<<a<<"\n\n\n 1. GREATER THAN \n2. LESS THAN \n3. EQUAL \n4. NOT EQUAL\n\n\n";

char s;s=getch();

switch(s)

{

case '1':

a<a;break;

case '2':

a>a;break;

case '3':

a==a;break;

case '4':

a!=a;break;}}

int main()

{

do

{cout<<"\n\n\n\t\t\t\t\t..:: MENU ::..\n\n”<<"1.UNARY\n2.BINARY \n3. RElaTIONAL\n\n\n\t\t\t";

char c;c=getch();

switch(c)

{case'1':

unary(); break;

case '2':

binary(); break;

case '3':

relational();break;

}

return 0;

}

}while(c<’3’);

**ASSIGNMENT 4**

#include<iostream.h>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

class data

{ protected:

string name, dob;

char bgroup;

public:

data()

{ name="XXXXX";

dob="00-00-0000";

bgroup='x';}

data(string a,string b,char c)

{ name=a;dob=b;bgroup=c;}

};

class data2

{ protected:

int weight, height;

public:

data2()

{ weight=0;height=0;}

data2(int a,int b)

{ weight=a;height=b;}

};

class data3

{ protected:

int policy;

char address[100];

public:

data3()

{ policy=0;

strcpy(address," ");}

data3(int a,char b[])

{ policy=a;

strcpy(address,b);}};

class data4:protected data,protected data2,protected data3

{ string telephone;

int licence;

public:

void displayed()

{ cout<<endl<<"\t\t"<<name<<"\t"<<dob<<"\t"<<bgroup<<"\t"<<weight<<"\t"<< height<<"\t"<<policy<<"\t"<<address<<"\t"<<telephone<<"\t"<<licence;}

string searches()

{return name;}

string searchess()

{return telephone;}

data4(string a,string b,char c,int d,int e,int f,char g[],string h,int i):

data(a,b,c),data2(d,e),data3(f,g)

{telephone=h;licence=i;}

void entry()

{ cout<<"\n\n\nENTER THE NAME OF CANDIDATE - ";

cin>>name;

cout<<"\nENTER THE DATE OF BIRTH ";

cin>>dob;

cout<<"\nENTER THE BLOOD GROUP ";

cin>>bgroup;

cout<<"\nENTER THE WEIGHT OF CANDIDATE ";

cin>>weight;

cout<<"\nENTER THE HEIGHT OF CANDIDATE ";

cin>>height;

cout<<"\nENTER THE POLICY NUMBER ";

cin>>policy;

cout<<"\nENTER TE ADDRESS ";

scanf("%s",address);

cout<<"\nENTER THE TELEPHONE NUMBER ";

cin>>telephone;

cout<<"\nENTER THE DRIVING LICENCE NUMBER ";

cin>>licence;}

data4 \*node;

~data4()

{ cout<<"daat deleted";}

};

data4 \* create()

{ string a;string b;char c;int d;int e;int f;char g[100];string h;int i;

cout<<"\n\n\nENTER THE NAME OF CANDIDATE - ";

cin>>a;

cout<<"\nENTER THE DATE OF BIRTH ";

cin>>b;

cout<<"\nENTER THE BLOOD GROUP ";

cin>>c;

cout<<"\nENTER THE WEIGHT OF CANDIDATE ";

cin>>d;

cout<<"\nENTER THE HEIGHT OF CANDIDATE ";

cin>>e;

cout<<"\nENTER THE POLICY NUMBER ";

cin>>f;

cout<<"\nENTER TE ADDRESS ";

scanf("%s",g);

cout<<"\nENTER THE TELEPHONE NUMBER ";

cin>>h;

cout<<"\nENTER THE DRIVING LICENCE NUMBER ";

cin>>i;

data4 \* abc=new data4(a,b,c,d,e,f,g,h,i);

return abc;}

void display(data4 \* p,int i)

{printf("\n\n\n\nNAME\tDOB\tBGROUP\tWEIGHT\tHEIGHT\tPOLICY\tADDRESS\tTELEPHONE\tDRIVE");

if(i==0)

{ while(p!=NULL)

{ p->displayed();

p=p->node;}}

string ser;

if(i==1)

{ cout<<"\n\nENRER THE NAME OF CANDIDATE TO BE SEARCHED ";

cin>>ser;

cout<<endl<<endl;

while(p!=NULL)

{ if(p->searches()==ser)

{p->displayed();}

p=p->node;}}}

data4 \* deletes(data4 \* f)

{ int i,j;

struct data4 \*q=f,\*r=f,\*p=NULL;

if(f==NULL)

{ printf(" \n\n\tNO DATA FURTHER DELETION NOT POSSIBLE ");

return 0;}

printf("\n\nENTER THE RECORD NUMBER TO BE DELETED start from 1");

scanf("%d",&i);

i=i-1;

if(i==0)

{ r=r->node;

free(f);

f=r;

return f;}

for(j=1;j<i;j++)

{ q=q->node;

r=r->node;}

r=r->node;

r=r->node;

p=q->node;

q->node=r;

free(p);

return f;}

int edit(data4 \* q)

{ string qq,rr;

cout<<"\n\n\n ENTER THE NAME OF DATA TO BE EDITED ";

cin>>qq;

cout<<"\n\nENTER THE TELEPHONE NUMBER TO BE EDITED ";

cin>>rr;

data4 \*a=q;

int o=0;

while(a!=NULL)

{ if(qq==a->searches()&&rr==a->searchess())

{ a->entry();

return 0;}

a=a->node;}

if(o==0)

printf("\n\n\n\t\t\t.. SORRY NO DATA FOUND ");

return 0;

}

int main()

{ data4 \*p=NULL , \*starts=NULL;char a;

do

{printf(" \n\n\n\t\t\t<< MENU >>\n\n1.CREATE AND INSERT\n2.DELETE \n3.EDIT\n4.DISPLAY\n5.DISPLAY ALL\n\t\t");

a=getch();

switch(a)

{case '1':

p=create();

if(starts==NULL)

{starts=p;}

else

{ p->node=starts;

starts=p;

}break;

case '2':

display(starts,0);

starts= deletes(starts);break;

case '3':

edit(starts);break;

case '4':

display(starts,1);

break;

case '5':

display(starts,0);

break;}return 0;}

**ASSIGNMENT 5**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

using namespace std;

template <class A>

class matrix

{ A \*\*b, \*\*a;int c,d,e,f;

public:

void create()

{ cout<<"\n\nENTER THE SIZE OF MATRIX 1 - ";

cin>>c>>d;

b=new A\* [c];

for(int i=0;i<c;i++)

b[i]=new A [d];

cout<<"\n\nENTER THE SIZE OF MATRIX 2 - ";

cin>>e>>f;

a=new A\* [e];

for(int i=0;i<e;i++)

a[i]=new A [f];}

void dataentry()

{ cout<<"\n\n\nENTER THE MATRIX 1:- ";

for(int i=0;i<c;i++)

{ for(int j=0;j<d;j++)

{ cin>>b[i][j];}}

cout<<"\n\n\nENTER THE MATRIX 2:- ";

for(int i=0;i<e;i++)

{ for(int j=0;j<f;j++)

{ cin>>a[i][j];}}}

void display(int q)

{ if(q==1)

{ for(int i=0;i<e;i++)

{ for(int j=0;j<f;j++)

{ cout<<b[i][j];}}}

else

{ for(int i=0;i<e;i++)

{ for(int j=0;j<f;j++)

{ cout<<a[i][j];}}}

}

void addition()

{ int i,j;

if(c==e&&d==f)

{for( i=0;i<c;i++)

{ printf("\n");

for( j=0;j<d;j++)

{cout<<a[i][j]+b[i][j];}

}}

else

printf("..::SORRY! NO ADDITION POSSIBLE ::..");}

void subtraction()

{int i,j;

if(c==e&&d==f)

{

for( i=0;i<c;i++)

{ printf("\n");

for( j=0;j<d;j++)

{cout<<a[i][j]-b[i][j];}

}

}

else

printf("..::SORRY! NO SUBTRACTION POSSIBLE ::..");

}

int multi()

{int i,j,k;

float temp,abg=0;

if(e!=d)

{printf("SORRY! no multiplication possible : ");}

for(i=0;i<c;i++)

{printf("\n");

for(j=0;j<d;j++)

{ abg=0;

for(k=0;k<f;k++)

{ abg=abg+ (a[i][k] \* b[k][j]);}

cout<<abc;}

}

return 0;

}

};

template <class QWER>

void INTEGER( QWER cd)

{matrix <QWER > aa;

aa.create();

aa.dataentry();

char a;

ab:

cout<<"\n\n\n\t\t\tMENU\n\n\n1.ADDITION\n2.MULTIPLICATION\n3.SUBTRACTION\n\n";

printf("\n\n\n\t\tNOW ENTER YOUR CHOICE : ");

a=getch();

switch(a)

{

case '1':

aa.addition();

break ab;

case '2':

aa.multi();

break ab;

case'3':

aa.subtraction();

break ab;

}

getch();

}

int main()

{ ab:

cout<<"\n\n\n\t\t\t..:: MENU ::..\n\n1. INTERGER MATRIX MANIPULATION\n2. FLOAT MATRIX MANIPULATION\n\n\t\t";

char c;

c=getch();

switch(c)

{

case '1':

INTEGER(1);

break ab;

case '2':

INTEGER(1.1f);

break ab;

}

return 0;

}

**ASSIGNMENT 6**

#include<iostream.h>

#include<conio.h>

using namespace std;

class shape

{ protected:

double a;

double b;

public:

shape()

{a=0;b=0;}

Virtual void area()=0;

virtual void getdata(){}

};

class triangle:public shape

{public:

void getdata()

{ cout<<"ENTER THE HEIGHT - ";

cin>>a;

cout<<"\n\nENTER THE BASE \_ ";

cin>>b;}

void area()

{cout<<"AREA OF THE TRIANGLE \_ ";

cout<<0.5\*a\*b;

}

};

class rectangle:public shape

{ public:

void getdata()

{cout<<"ENTER THE HEIGHT - ";

cin>>a;

cout<<"\n\nENTER THE width - ";

cin>>b;

}

void area()

{ cout<<"AREA OF THE RECTANGLE - ";

cout<<a\*b;}

};

class circle:public shape

{public:

void getdata()

{cout<<"ENTER THE RADIUS - ";

cin>>a;}

void area()

{ cout<<"AREA OF THE CIRCLE - ";

cout<<3.14\*a\*;}

};

class square:public shape

{public:

void getdata()

{cout<<"ENTER THE side - ";

cin>>a;

}

void area()

{cout<<"AREA OF THE CIRCLE - ";

cout<<a\*a;} };

int main()

{ char a='1';

shape \*r;

circle s;

triangle t;

rectangle u;

square v;

while(a<='4')

{cout<<"\n\n\n\t\t\t..::MENU::..\n1.RECTANGLE\n2.TRIANGLE\n3.CIRCLE\n4.SQUARE\n\n\n\n\t\t\t";

a=getch();

switch(a)

{case '1':

r=&u;

r->getdata();

r->area();

break;

case '2':

r=&t;

r->getdata();

r->area();

break;

case '3':

r=&s;

r->getdata();

r->area();break;

case'4':

r=&v;

r->getdata();

r->area();break;}}

return 0;}

**ASSIGNMENT 7**

#include<iostream>

#include<conio.h>

using namespace std;

void doub(double &c,double &d)

{ cin>>c>>d;

}

double fun(double b)

{ if(b==0)

throw b;

else

return b;

}

int main()

{ double a,b;

cout<<"\n\n\nENTER TWO NUMBERS TO FIND DIVISION ";

doub(a,b);

a=a/b;

cout<<a;

/\*try

{

if(b==0)

fun(b);

}

catch(double a)

{

cout<<"\n\n\n\t\t\tSECOND VALUE IS "<<a;

cout<<"\n\nEXECUTION NOT COMPLETED \n\nPRESS ANY KEY TO EXIT";

getch();

// return 0;

}

catch(...)

{

cout<<"executed";

}

\*/

return 0;

}

ASSIGNMENT 8

#include<fstream>

#include<stdio.h>

using namespace std;

class student

{ private:

int roll;

char name[30];

int clas;

int marks;

public:

student()

{

roll=0;

clas=0;

marks=0;

}

student(int rolled,char nam[], int cla, int mar)

{

roll=rolled;

strcpy(name,nam);

clas=cla;

marks=mar;

}

void display()

{

cout<<"\n\nNAME - ";

puts(name);

cout<<"ROLL NUMBER "<<roll<<"\nCLASS - "<<clas<<"\nMARKS - "<<marks;

}

int rol()

{

return roll;

}

int getdata()

{

cout<<"\n\n\n\t\t\tENTER THE REQUIRED DATA : - ";

cout<<"\nROLL NUMBER ";

cin>>roll;

fstream file;student r;

file.open("DATAFILE.dat",ios::in);

while(file.read((char\*)&r,sizeof(r)))

{

if(r.rol()==roll)

{

cout<<" ROLL NUMBER ALREADY EXIST NO DATA CREATED ";

break abc;}

}

file.close();

cout<<"\n\nNAME - ";

cin>>name;

cout<<"\nENTER THE CLASS ";

cin>>clas;

cout<<"\nTHE THE MARKS ";

cin>>marks;

return 1;

return 0;

}

};

int disp(int c,int p)

{ student r;

int y=0;

fstream file("DATAFILE.txt",ios::in|ios::out);

fstream file2("DATAFILE.txt",ios::in|ios::out);

cout<<"\n\n";

while(file.read((char\*)&r,sizeof(r)))

{

if(r.rol()==c && p==0 )

{ r.display(); break;}

if(r.rol()==c && p==1)

{

file.seekg(y);

r.getdata();

file.write((char\*)&r,sizeof(r));

break;

}

file2.read((char\*)&r,sizeof(r));

y=file2.tellg();

}

file.close();

file2.close();

return 0;

}

int displ(int c)

{student r;

fstream file("DATAFILE.txt",ios::in);

fstream file2("DATAFILEs.txt",ios::out|ios::app);

while(file.read((char\*)&r,sizeof(r)))

{if(r.rol()!= c)

{ file2.write((char\*)&r,sizeof(r));}}

file.close();

file2.close();

remove("DATAFILE.txt");

rename("DATAFILEs.txt","DATAFILE.dat");

return 0;

}

int main()

{ ab:

student r,s;

int c;

fstream file;

cout<<"\n\n\n\t\t\t\t..:: MENU ::..\n\n";

cout<<"1. CREATE NEW DATA \n2. DISPLAY ALL\n3. DISPLAY \n4. MODIFY\n5. DELETE ";

char a;

a=getch();

switch(a)

{ case '1':

c= r.getdata();

if(c==0)

{ break ;}

file.open("DATAFILE.txt",ios::out|ios::app);

file.write((char\*)&r,sizeof(r));

file.close();

break ;

case '2':

file.open("DATAFILE.txt",ios::in);

cout<<"\n\n";

while(file.read((char\*)&r,sizeof(r)))

{ r.display();

cout<<"\n\n\n";}

file.close();

break ;

case '3':

cout<<"\n\n\n\tENTER THE ROLL NUMBER TO BE DISPLAYED - " ;

cin>>c;

disp(c,0);

break ;

case '4':

cout<<"\n\n\n\tENTER THE ROLL NUMBER TO BE EDITED - " ;

cin>>c;

disp(c,1);

break ;

case '5':

cout<<"\n\n\n\tENTER THE ROLL NUMBER TO BE DELETED - " ;

cin>>c;

displ(c);

break ; }

return 0;

}

**ASSIGNMENT 9**

#include<iostream>

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

#include<iomanip>

class personal

{

protected:

char name[100];

char dob[15];

char bgr;

public:

personal(char a[],char b[],char c)

{

strcpy(name,a);

strcpy(dob,b);

bgr=c;

}

};

class pro:public personal

{ protected:

char post[100];

int salary;

public:

pro(char c[],char d[], char e,char a[],int b):personal(c,d,e)

{ strcpy(post,a);

salary=b;}

};

class acc:public personal

{protected:

char credit;

int project;

public:

acc(char c[],char d[], char e,char a,int b):personal(c,d,e)

{

credit=a;

project=b;

}

};

class bio:public personal,public pro,public acc

{

int gj;

public:

bio(char c[],char d[], char e,char a,int b,char f[],int g,int j):personal(c,d,e),acc(c,d,e,a,b),pro(c,d,e,f,g)

{

}

void display()

{ cout<<"\n\n";

cout<<setw(15)<<pro::name<<" "<<pro::dob<<" "<<pro::bgr<<" "<<post<<" "<<salary<<" "<<credit<<" "<<project;

}

int names(char d[])

{

if( !strcmp(pro::name,d))

{

return 1;}

cout<<"not entered";

return 0;

}

int dobed(char d[])

{

if( !strcmp(pro::dob,d))

return 1;

return 0;

}

};

bio \*\* insertr(bio \*\* p,int&cu ,int r )

{

fun();

char c[100], d[15], e, a;int b;char f[25];int g;

cout<<"\n\n\n\t\t\t...:: MENU ::...\n\nENTER DATA -\n";

cout<<"\n\nENTER NAME OF CANDIDATE - ";

gets(c);

fflush(stdin);

cout<<"\nENTER DATE OF BIRTH - ";

gets(d);

fflush(stdin);

cout<<"\nENTER blood group - ";

cin>>e;

fflush(stdin);

cout<<"\nENTER CREDIT OBTAINED - ";

cin>>a;

fflush(stdin);

cout<<"\nENTER PROJECT NO - ";

cin>>b;

fflush(stdin);

cout<<"\nENTER POST - ";

gets(f);

fflush(stdin);

cout<<"ENTER SALARY - ";

cin>>g;

fflush(stdin);

bio \*t;

t=new bio(c,d,e,a,b,f,g,1);

if(r==0)

{++cu;

p[cu]=(bio \*)realloc((bio\*)p,sizeof(p)\*cu);

p[cu-1]=t;

}

if(r==1)

{

p[cu]=t;

}

return p;

}

bio\*\* modify(bio \*\*a, int& cu,int r)

{

fun();

char name[100],dob[15];

cout<<"\nENtER NAME TO BE SEARCHED - ";

gets(name);

fflush(stdin);

cout<<"\nENtER DATE OF BIRTH - ";

gets(dob);

fflush(stdin);

for(int i=0;i<cu;i++)

{

if(a[i]->names(name) && a[i]->dobed(dob))

{

if(r==0)

{a=insertr(a,i,1);

return a;

}

if(r==1)

{ fun();

bio\* o=new bio("a","a",'a','a',1,"a",1,1);

o=a[cu-1];

a[cu-1]=a[i];

a[i]=o;

free(a[cu-1]);

--cu;

a[cu]=(bio \*)realloc((bio\*)a,sizeof(a)\*cu);}}}

return a;

}

int main()

{

fun();

int c=0;

bio \*\*p;

p=(bio\*\*)malloc(sizeof(bio\*)\*0);

ab:

cout<<"\n\n\n\t\t\t..:: MENU ::..\n\n";

cout<<"1. CREATE DATA \n2. DISPLAY ALL \n3. MODIFY \n4.DELETE";

char a;

a=getch();

switch(a)

{

case'1':

p=insertr(p,c,0);

break ab;

case'2':

for(int i=0;i<c;i++)

{p[i]->display();

cout<<endl;

}break ab;

case'3':

p=modify(p,c,0);

break ab;

case '4':

p=modify(p,c,1);

break ab;

}

return 0;

}

**ASSIGNMENT 10**

#include<stdio.h>

#include<conio.h>

#include<stdlib.h>

#include<fstream>

#include<iostream>

using namespace std;

class student

{public:

char name[50];

int rno;

int classs;

};

int display(student w)

{

cout<<"\n"<<w.name<<"\t"<<w.rno<<"\t"<<w.classs;

}

student getdata(student w)

{ printf("\n\n\nenter the name of student - ");

scanf("%s",w.name);

printf("\nenter the roll number ");

scanf("%d",&w.rno);

printf("\nenter the class ");

scanf("%d",&w.classs);

return w;

}

int main()

{

student aa;

fstream b;

system("color f0");

cout<<"MENU\n\n"<<"1.create and insert\n"<<"2.retrive";

char c;

do

{ c=getch();

switch(c)

{

case '1':

b.open("abc.txt",ios::out);

getdata(aa);

b.write((char\*)&aa,sizeof(aa));

b.close();

break;

case '2':

b.open("abc.txt",ios::in);

while(b.read((char\*)&aa,sizeof(aa)))

{

display(aa);

}

b.close();

break;

}

}while(c<=

'2');

}