

### Assignment 3.

class - SEIV      Roll No - 21430  
Batch - F4      Date - 10/10/2023  
Length - 90 min. + 10 minute break.

#### Problem Statement:

Develop an object oriented program in C++ to create a database of personal information system containing the following information: Name, Date of Birth, Blood group, Height, weight, address, telephone number, license number etc. construct database with suitable member function for initializing & destroying data viz: constructor, copy, constructor, static members functions, friend class, this pointer, inline code & dynamic memory allocation - delete, new.

#### Objectives:-

To learn the concept of constructor, default constructor, copy, destructor, static member function, friend class, this pointer, inline code and dynamic memory allocation operators - new and delete.

#### Outcomes:-

I will be able to implement constructor & its types, destructor, static member functions, friend class, this pointer, inline code & dynamic memory allocation.

### 3. Lazarus

Software & Hardware Requirements:-

1. 64-bit Open source Linux or its derivatives
2. open source c++ programming tool like EIDE/GCC.
3. open source c++ IDE Eclipse

Theory :-

(i) Constructor :- Before this is also known as constructor. A special method of class that will be automatically invoked when an instance of class is created.

i) Used to initialize data members of class.

ii) Name is same as class name. It returns nothing.

iii) Automatically called when object of class is created.

iv) Don't have return type starting with void.

⇒ @Default constructor -

Also known as empty constructor, which has no arguments and all datamembers are initialized to zero value.

⇒ @Parameterised constructor -

This is another type of constructor which has some parameters and these parameters are used to initialize data member.

⇒ @Copy constructor -

In this constructor we pass object of same class as a parameter. This is used for copying the values of class object into another object of class.

### ii) Destructor :-

Destructor is used to free up memory by destroying the objects that were created. It is automatically at the end of program. It has also same name as that of class, but with a preceding (~) tildesign. It does not contain parameters or arguments.

### iii) Static Member :-

A class can contain static members or static member functions.

static member functions has only one copy is shared between all objects. It is visible only within class but lifetime is in entire program.

Static data members are class variables because only one value for all obj of some class. To avoid multiple declaration we only include the prototype in class not definition. Initialization is in global scope as it is unique for all objects.

Accessed by objects or by class Name.

static member function has no contructor.

i) Access only static member function in class.

ii) Can be called using class name instead of its object name.

They cannot access non-static members; cannot be declared as virtual, cannot be overloaded with non-static member functions.

E.g. -

## class statictest

```

private int count = 0;
public static int count() {
    return count;
}

```

```

int main() {
    cout << statictest::count();
}
```

Output - 0  
 Explanation - 0 is returned because static variable is initialized to 0.

- iv) **Friend functions** - It is a function which can access private and protected members of a class. If we declare the classes of function as friend, they can access private & protected members from outside the class using friend function concept.
- i) It cannot be called using object of class.
- ii) Can be called as normal function without using object.
- iii) Can be declared in public & private section.
- iv) It cannot directly access datamember, it must use object and dot operator.

Date \_\_\_\_\_  
 Page \_\_\_\_\_

```

class child {
    int x;
public:
    friend void access ( child );
};

void access ( child c )
{
    c.x = 20;
    cout << "x = " << c.x;
}

int main()
{
    child p;
    access(p);
    return 0;
}
  
```

### v) Friend class:-

Like we have friend function, friend class are ones who can access private and protected members of class `fun` in which it has been declared as `friend`.

Eg. `class fun;`  
`class child`  
`{`  
 `int x;`  
 `friend void show();`  
`};`  
`void show()`  
`{`  
 `cout << "x = " << x;`  
`}`

friend class fun;

{;

class fun

{

int b; // blids zero basic

public:

void temp(child c)

{

c.a = 20;

c.show();

}

{;

int main()

{

child a;

fun p;

et

a.temp(p);

return 0;

}

blids zero

; os =

; child

int b;

if

{

(2 blids) zero basic

; os = x.c

confuse x = x"20400

parent for

; 9 blids

((9) zero basic

; o mother

zero basic

zero basic, without basic avoid see axis

ix) Pointers: storing zero in var zero

A pointer is a derived data type that refers to another data variable by storing variables memory address rather than data.

vii) this pointer -

it uses a unique keyword called this to represents an object that invokes a member of class. this pointer points to Obj which calls a + function.

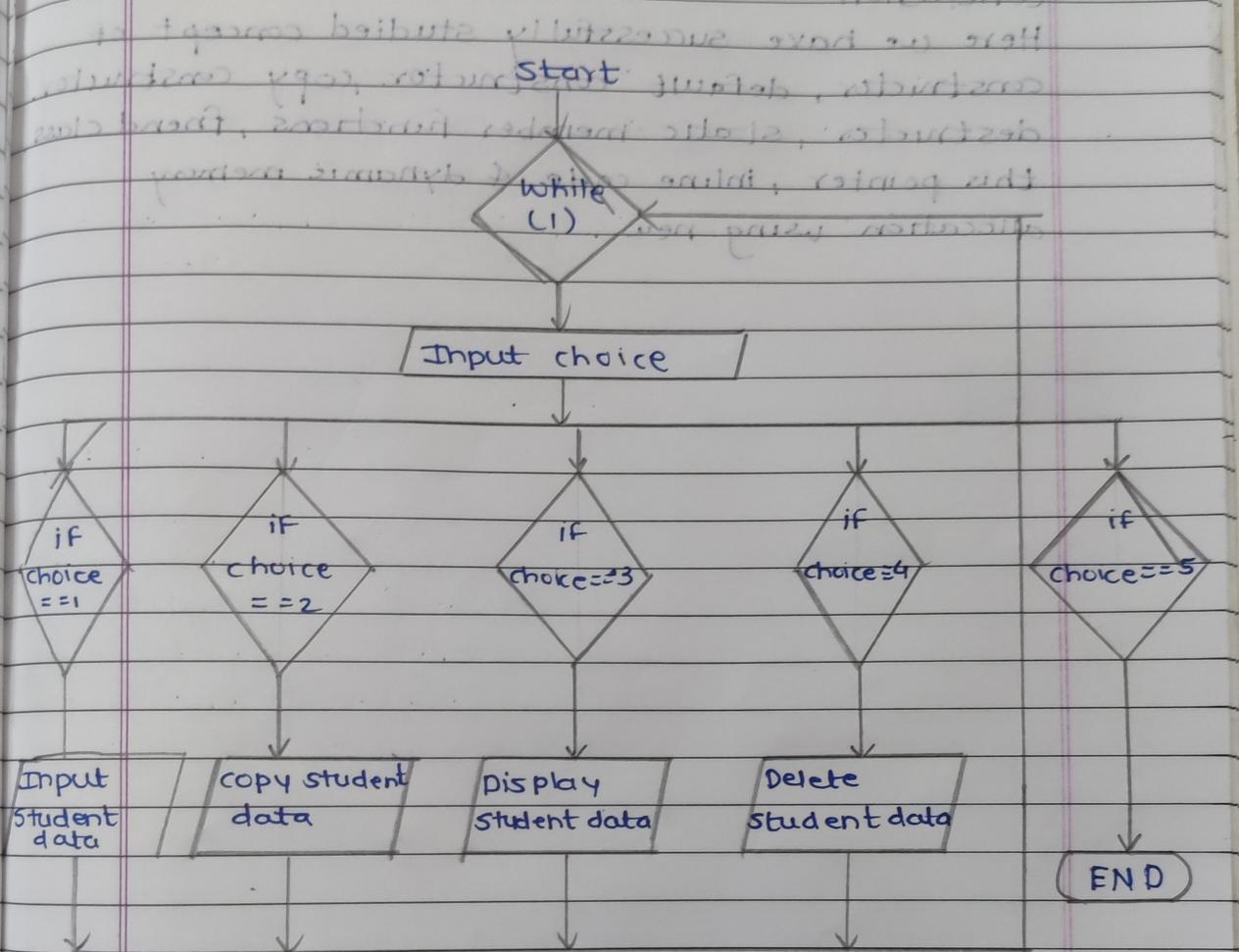
; aux zero

blids zero

parent basic

"x" = x"20400

## PFlowchart :-



## \* Test case :-

Description	I/P	Expected O/P	Actual O/P	Result
i. Giving i/p of person information	Name=Ganesh dob - 14NOV2001 B.G="O+ve" Add="pune" ph.NO - 12345 Lic.NO - 98765	Name is Ganesh DOB - 14 NO 2001 B.G = O+ve Add = Pune Ph.NO = 12345 Lic.NO - 98765	Name is Ganesh DOB - 14 NOV 2001 BG = O+ve Add = Pune Ph.NO = 12345 Lic.NO = 98765	pass.

## Conclusion :-

Here we have successfully studied concept of constructor, default constructor, copy constructor, destructor, static member functions, friend class, this pointer, inline code, & dynamic memory allocation using new. (1)

## Types of operators

Arithmetic operators	Assignment operators	Comparison operators	Logical operators	Bitwise operators	Character operators	Relational operators
operator +	operator =	operator <	operator !	operator ~	operator %	operator ==
operator -	operator +=	operator >	operator &	operator <<	operator >>	operator !=
operator *	operator -=	operator <=	operator	operator ^	operator ==	operator >
operator /	operator *=	operator >=	operator ~&	operator <<=	operator >>=	operator <
operator %	operator /=	operator !=	operator &~	operator  =	operator <<>	operator >><
operator ++	operator /==	operator <<=	operator ^~	operator ^=	operator >>>	operator <<<
operator --	operator *==	operator >>>=	operator &~&	operator <<=	operator >>>=	operator <<<=

END

11.2023

Version

v1.0

Date: 2023-01-11

9/13 modified()

Arithmetic operators: +, -, \*, /, %, ++, --, etc.

Assignment operators: =, +=, -=, \*=, /=, %=, etc.

Comparison operators: &lt;, &gt;, &lt;=, &gt;=, !=, ==, etc.

Logical operators: !, &amp;, |, ~, &amp;~, |~, etc.

Bitwise operators: &lt;&lt;, &lt;&lt;=, &gt;&gt;, &gt;&gt;=, &amp;, |, ^, ~, etc.

```
1 #include <iostream>
2 #include <string.h>
3 using namespace std;
4 class Person
5 {
6     char name[20];
7     class dob
8     {
9         int date;
10        char mon[10];
11        int yr;
12        friend class Student;
13    };
14    char add[20];
15    long long Number;
16    long long licnum;
17    friend class Student;
18};
19
20 class Student
21 {
22     public:
23     static int count;
24     int roll,clss;
25     char div;
26     Person p;
27     Person::dob b;
28     public:
29     Student()
30     {
31         strcpy(p.name," ");
32         roll=0;
33         clss=0;
34         b.date=0;
35         strcpy(b.mon," ");
36         b.yr=0;
```

```
37
38     div=' ';
39     strcpy(p.add,"");
40     p.Number=0;
41     p.licnum=0;
42     count++;
43 }
44 Student(char *nam,int roll,int clss,char d,int dat,char *mont,int y,char *ad,double n,double li)
45 {
46     strcpy(p.name,nam);
47     this->roll=roll;
48     this->clss=clss;
49     b.date=dat;
50     strcpy(b.mon,mont);
51     b.yr=y;
52     div=d;
53     strcpy(p.add,ad);
54     p.Number=n;
55     p.licnum=li;
56 }
57 Student(Student &s)
58 {
59     p=s.p;
60     b=s.b;
61     roll=s.roll;
62     clss=s.clss;
63     div=s.div;
64     count++;
65 }
66 inline void getdata()
67 {
68     cout<<"Enter name of student : ";
69     cin.getline(p.name,19);
70     cout<<"Enter roll no of student : ";
71     cin>>roll;
72     cout<<"Enter class of student : ";
73     cin>>clss;
```

```
73 cout<<"Enter division of student : ";
74 cin>>div;
75 cout<<"Enter date,month and year of birth of student\n";
76 cout<<"Date : ";
77 cin>>b.date;
78 cout<<"Month : ";
79 cin>>b.mon;
80 cout<<"Year : ";
81 cin>>b.yr;
82 cin.ignore(1);
83 cout<<"Enter address of student ";
84 cin.getline(p.add,29);
85 cout<<"Enter phn no. of student ";
86 cin>>p.Number;
87 cout<<"Enter driving liscence no. of student ";
88 cin>>p.licnum;
89 }
90 void showdata()
91 {
92 cout<<"Name of student : ";
93 cout<<p.name<<endl;
94 cout<<"Roll no of student : ";
95 cout<<roll<<endl;
96 cout<<"Class of student : "<<klass<<" Year"<<endl;
97 cout<<"Division of student : ";
98 cout<<div<<endl;
99 cout<<"Date of birth of student : ";
100 cout<<b.date<<" <<b.mon<<" "<<b.yr<<endl;
101 cout<<"Address of student : ";
102 cout<<p.add<<endl;
103 cout<<"Phone Number of student : ";
104 cout<<p.Number<<endl;
105 cout<<"Driving liscence Number of student ";
106 cout<<p.licnum<<endl;
107 cout<<"\n-----" <<endl;
108 }
```

```
109     static int getcount()
110     {
111         return count;
112     }
113     ~Student()
114     {
115     }
116 }
117 };
118 int Student::count=0;
119 int main()
120 {
121     int num,i=0,j;
122     cout<<"How many more Student data you want to add?? \n";
123     cin>>num;
124     Student *s[100];
125     int choice,num1,ch;
126     while(1)
127     {
128
129         cout<<"\nEnter your choice";
130         cout<<"\nEnter 1 to Enter Student data";
131         cout<<"\nEnter 2 for copy Information of Student";
132         cout<<"\nEnter 3 for Dispaly Information of Student";
133         cout<<"\nEnter 4 for Deleting any Student Information";
134         cout<<"\nEnter 5 to Exit\n";
135         cin>>choice;
136
137         switch (choice)
138         {
139             cout<<"Chpice = "<<choice;
140             case 1:
141                 if(i<num)
142                 {
143                     cout<<"-----Enter data of student-----\n\n";
```

Activate Windows  
Go to PC settings to activate Windows.

```
148         p->getdata();
149         *(s[i])=p;
150         i++;
151     }
152     else
153     {
154         cout<<"\nYou have Entered all Student data";
155     }
156
157     break;
158 }
case 2:
{
    cout<<"\nEnter Roll no of student whos data do you want to copy\n";
    cin>>ch;
    int p=0;
    for(j=0;j<num;j++)
    {
        if((s[j])->roll==ch)
        {
            s[i]=new Student();
            *(s[i])=*(s[j]);
            p=1;
            cout<<"\n-----Student data copy Successfully-----\n";
            i++;
            break;
        }
    }
    if(p==0)
    {
        cout<<"Roll no : "<<ch<<" Student does not found \n";
    }
    break;
}
case 3://
```

```
197
198     if(p==0)
199     {
200         cout<<"Roll no : "<<ch<<" Student does not found \n";
201     }
202     break;
203 }
204 case 4:{ 
205     cout<<"\nEnter Roll no of student whose data do you want to Delete\n";
206     cin>>ch;
207     int p=0;
208     for(j=0;j<i;j++)
209     {
210         if((s[j])>roll==ch)
211         {
212             delete s[j];
213             p=1;
214             cout<<"\n-----Student data Deleted successfully-----\n";
215             i--;
216             break;
217         }
218     if(p==0)
219     {
220         cout<<"Roll no : "<<ch<<" Student does not found \n";
221     }
222     break;
223 }
224 case 5:exit(0);
225 default:cout<<"Invalid choice Re-enter your choice\n";
226
227
228 }
229 return 0;
230 }
```

How many more Student data you want to add??

2

Enter your choice

Enter 1 to Enter Student data

Enter 2 for copy Information of Student

Enter 3 for Dispaly Information of Student

Enter 4 for Deleting any Student Information

Enter 5 to Exit

1

-----Enter data of student-----

Enter name of student : ganeh kandepalli

Enter roll no of student : 30

Enter class of student : 1

Enter division of student : a

Enter date,month and year of birth of student

Date : 18

Month : Nov

Year : 2001

Enter address of student Pune

Enter phn no. of student 12345

Enter driving liscence no. of student 98765

Enter your choice

Enter 1 to Enter Student data

Enter 2 for copy Information of Student

Enter 3 for Dispaly Information of Student

Enter 4 for Deleting any Student Information

Enter 5 to Exit

2

Enter Roll no of student whos data do you want to copy

30

-----Student data copy Successfully-----

Enter your choice

Enter 1 to Enter Student data

Enter 2 for copy Information of Student

Enter 3 for Dispaly Information of Student

Enter 4 for Deleting any Student Information

Enter 5 to Exit

3

Enter Roll no of student whos data do you want to Display

30

Name of student : ganeh kandepalli

Roll no of student : 30

Class of student : 1 Year

Division of student : a

Date of birth of student : 18 Nov 2001

Address of student : Pune

Phone Number of student : 12345

Driving liscence Number of student 98765

```
Enter your choice  
Enter 1 to Enter Student data  
Enter 2 for copy Information of Student  
Enter 3 for Dispaly Information of Student  
Enter 4 for Deleting any Student Information  
Enter 5 to Exit  
3
```

```
Enter Roll no of student whos data do you want to Display  
12  
Roll no : 12 Student does not found
```

```
Enter your choice  
Enter 1 to Enter Student data  
Enter 2 for copy Information of Student  
Enter 3 for Dispaly Information of Student  
Enter 4 for Deleting any Student Information  
Enter 5 to Exit  
4
```

```
Enter Roll no of student whose data do you want to Delete  
30
```

```
-----Student data Deleted successfully-----
```

```
Enter your choice  
Enter 1 to Enter Student data  
Enter 2 for copy Information of Student  
Enter 3 for Dispaly Information of Student  
Enter 4 for Deleting any Student Information  
Enter 5 to Exit  
5
```

```
Process exited after 66.24 seconds with return value 0  
Press any key to continue . . .
```