Practical Set-1

Basics Concepts of C++, Tokens Expression and Control structures

P13 Problem: **Define a C++ Structure Rectangle with data member's width and height. It has get_values(member functions to get the data from user and area() member functions to print the area of rectang Also create a C++ Class for the above program. Define both functions inside the class. Member function defined inside the class behaves like an inline function and illustrate the difference between C++ Structure and C++ Class.**

Source Code:

```
// with structure….

#include<iostream>
using namespace std;

struct rectangle{     int
hieght;     int width;
}r;

void get_values(){
   cout<<"enter the value of hieght "<<endl;     cin>>r.hieght;

   cout<<"enter the value of width "<<endl;     cin>>r.width;
}

void area(){
   int AREA=r.width*r.hieght;
   cout<<"the area of the triangle is "<<AREA<<endl;
}

int
main(){     get_values();
area();     return 0;
}
```

```
//with class …..

#include<iostream> using
namespace std;

class
rectangle{    int
hieght,width;
public:    void
get_values(){
      cout<<"enter the value of hieght "<<endl;
cin>>hieght;

      cout<<"enter the value of width "<<endl;
cin>>width;
   }

   void area(){
      int AREA=hieght*width;
      cout<<"the value of area of rectangle is "<<AREA<<endl;
   }
};

int
main(){    rectangle
obj;
obj.get_values();
obj.area();
return 0;
}
```

Errors:

NO

Output:(Screenshot)ctr+alt+printscrn

```
enter the value of hieght
34
enter the value of width
10
the area of the triangle is 340
```

	What You have learned from the Program: **Concept of structure…..**

P14	**Problem:** **Write a C++ program having class Batsman. It has private data members: batsman name, bcode (4 Digit Code Number), innings, not out, runs, batting average. Innings, not out and runs are in integer and batting average is in float.Define following function outside the class using scope resolution operator. 1) Public member function getdata()to read values of data members. 2) Public member function putdata()to display values of data members. 3) Private member function calcavg() which calculates the batting average of a batsman. Also make this outside function inline.** **Hint : batting average = runs/(innings - notout)**

Source Code:

```
#include<iostream>
#include<string>
using namespace std;

class batsman{    string
batsman_name;    int
bcode,innings,not_outs,runs;
float batting_average;
   float calcavg();
public:    void
getdata();    void
putdata();
};

void batsman :: getdata(){
   cout<<"enter name of batsman "<<endl;
getline(cin,batsman_name);
   cout<<"enter bcode of batsman "<<endl;
cin>>bcode;
   cout<<"enter total innings played by "<<batsman_name<<endl;
cin>>innings;
   cout<<"enter not_outs of the "<<batsman_name<<endl;
cin>>not_outs;
   cout<<"enter the total runs of the "<<batsman_name<<endl;
cin>>runs;
}

void batsman :: putdata(){
   cout<<"the batsman is "<<batsman_name<<endl;
   cout<<"the bcode of "<<batsman_name<<" is "<<bcode<<endl;
   cout<<"the total innings played by "<<batsman_name<<" is "<<innings<<endl;
cout<<"the not_outs of "<<batsman_name<<" is "<<not_outs<<endl;
```

```
   cout<<"the total runs of "<<batsman_name<<" is "<<runs<<endl;
   cout<<"the batting average of "<<batsman_name<<" is "<<calcavg()<<endl;
}

inline float batsman :: calcavg(){
   batting_average=runs/(innings-not_outs);
return batting_average;
}

int
main(){    batsman
obj;
obj.getdata();
obj.putdata();
return 0;
}
```

Errors:

NO

Output:(Screenshot)

What You have learned from the Program:
How to make out side function inline function

P15 **Problem:**
Define class Currency having two integer data members rupee and paisa. A class has member fun enter() to get the data and show() to print the amount in 22.50 format. Define one member function that adds two objects of the class and stores answer in the third object i.e. c3=c1.sum (c2). The s member function should add two objects of type currency passed as arguments such that it sup c3.add(c1,c2); where c1, c2 and c3 are objects of class Currency. Also Validate your answer if paisa Write a main( )program to test all the functions. Use concepts of Object as Function Arguments, fu returning object and function overloading.

Source Code:

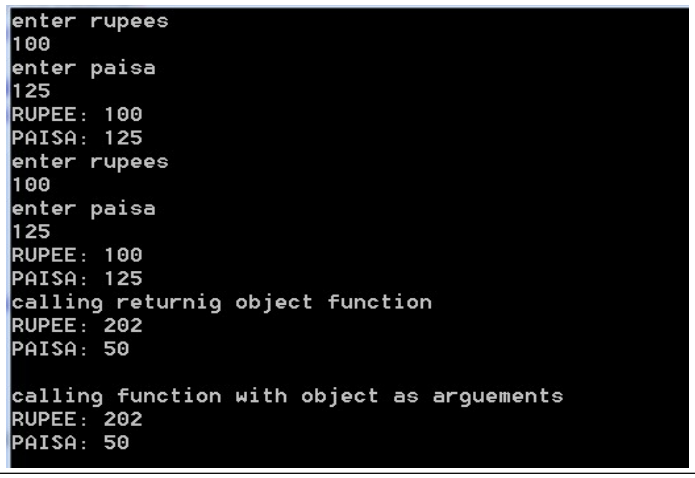
```
#include<iostream>
using namespace std;

class Currency{
   int rupee,paisa;

   public:    void
enter(){
      cout<<"enter rupees"<<endl;       cin>>rupee;
      cout<<"enter paisa"<<endl;
      cin>>paisa;
   }

   void show(){
      cout<<"RUPEE: "<<rupee<<endl;       cout<<"PAISA:
"<<paisa<<endl;
   }

   Currency sum(Currency o1){       Currency o2;
o2.paisa=paisa+o1.paisa;
o2.rupee=rupee+o1.rupee;
while(o2.paisa>=100){          o2.rupee=o2.rupee+1;
         o2.paisa=o2.paisa-100;
      }
      return o2;
   }

   void add(Currency o1,Currency o2){       rupee=o1.rupee+o2.rupee;
```

```
        paisa=o1.paisa+o2.paisa;


while(paisa>=100){          rupee=rupee+1;
          paisa=paisa-100;
       }
    }
};

int main(){
    Currency c1,c2,c3;
    c1.enter();
    c1.show();

    c2.enter();
    c2.show();

    cout<<"calling returnig object function"<<endl;
c3=c1.sum(c2);     c3.show();
    cout<<"\n";

    cout<<"calling function with object as arguements"<<endl;
c3.add(c1,c2);     c3.show();     return 0;
}
```

Errors:

NO

Output:(Screenshot)



What You have learned from the Program:
How to function returning object & concept of object as function arguments…

P16

Define a class Dist with int feet and float inches. Define member function that displays distance in 1'-2.5" format. Also define member function scale ( ) function that takes object by reference and scale factor in float as an input argument. The function will scale the distance accordingly. For example, 20'-5.5" and Scale Factor is 0.5 then answer is 10'-2.75"

Source Code:

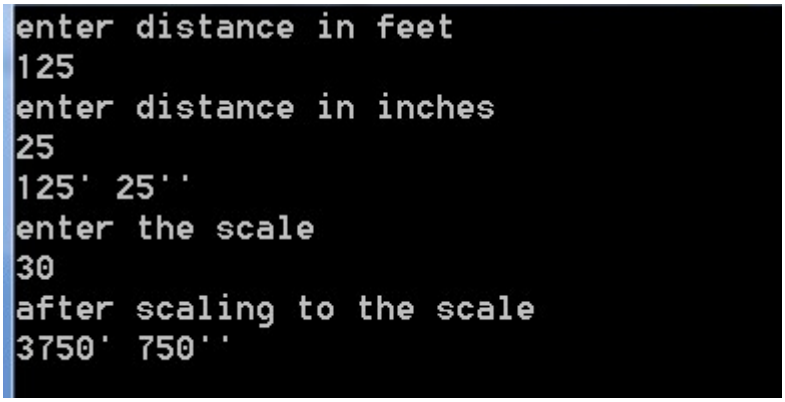
```
#include<iostream>
using namespace std;

class Dist{
   int feet;
   float inches;

   public:

   void enter(){
      cout<<"enter distance in feet "<<endl;
      cin>>feet;
      cout<<"enter distance in inches "<<endl;
      cin>>inches;
   }

   void display(){
      cout<<feet<<"' "<<inches<<"""<<endl;
   }

   void scale(Dist &,float );
};

void Dist :: scale(Dist &d,float scale){
   cout<<"after scaling to the scale "<<endl;
   d.feet=d.feet*scale;
   d.inches=d.inches*scale;
}

int main(){
   float scale;
   Dist d1;
   d1.enter();
   d1.display();
```

```
    cout<<"enter the scale "<<endl;
    cin>>scale;
    d1.scale(d1,scale);
    d1.display();
    return 0;
}
```

Errors:

NO

Output:(Screenshot)



What You have learned from the Program:

How to take object by reference…..

P17

Create a Class Gate for students appearing in Gate (Graduate Aptitude test for Engineering) exam. There are three examination center Vadodara, Surat, and Ahmedabad where Gate exams are conducted. A class has data members: Registration number, Name of student, Examination center. Class also Contains static data member ECV_Cnt, ECS_Cnt and ECA_Cnt which counts the number of students in Vadodara, Surat and Ahmedabad exam center respectively. Class Contains two Member function getdata () which gets all information of students and counts total students in each exam center and pudata () which prints all information about the students. Class also contains one static member function getcount () which displays the total number of students in each examination center. Write a program for 5 students and display the total number of students in each examination center. Use static data member, static member function and Array of Objects.

Source Code:

```
#include<iostream>
#include<string.h>
using namespace std;
```

```
class Gate
{
   int Reg_no;    char name[50];    char center[10];    static int
ECV_Cnt,ECS_Cnt,ECA_Cnt; // IN  STATIC BT DEFULT
ECV_Cnt,ECS_Cnt,ECA_Cnt VALUES IS 0....    public:    void
getdata()
   {
      cout<<endl<<"Enter the registration number of the student:";
cin>>Reg_no;
      cout<<"Enter the name of the student:";
cin>>name;
      cout<<"Enter the name of the examination center for the
student"<<endl<<"(Surat/Vadodara/Ahmedabad):";
cin>>center;
      if(center[0]=='S' || center[0]=='s') //  if(Exam_center==Surat)
      {
         ECS_Cnt++;
      }
      else if(center[0]=='V' || center[0]=='v') //  if(Exam_center==Vadodara)
      {
         ECV_Cnt++;
      }
      else if(center[0]=='A' || center[0]=='a') //  if(Exam_center==Ahmedabad)
      {
         ECA_Cnt++;
      }
else
      {
         cout<<"INVALID EXAM CENTER";
      }
   }
   void putdata()
   {
      static int i=1;
      cout<<endl<<"Data of student-"<<i<<endl<<"Name:"<<name<<endl<<"Registration
Number:"<<Reg_no<<endl<<"Examination Center:"<<center<<endl;
i++;
   }
   static void getcount()
   {
      cout<<endl<<"Number of students having examination center as
Surat="<<ECS_Cnt<<endl<<"Number of students having examination center as
```

```
Vadodara="<<ECV_Cnt<<endl<<"Number of students having examination center as
Ahmedabad="<<ECA_Cnt<<endl;
   }
};
int Gate::ECS_Cnt=0; int
Gate::ECV_Cnt=0; int
Gate::ECA_Cnt=0;
int main()
{
   Gate student[5],N;
   for(int i=0;i<5;i++)
   {
      cout<<"-----Enter the Details for GATE student["<<i+1<<"]-----
"<<endl;       student[i].getdata();       student[i].putdata();       cout<<endl;
   }
   N.getcount();
return 0;
}
```

Errors:
NO

Output:(Screenshot):

```
-----Enter the Details for GATE student[4]-----

Enter the registration number of the student:4
Enter the name of the student:bhakti
Enter the name of the examination center for the student
(Surat/Vadodara/Ahmedabad):v

Data of student-4
Name:bhakti
Registration Number:4
Examination Center:v

-----Enter the Details for GATE student[5]-----

Enter the registration number of the student:5
Enter the name of the student:utsav
Enter the name of the examination center for the student
(Surat/Vadodara/Ahmedabad):a

Data of student-5
Name:utsav
Registration Number:5
Examination Center:a


Number of students having examination center as Surat=2
Number of students having examination center as Vadodara=2
Number of students having examination center as Ahmedabad=1
```

What You have learned from the Program:

How to use static member & array of object in class…..

P18

Define a class Fahrenheit with float temp as data member. Define another class Celsius with float temperature as data member. Both classes have member functions to input and print data. Write a non-member function that receives objects of both the classes and declare which one is higher than another according to their values. Also define main() to test the function. Define all member functions outsi class. (Formula for converting Celsius to Fahrenheit is F = (9C/5) + 32). Use the concept of friend fur

Source Code:

```
#include<iostream> using
namespace std;

class celcius;

class
farenhit{     float
temp;     public:
void enter();
void show();
   friend void max(farenhit,celcius);
};

void farenhit :: enter(){
   cout<<"enter the temperature in farenhit"<<endl;
cin>>temp;
}

void farenhit :: show(){
   cout<<"temperature in farenhit is "<<temp<<"F"<<endl;
}

class celcius{
   float temperature;

   public:
void enter();
void show();
   friend void max(farenhit,celcius);
};

void celcius :: enter(){
   cout<<"enter the temperature in celcius"<<endl;
cin>>temperature;
}

void celcius :: show(){
   cout<<"temperature in celcius is "<<temperature<<"C"<<endl;
}
```

```
void max(farenhit f,celcius c){     float
ftemp=(9*c.temperature)/5 + 32;
if(f.temp>ftemp)
```

```
    cout<<"farenhit object ha a higher temperature "<<f.temp<<"(F)"<<endl;
    else
    cout<<"celcius object has a higher temperature "<<ftemp<<"(F)"<<endl;
}

int main(){
    farenhit f;
    celcius c;

    f.enter();
    f.show();

    cout<<"\n";

    c.enter();
    c.show();
    cout<<"\n";

    cout<<"checking whether which temperature is bigger \n\n";
    max(f,c);
    return 0;
}
```

ERROR:
NO

Output:(Screenshot)

```
enter the temperature in farenhit
100
temperature in farenhit is 100F

enter the temperature in celcius
25
temperature in celcius is 25C

checking whether which temperature is bigger

farenhit object ha a higher temperature 100(F)
```

	What You have learned from the Program: How to compaie different values & print big values …..
P19	**Create a Class Date having data members: int dd, mm, yyyy. Class has one member function to input the dates and another member function which prints the dates. Write a main() function which takes two dates as input. Write a friend function swapdates() which takes two objects by reference of type Date and swaps both the dates. Use the concept of Friend function which takes objects by reference**

Source Code:

```
#include<iostream>
using namespace std;

class Date
{
   int dd , mm , yyyy;

   friend void swapdata(Date &d1,Date &d2);

public:
   void input()
   {
      cout<<"Enter the day=";
cin>>dd;
      cout<<"Enter the month=";
cin>>mm;
      cout<<"Enter the year=";
      cin>>yyyy;
   }
   void print()
   {
      cout<<"Date="<<dd<<"-"<<mm<<"-"<<yyyy<<endl;
   }
};
void swapdata(Date &d1,Date &d2)
{
   Date temp;
temp=d1;    d1=d2;
d2=temp;
}
main()
{
   Date d1,d2;
```

```
    d1.input();
    d2.input();

    cout<<'\n'<<"Before swaping"<<endl;
    d1.print();
    d2.print();

    swapdata(d1,d2);

    cout<<'\n'<<"After swaping"<<endl;
    d1.print();
    d2.print();
}
```

ERROR:
NO

Output:(Screenshot)

What You have learned from the Program:
How to swape two values with concept of freind function…..

P20

Create a class LAND having data members: length, width, area1. Write member functions to read and display the data of land. Also, calculates the area of the land. Create another class TILES having data members: l, w, area2. Write a member function to get the data of tile. Calculate the area of one tile. Class TILE has a member function named number_of_tiles() which is a friend of class LAND and takes the object of class LAND by reference which calculates the number of tiles which can be put over the land area. Write the main function to test all the functions. Use the concept of member function of one class can be a friend function of another class.

Source Code:

```
#include<iostream> using
namespace std;

class LAND;

class TILES{
   float l,w,area2;

   public:
void read(){
      cout<<"enter length of the tile"<<endl;
cin>>l;
      cout<<"enter width of the tile"<<endl;
cin>>w;
      area2=l*w;
   }

   void display(){       cout<<"\n\nthe length of the
tile is "<<l<<endl;       cout<<"the width of the tile
is "<<w<<endl;       cout<<"the area of the tile is
"<<area2<<endl;
   }

   void no_of_tiles(LAND &o);
};

class LAND{
   float length,width,area1;

   public:
   void read(){
      cout<<"enter length of land"<<endl;
cin>>length;
      cout<<"enter width of the land"<<endl;
cin>>width;
      area1=length*width;
   }

   void display(){
      cout<<"the length of the land is "<<length<<"\nthe width of the land is "<<width<<endl;
cout<<"the area of the land is "<<area1<<endl;
   }
```

```
   friend void TILES :: no_of_tiles(LAND &);
};

void TILES :: no_of_tiles(LAND &o){
   //number of tiles=area of land / area of one tile
   cout<<"\n\nthe number of tiles required to fill the land are "<<(o.area1)/area2<<endl;
}

int
main(){    LAND
o1;
   TILES o2;

   o1.read();
o1.display();

   o2.read();
o2.display();
o2.no_of_tiles(o1);
return 0;
}
```

ERROR:

NO

Output:(Screenshot)

```
enter length of land
100
enter width of the land
100
the length of the land is 100
the width of the land is 100
the area of the land is 10000
enter length of the tile
200
enter width of the tile
200


the length of the tile is 200
the width of the tile is 200
the area of the tile is 40000


the number of tiles required to fill the land are 0.25
```

	What You have learned from the Program: How to use member function of one class as friend function of another class….
P21	**Create a class Child having data members: name of the child and gender and a member function to get and print child data. Create another class Parent which is a friend class of child class. Class Parent have member function ReadChildData() which takes child's object by reference as input argument and Reads the childs data and DisplayChildData() which takes childs object as argument and displays childs data. Use the concepts of Friend Class.**

Source Code:

```
#include<iostream>
using namespace std;

class child;

class parent{
   string name_of_parent,gender;

   public:
   void realchilddata(child &);
void displaychilddata(child);
   void read(){
      cout<<"enter the name of parent"<<endl;
cin>>name_of_parent;
      cout<<"enter the gender of the parent"<<endl;
cin>>gender;
   }
};

class child{
   string name_of_child,gender;

   public:
friend parent;
};

void parent :: realchilddata(child &c){
   cout<<"\n\nenter the name of the child"<<endl;
cin>>c.name_of_child;
   cout<<"enter the gender of the child"<<endl;
cin>>c.gender;
}

void parent :: displaychilddata(child c){
   cout<<"\n\nthe data of children given by the parent is "<<name_of_parent<<endl;
cout<<"the name of the child is "<<c.name_of_child<<endl;
```

```
    cout<<"the gender of the child is "<<c.gender<<endl;
}

int main(){
    parent p;
    child c;

    p.read();
    p.realchilddata(c);
    p.displaychilddata(c);
    return 0;
}
```

ERROR:
NO

Output:(Screenshot)

What You have learned from the Program:
How to make & use friend function…..

P22

Check the following C++ code and find if there is any error in code, give justification for the error, correct the code and write the output.

1. Example of const member functions

```
#include<iostream>
 using namespace std;
class sample
 { int m, n;
public:
    void getdata();
    void putdata() const;
```

```
};
void sample::getdata()
{
cout<< "Enter m & n";
cin>>m>>n;
}
void sample::putdata() const {
 m=12;
n=34;
cout<< " m = "<<m<<"n= "<<n;
}
int main()
{
 sample s1;
 s1.getdata();
 s1.putdata();
 return 0;
}
```

Error:

```
File            L...  Message
                      === Build: Debug in p22....1 (compiler: GNU GCC Compiler) ===
D:\C++\p2...          In member function 'void sample::putdata() const':
D:\C++\p2... 15       error: assignment of member 'sample::m' in read-only object
D:\C++\p2... 16       error: assignment of member 'sample::n' in read-only object
                      === Build failed: 2 error(s), 0 warning(s) (0 minute(s), 0 ...
```

Justification for the error:
When The function try to alter the data values this error come out…

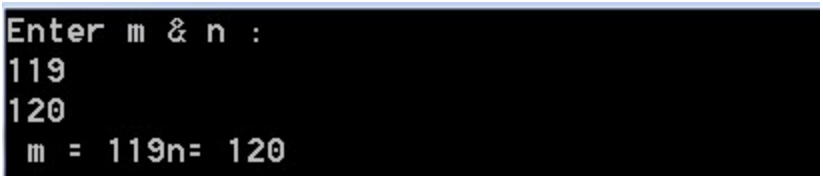
Correct Source Code:

```
#include<iostream>
 using namespace std;
class sample
 { int m, n;
public:
    void getdata();
    void putdata() const;
 };
void sample::getdata()
 {
cout<< "Enter m & n :"<<endl;
cin>>m>>n;
}
```

```
void sample::putdata() const {
// m=12;
//n=34;
cout<< " m = "<<m<<"n= "<<n;
}
int main()
 {
 sample s1;
 s1.getdata();
 s1.putdata();
 return 0;
}
```

ERROR:
NO

Output:(Screenshot)



2. (a) Pointer to data members

```
#include<iostream>
using namespace std;
class student
{
public:
int roll_no;
 };
 int main()
{
// declaring pointer to data member
 int student :: *p1 = &student::roll_no;
 student s;
 student *optr = &s;
s->*p1 = 42;
cout<<"Roll no is "<< s->*p1<<endl;
optr.*p1 = 45;
cout<<"Roll no is"<<optr.*p1<<endl;
return 0;
}
```

Error:

```
File            L...  Message
                      === Build: Debug in p22......2 (compiler: GNU GCC Compiler)...
D:\C++\p2...          In function 'int main()':
D:\C++\p2... 14       error: no match for 'operator->*' (operand types are 'stude...
D:\C++\p2... 15       error: no match for 'operator->*' (operand types are 'stude...
D:\C++\p2... 16       error: cannot apply member pointer 'p1' to 'optr', which is...
D:\C++\p2... 17       error: cannot apply member pointer 'p1' to 'optr', which is...
                      === Build failed: 4 error(s), 0 warning(s) (0 minute(s), 0 ...
```

Justification for the error:

1) ->* used to access a member when we use pointer to object & member.
2) .>*used to when object is uded with the member pointer..

Correct Source Code:

```
#include<iostream>
 using namespace std;
 class student
{
public:
int roll_no;
 };
 int main()
{
// declaring pointer to data member
 int student :: *p1 = &student::roll_no;
 student s;
 student *optr = &s;
//s->*p1 = 42;
s.*p1 = 42;
cout<<"Roll no is "<< s.*p1<<endl;
//optr.*p1 = 45;
optr->*p1 = 45;
cout<<"Roll no is"<<optr->*p1<<endl;
return 0;
}
```

ERROR:

NO

Output:(Screenshot)

What You have learned from the Program:

How to write easy code for pointer to data member…

2. (b)Pointer to member functions

```
#include<iostream>
 class employee
 {
public:
 void hello()
{
cout<<"Hi hello"<<endl;
}
};
int main()
{
// declaring pointer to member function hello
void (employee ::*fp)() = &employee::hello;
 employee e;
employee *optr = &e;
 (e->*fp)();
 (optr.*fp)();
 return 0;
 }
```

Error:

File	L...	Message
D:\C++\20...	7	error: 'endl' was not declared in this scope
D:\C++\20...	7	note: suggested alternative:
C:\Progra...	564	note: 'std::endl'
D:\C++\20...		In function 'int main()':
D:\C++\20...	16	error: no match for 'operator->*' (operand types are 'emplo...
D:\C++\20...	17	error: cannot apply member pointer 'fp' to 'optr', which is...
		=== Build failed: 4 error(s), 0 warning(s) (0 minute(s), 0 ...

Justification for the error:

3) ->* used to access a member when we use pointer to object & member.
4) .>*used to when object is uded with the member pointer..

Correct Source Code:

```
#include<iostream>
using namespace std;
 class employee
 {
public:
 void hello()
{
cout<<"Hi hello"<<endl;
}
};
int main()
{
// declaring pointer to member function hello
void (employee ::*fp)() = &employee::hello;
 employee e;
employee *optr = &e;
//(e->*fp)();
(e.*fp)();
//(optr.*fp)();
(optr->*fp)();
 return 0;
 }
```

ERROR:
NO

Output:(Screenshot)

```
Hi hello
Hi hello
```

What You have learned from the Program:

How to write code for pointer to data member…

3. Example of Local Classes

Source code:

```
#include<iostream>
 using namespace std;
 void testlocalclass()
 {
 class Test {
static int cnt;
 public:
void set()
{cout<<"Enter Count: ";
 cin>>cnt;
 }
void get();
};
void Test:: get()
{ cout<<"Count: = " <<cnt;}
Test t;
 t.set();
 t.get();
 }
 int main()
{
testlocalclass();
 return 0;
}
```

Error:

```
File            L... Message
                     === Build: Debug in p22....3 (compiler: GNU GCC Compiler) ===
D:\C++\20...         In function 'void testlocalclass()':
D:\C++\20... 6       error: local class 'class testlocalclass()::Test' shall not...
D:\C++\20... 14      error: qualified-id in declaration before '(' token
                     === Build failed: 2 error(s), 0 warning(s) (0 minute(s), 0 ...
```

Justification for the error:

Local classes can not have static data members…

Correct Source Code:

```
#include<iostream>
 using namespace std;
 void testlocalclass()
 {
 class Test {
//static int cnt;
int cnt;
 public:
void set()
{cout<<"Enter Count: ";
 cin>>cnt;
 }
void get()
{
   cout<<"count"<<endl;
}
};
/*void Test:: get()
{ cout<<"Count: = " <<cnt;}
*/
Test t;
 t.set();
 t.get();
 }

 int main()
{
testlocalclass();
 return 0;
}
```

ERROR:
NO

Output:(Screenshot)

What You have learned from the Program:
How to write code for local class…

P23 Write a C++ program having class time with data members: hr, min and sec. Define following member functions. 1) getdata() to enter hour, minute and second values 2) putdata() to print the time in the format 11:59:59 3) default constructor 4) parameterized constructor 5) copy constructor 6) Destructor. Use 52 as default value for sec in parameterized constructor. Use the concepts of default constructor, parameterized constructor, Copy constructor, constructor with default arguments and destructor.

Source Code:

```
#include
<iostream> using
namespace std; class
Time
{ public:
   int hr, min, sec, x;
   void getdata()
   {
      cout << "\nEnter the hours :";
cin >> hr;
      cout << "Enter the minutes :";
cin >> min;
      cout << "Enter the seconds :";
cin >> sec;
   }
   void putdata()
   {
      if (sec > 60 || sec == 60)
      {
         x = sec / 60;
min = min + x;
         sec = sec - (x * 60);
      }
      if (min > 60 || min == 60)
      {
         x = min / 60;
hr = hr + x;
         min = min - (x * 60);
      }
      cout << endl
          << "Time is " << hr << ":" << min << ":" << sec;
   }
   Time()
   {
      cout << "\n*****Default constructor*****";
hr = min = sec = 0;
   }
   Time(int h, int m, int s)
```

```
    {
        cout << "\n*****Parameterised constructor*****";
        hr = h;
        min = m;
        sec = s;
    }
    Time(Time &t)
    {
        cout << "\n*****Copy constructor*****";
        hr = t.hr;
        min = t.min;
        sec = t.sec;
    }
    ~Time()
    {
        cout << "\n*****Destructor*****";
    }
};
int main()
{
    Time t1;
    t1.getdata();
    t1.putdata();
    Time t2(12, 59, 52);
    t2.putdata();
    Time t3(t2);
    t3.putdata();
    return 0;
}
```

Error:
No

Output:(Screenshot

```
*****Default constructor*****
Enter the hours :12
Enter the minutes :50
Enter the seconds :50

Time is 12:50:50
*****Parameterised constructor*****
Time is 12:59:52
*****Copy constructor*****
Time is 12:59:52
*****Destructor*****
*****Destructor*****
*****Destructor*****
```

What You have learned from the Program:
How to use default constructor , parameterized constructor , copy constructor ,& constructor with default arguments & Destructor.

P24

Create a class Number having int num as member. The class has input and output functions. Overload unary operator (++) such that it supports N1=N2++ and N3=++N1 and Overload unary (-) such that it supports N3 = - N3. Also define default, parameterized and copy constructor for the class. Also explain use of nameless object in operator overloading. Use the concept of Overloading Unary Operators

Source Code :

```
#include <iostream>
using namespace std;
class number
{
public:
   int num;
   void input();
   void output();
   number() //default
   {
      num = 0;
   }
   number(int x) //parametrized
   {
      num = x;
   }
   number(number &N, number &NU) //copy constructor
   {
      num = N.num;
      num = NU.num;
   }
   number operator++();    //pre increment declaration
   number operator++(int); //post increment declaration
   number operator-();     //negative sign
};
void number::input() //input function calling
{
   cout << "Enter the number : ";
   cin >> num;
}
void number::output() //output function calling
{
   cout << num << endl;
}
```

```
number number::operator++()
{
   return number(++num);
}
number number::operator++(int)
{
   return number(num++);
}
number number::operator-()
{
   int temp;
temp = -num;
   return temp;
}
int main()
{
   number N1, N2, N3;
   cout << "Parameterized
constructor\n";    number n(7);
n.output();    N2.input();
   N1 = N2++;
   N3 = ++N1;
N3 = -N3;
   cout << "N1 is : ";
N1.output();    cout
<< "N2 is : ";
N2.output();    cout
<< "N3 is : ";
N3.output();
return 0;
}
```

Error:
No

Output:(Screenshot)

```
Parameterized constructor
7
Enter the number : 55
N1 is : 56
N2 is : 56
N3 is : -56
```

What You have learned from the Program:
How to over load unary operators.

P25

Create a class complex having data members int real , img and member function to print data. Overload Unary operator (-) using friend function such that it supports – C1 where C1 is the object of class complex. Also define default, parameterized and copy constructor for the class. Use the concept of Overloading Unary Operators with friend function.

Source Code :

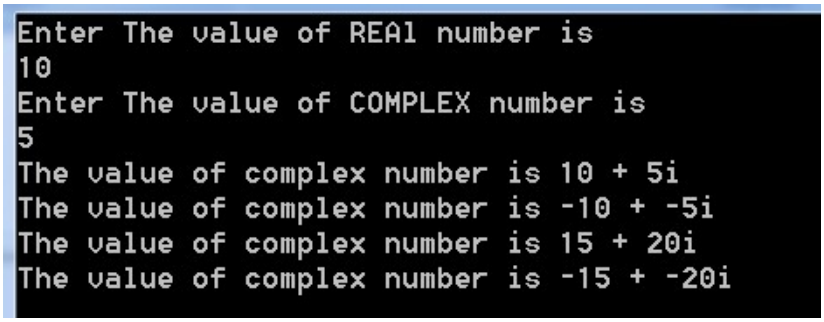
```
#include <iostream>
using namespace std;
class complex
{
   int real, img;

public:
   void showdata()
   {
      cout << "The value of complex number is " << real << " + " << img << "i" << endl;
   }
   complex()
   {
      cout << "Enter The value of REAl number is " <<endl;
      cin>>real;

      cout << "Enter The value of COMPLEX number is " <<endl;
      cin>>img;
   }
   complex(int a, int b)
   {
      real = a;
      img = b;
   }
   void operator-()
   {
      real = -real;
      img = -img;
   }
};
int main()
{
   complex c1, c2(15, 20);
   c1.showdata();
   -c1;
   c1.showdata();
   c2.showdata();
```

```
  -c2;
  c2.showdata();
  return 0;
}
```

Error :

No

Output:(Screenshot)



What You have learned from the Program:

How to over load unary operators with use of friend function.

P26

Create a class String having character array. Class includes constructor and required member functions to get and display the object. Overload the operators +(s3=s1+s2), ==(s1<s2),(s1+=s2) for the class. Use the concept of Overloading Binary Operators.

Source code:

```
#include <iostream>
#include <string.h>
#include <iomanip>
using namespace std;
class Str
{
   char st[25];

public:
   Str()
   {
      st[0] = '\0';
   }
   void getdata()
   {
```

```
      cout << "Enter 1st String: ";
cin >> st;
   }
   void getdata1()
   {
      cout << "Enter 2nd String: ";
cin >> st;
   }
   void putdata()
   {
      cout << st << endl;
   }
   Str operator+(Str s2)
   {
      Str t;
strcat(t.st, st);
strcat(t.st, s2.st);
return t;
   }
   Str operator==(Str s2)

{       int
a;
      a = strcmp(st, s2.st);
      if (a == 0)
      {
         cout << "Both string are equal" << endl;
      }
else
      {
         cout << "Not Equal" << endl;
      }
   }
   Str operator<(Str s2)

{       int
a;
      a = strcmp(st, s2.st);
      if (a > 0)
      {
         cout << "Str1 is greater than Str2" << endl;
      }
else
      {
         cout << "Str1 is less than Str2" << endl;
```

```
    }
}
```

What You have learned from the Program:
How to overload binary operators..

P27

Create a class Measure having members: meter and cm. The class has get( ) and put( ) functions. Overload operator + and – such that they support M1=M2+15 and M3=M1 – 4.5. Also overload + and – such that they support M1=5.0+M2 and M3=2.0 – M4. Write a main( ) to test the class. Use the concept of Overloading Binary Operators with friend function.

Source Code:

```
#include <iostream>
#include <string.h>
#include <iomanip>
using namespace std;
class Measure
{
   float m, cm;

public:
   void getdata()
   {
      cout << "Enter in metre: ";
      cin >> m;
      cout << "Enter in cm: ";
      cin >> cm;
   }
   void putdata()
   {
      cout << "In metre: " << m << endl;
      cout << "In cm: " << cm << endl;
   }
   Measure operator+(int a)
   {
      Measure m1;
      m1.m = m + a;
      m1.cm = cm + a;
      return m1;
   }
   Measure operator-(float b)
   {
      Measure m1;
      m1.m = m - b;
      m1.cm = cm - b;
      return m1;
   }
   Measure friend operator+(float i, Measure m2);
```

Output:(Screenshot)

```
Enter in metre: 100
Enter in cm: 150

        After Addition

In metre: 115
In cm: 165

        After Sub new values:

In metre: 110.5
In cm: 160.5

        After adding using friend func

In metre: 105
In cm: 155

        After Subtracting using friend function

In metre: -98
In cm: -148
```

What You have learned from the Program:

How to overload binary operators using friend function…

P28

Create a class Celsius with float. Define appropriate member functions such that it support the statements: C1=30.5F; float temperature; temperature=C2; Use the concept of Type conversion from basic type to class type and class type to basic type.

Source Code:

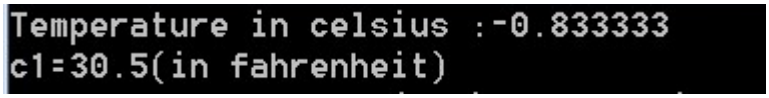
```
#include <iostream>
using namespace std;

class celsius
{
   float temp;

public:
   void operator=(float i)
   {
      temp = ((float)5 / 9) * (i - 32);
   }

   void display()
   {
      cout << "Temperature in celsius :" << temp << "\nc1=30.5(in fahrenheit)";
   }
};
```

```
int main()
{
   celsius c1;

   c1 = 30.5;

   c1.display();

   return 0;
}
```

Error :

No

Output:(Screenshot)



What You have learned from the Program:

How to do type conversion from basic to class & class to basic…

P29

Create classes Celsius and Fahrenheit with float. Define appropriate member functions such that they support the statements in main( ): Celsius C1, C2=5.0; Fahrenheit F1, F2; F1=C2; C1=F2;Use the concepts of Type conversion from class type to class type.Write this Program in two ways. Define appropriatemember function in class Celsius. Define appropriate member function in class Fahrenheit

Source Code:

```
#include <iostream>
using namespace std;
class Fahrenheit;
class Celsius
{
   float celsius;

public:
   Celsius(){};
   Celsius(float c) : celsius(c){};
```

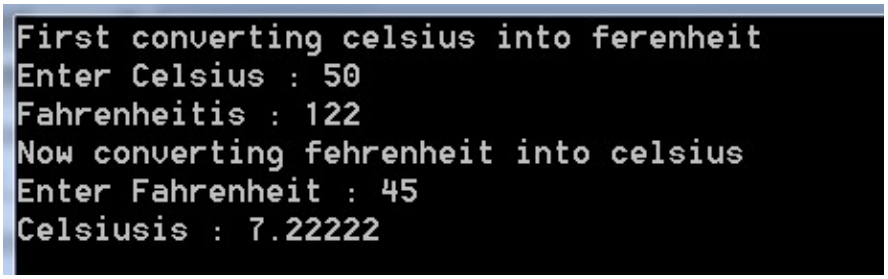
```
   float getvalue()
   {
      return celsius;
   }
   Celsius(Fahrenheit);
   void getc()
   {
      cout << "Enter Celsius : ";
      cin >> celsius;
   }
   void putc()
   {
      cout << "Celsiusis : " << celsius;
   }
};
class Fahrenheit
{
   float fahrenheit;

public:
   Fahrenheit(){

   };
   Fahrenheit(float f) : fahrenheit(f){

                  };
   float getvalue2()
   {
      return fahrenheit;
   }
   Fahrenheit(Celsius c)
   {
      fahrenheit = (9 * c.getvalue() / 5) + 32;
   }
   void getf()
   {
      cout << "Enter Fahrenheit : ";
cin >> fahrenheit;
   }
   void putf()
   {
      cout << "Fahrenheitis : " << fahrenheit;
   }
};
Celsius ::Celsius(Fahrenheit f)
{
```

```
    celsius = (f.getvalue2() - 32) * (5) / 9;
}
int main()
{
    Celsius c1, c2;
    Fahrenheit f1, f2;
    cout << "First converting celsius into ferenheit" << endl;
    c2.getc();
    f1 = c2;
    f1.putf();
    cout << endl;
    cout << "Now converting fehrenheit into celsius" << endl;
    f2.getf();
    c1 = f2;
    c1.putc();
    cout << endl;

    return 0;
}
```

Error :

No

Output:(Screenshot)



What You have learned from the Program:

How to do conversion one cla to another class…

P30 **Define a Base Class Vegetable having data member Color and member function getdata() which takes color as an input and putdata() which print the color as an output.Vegetable Class has one subclass named Tomato having data members weight and size and member function gtdata() which takes weight and size as an input and ptdata() which prints weight and size as output. Write a C++ Program which inherit the data of Vegetable class in Tomato class using Single Inheritance.**

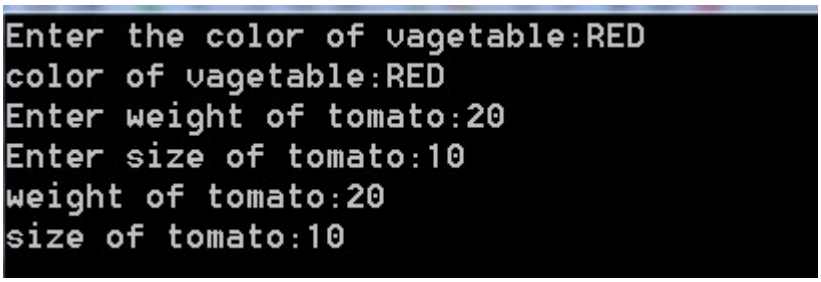
Source Code:

```
#include<iostream>
using namespace std;

class vegetable
{    char
color[20];
public:    void
getdata()
   {
cout<<"Enter the color of vagetable:";
cin>>color;
   }
   void putdata()
   {
cout<<"color of vagetable:"<<color<<endl;
   }
};
class tomato:public vegetable
{    float
weight,size;
public:
   void gtdata()
   {
cout<<"Enter weight of tomato:";
cin>>weight;
cout<<"Enter size of tomato:";
cin>>size;

   }
   void ptdata()
   {
cout<<"weight of tomato:"<<weight<<endl<<"size of tomato:"<<size<<endl;
   }
};
int main()
{
```

```
    tomato t; t.getdata();
t.putdata();
t.gtdata();
t.ptdata();
}
```

Error :
No

Output:(Screenshot)



What You have learned from the Program:
I learnt the conceot of Single Inheritance.

P31

Write a program to create a class Medicine which stores type of medicine, name of company, date of manufacturing. Class Tablet is inherited from Medicine. Tablet class has name of tablet, quantity per pack, price of one tablet as members. Class Syrup is also inherited from M edicine and it has quantity per bottle, dosage unit as members. Both the classes contain necessary member functions for input and output data. Write a main( ) that enter data for tablet and syrup, also display the data. Use the concepts of Hierarchical Inheritance.

Source Code:

```
#include<iostream>
using namespace std;

class Medicine
{
   char type_medicine[50],name_company[20],date_manufacturing[30];

public:
   void getdata()
   {
cout<<"Enter the type if medicine(tablet/syrup):";
cin>>type_medicine;
cout<<endl<<"Enter the name of manufacturing company:";
cin>>name_company;
cout<<endl<<"Enter the date of manufacturing(e.g. 21-jan-2015):";
cin>>date_manufacturing;
```

```
    }
    void putdata()
    {
cout<<"your medicine is "<<type_medicine; cout<<"
manufactured by "<<name_company; cout<<" on
"<<date_manufacturing;
    }
};

class Tablet:public Medicine
{
    char name_tablet[30]; int
quantity_per_pack,price_of_one_tablet;
public:
    void gtdata()
    {
cout<<endl<<"Enter name of tablet:";
cin>>name_tablet;
cout<<endl<<"Enter number of tablets per pack:"; cin>>quantity_per_pack;
cout<<endl<<"Enter the price of each tablet:";
cin>>price_of_one_tablet;
    }
    void ptdata()
    {
cout<<endl<<name_tablet<<" has "<<quantity_per_pack<<" tablets per pack having
"<<price_of_one_tablet<<" Rs. price per tablet.";
    }
};

class Syrup:public Medicine
{
int quantity_per_bottle,dosage_unit;
public:     void gsdata()
    {
cout<<endl<<"Enter the quantity of syrup per bottle in ml:";
cin>>quantity_per_bottle;
cout<<endl<<"Enter the dosage unit:";
```

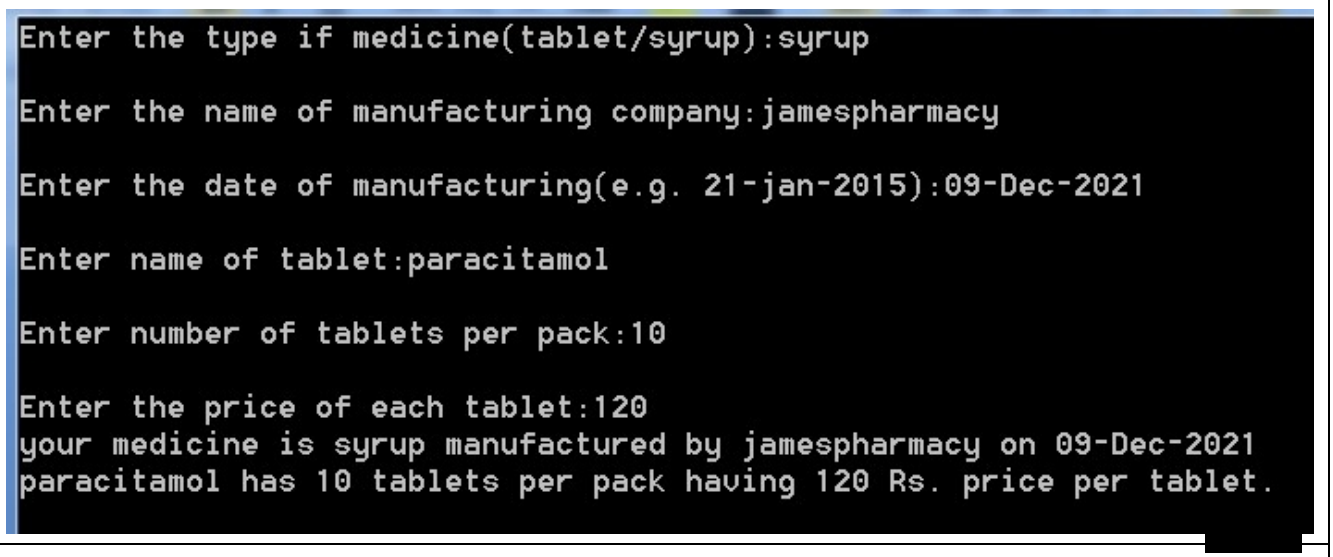
```
cin>>dosage_unit;
   }
   void psdata()
   {
cout<<endl<<" one bottle of syrup is of "<<quantity_per_bottle<<" ml and having
"<<dosage_unit<<" dosage unit ";
```

```
    }
};

int main()
{
    Tablet t;
    Syrup s;

t.getdata();
t.gtdata();
t.putdata();
t.ptdata();
cout<<endl<<endl;

s.getdata();
s.gsdata();
s.putdata();
s.psdata();

    return 0;
}
```

Error :

No

Output:(Screenshot)



	What You have learned from the Program: I learnt the concepts of Hierarchical Inheritance.
P32	**Create a Class alpha having data member: int x and one argument constructor which initializes the value of x. It also has member function which displays the value of x. Create another class beta which contains data member: float y and one argument constructor which initializes the value of y. It also has member function which displays the value of y. Create a Class Gamma which publicly inherits from class alpha and class beta and has two data members: int m, n and a constructor which passes argument to the base class constructor as well as initializes its own data members. Class Gamma also has member function to print the values of m and n. Write main function which creates object of class Gamma which passes values of base class constructor as well as derived class constructor. Use the concept of Multiple Inheritance and Constructor in Derived Class.**

Source Code:

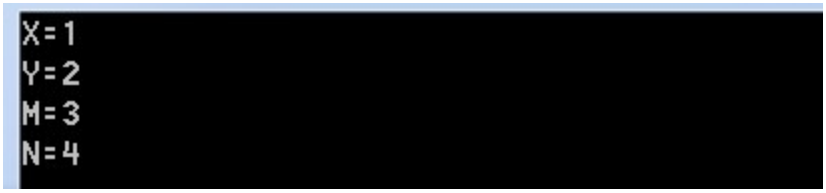
```
#include<iostream>
using namespace std;

class alpha
{ int x;
public:
   alpha(int a)
   {
      x = a;
   }
   void display()
   {
cout<< "X=" << x <<endl;
   }
};

class beta
{ int y;
public:
   beta(int b)
   {
      y = b;
   }
   void display()
   {
cout<< "Y=" << y <<endl;
```

```
    }
};

class Gamma : public alpha, public beta
{
int m, n;
public:
    Gamma(int a, int b, int c, int d):alpha(a), beta(b)
    {
        m = c;
        n = d;
    }
    void display()
    {
cout<< "M=" << m <<endl<< "N=" << n <<endl;
    }
};

int main()
{
    Gamma obj(1,2,3,4);
obj.alpha::display();
obj.beta::display();
obj.display();
}
```

Error :

No

Output:(Screenshot)



What You have learned from the Program:

I learnt the concept of Multiple Inheritance and Constructor in Derived Class.

P33 **Define a class Hospital having rollno and name as data members and member function to get and print data. Derive a class Ward from class Hospital having data members: ward number and member function to get and print data. Derive another class Room from Hospital having data member bed number and nature of illness and member function to get and print data. Derive class Patient from Class Ward and Class Room. In main () declare 5 object of Class Patient and get and display all the information. Use the concept of Virtual Base Class and Hybrid Inheritance.**

Source Code:

```
#include<iostream>
using namespace std;

class Hospital
{
introll_no;
   char name[10];
public :
   void get()
   {
cout<<"Enter roll no.:";
cin>>roll_no ;
cout<<"Enter name:";
cin>>name;
   }
   void print()
   {
cout<<"roll no. is:"<<roll_no<<" name is:"<< name;
   }
};
class Ward : public virtual Hospital
{
protected:
int number;
public:
   void get()
   {
cout<<"Enter number:";
cin>> number;
   }
   void print()
   {
cout<<" number is:"<< number;
```

```
    }
};
class Room : virtual public Hospital
{
protected:
intbed_number;    char
nature_of_illness[50];
public:
   void get()
   {
cout<<"Enter bed no.:"; cin>>bed_number;
cout<<"Enter nature of illness:"; cin>>nature_of_illness;
   }
   void print()
   {
cout<< " bed number is:"<<bed_number<<" nature of illness is:"<<nature_of_illness<<endl;
   }
};
class Patient : public Ward,public Room
{
};
int main()
{ inti;
   Patient p[5];
   for(i=0;i<5;i++)
   {
      p[i].Hospital::get();
p[i].Ward::get();
      p[i].Room::get();
   }
   for(i=0;i<5;i++)
   {
      p[i].Hospital::print();
p[i].Ward::print();
      p[i].Room::print();
   }
   return 0;
}
```

Error :

No

Output:(Screenshot)

```
Enter roll no.:6544
Enter name:boif
Enter number:56435
Enter bed no.:56465
Enter nature of illness:fever
Enter roll no.:1534
Enter name:hiud
Enter number:1354
Enter bed no.:131
Enter nature of illness:cold
Enter roll no.:245
Enter name:sddfc
Enter number:56465
Enter bed no.:353
Enter nature of illness:cough
Enter roll no.:35453
Enter name:iduff
Enter number:452
Enter bed no.:575
Enter nature of illness:dengue
Enter roll no.:14351
Enter name:sdfs
Enter number:13513
Enter bed no.:54343
Enter nature of illness:cold
roll no. is:6544 name is:boif number is:56435 bed number is:56465 nature of illn
ess is:fever
roll no. is:1534 name is:hiud number is:1354 bed number is:131 nature of illness
 is:cold
roll no. is:245 name is:sddfc number is:56465 bed number is:353 nature of illnes
s is:cough
roll no. is:35453 name is:iduff number is:452 bed number is:575 nature of illnes
s is:dengue
roll no. is:14351 name is:sdfs number is:13513 bed number is:54343 nature of ill
ness is:cold
```

What You have learned from the Program:

I learnt the concept of Virtual Base Class and Hybrid Inheritance.

P34 **Create a class shape having data member shape_name and member function to get and print shape_name. Derive a Class Circle which is inherited publicly from class shape and having data members radius of a circle and member function to get and print radius of a circle. Derive a Class Area which is inherited publicly from Class Circle and having data members area_of_circle and member function display () which displays area of a circle. Use object of class Area in main () function and get and display all the information. Use the concepts of Multilevel Inheritance.**

Source Code:

```
#include<iostream>
using namespace std;

class shape
{
   char shap_name[20];
public:
   void get()
   {
cout<<"Enter the shap:";
cin>>shap_name;
   }
    void print()
   {
cout<<"shap is:"<<shap_name<<endl;
   }
};
class circle:public shape
{
   protected:
   float radius;
public:
   void get()
   {
cout<<"Enter the radius of circle:";
cin>>radius;
   }
    void print()
   {
cout<<"radius is:"<<radius<<endl;
   }
};
class area:public circle
{
   float area_of_circle;
```

```
public:
   void disp()
   {
area_of_circle=3.14*radius*radius;
cout<<"area of circle is:"<<area_of_circle;
   }

};
int main()
{
   area a;
a.shape::get();
a.shape::print();
a.circle::get();
a.circle::print();
a.disp();

}
```

Error :

No

Output:(Screenshot)

```
Enter the shap:circle
shap is:circle
Enter the radius of circle:2
radius is:2
area of circle is:12.56
```

What You have learned from the Program:

I learnt the concept of Multiple Inheritance and Constructor in Derived Class.

P35

Create one application n a group of 3 person which implement all type of inheritance

Source Code:

```
#include <iostream>
#include <string>
using namespace std;
class shop
{
protected:
```

```
 int price, quantity;
string s; public:
void entry();
};
void shop::entry()
{
 cout << "ENTER ITEM NAME : ";
cin >> s;
 cout << "ENTER PRICE OF ITEM : ";
cin >> price;
 cout << "ENTER QUANTITY : ";
cin >> quantity;
}
class BILL : public virtual shop
{ public:
 void bill()
 {
 cout << "THE BILL AMOUNT OF " << s << " IS :" << price * quantity << endl;
 } };
class GST : public virtual BILL
{ public:
 void gst()
 {
 cout << "THE GST (5%) SO YOUR TOTAL PAYBLE AMOUNT IS : " << ((price
* quantity) * 0.05) + price * quantity << endl;
 } };
class payment : public virtual shop,
public virtual BILL,
public GST
{ protected:
 int RUPEES;
public:
 void pay()
 {
 cout << "ENTER THE AMOUNT YOU HAVE : ";
cin >> RUPEES;
 cout << "TO BUY THIS ITEM YOU WILL NEED " << (((price * quantity) * 0.05) +
price * quantity) - RUPEES << " rs more " << endl;
 } };
int main()
{
 payment p1;
```

```
p1.entry();
p1.bill();
p1.gst();
p1.pay();
return 0;
}
}
```

Error :

No

Output:(Screenshot)

```
ENTER ITEM NAME : TOY
ENTER PRICE OF ITEM : 150
ENTER QUANTITY : 1
THE BILL AMOUNT OF TOY IS :150
THE GST (5%) SO YOUR TOTAL PAYBLE AMOUNT IS : 157.5
ENTER THE AMOUNT YOU HAVE : 100
TO BUY THIS ITEM YOU WILL NEED 57.5 rs more
```

What You have learned from the Program:

I learnt Single , Multiple , Multilevel , Hybrid Inheritance.

P36 **What is the output of the following code:**

(a) Pointer to Objects

```
#include<iostream>
using namespace std;
class product
{
int code;
float price;
public:
void getdata(int a, float b)
{
code=a;
price=b;
}
void show()
```

```
{
cout<<"Code: "<<code<<endl; cout<<"Price:
"<<price<<endl;
}
};
 int main()
{
product * p = new product;  product
*d = p;
int x,i; float
y;
cout<<"Input code and price for product: ";
cin>>x>>y;  p->getdata(x,y);  d->show();
 }
```

(b) this pointer

```
 #include<iostream> using
namespace std;
class student
{
   int roll_no;    float
age; public:
student(int r, float a)
   {
   roll_no = r;
   age = a;
   }
   student & greater (student & x)
   {
   if(x.age>=age)
return x;    else
   return *this;
   }
   void display()
   {
   cout<<"Roll          No          "<<roll_no<<endl;
cout<<"Age "<<age<<endl;
   }
};
int main()
{
```

```
   student s1 (23,18),s2 (30,20),s3 (45,16);
student s = s1.greater(s3);    cout<<"Elder
Person is :"<<endl;    s.display();
}
```

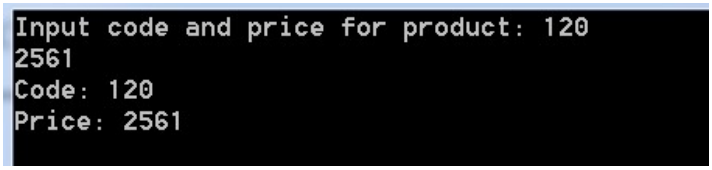
(c) Pointers to Derived Objects

```
#include<iostream> using
namespace std;
class BC
{
public:
   int b;
   void show()
   {
   cout<<"b = "<<b<<endl;
   }
};
class DC : public BC
{    public:
int d;
   void show()
   {
   cout<<"b = "<<b<<endl;
cout<<"d = "<<d<<endl;
   }
};
int main()
{
   BC *bptr;    BC
base;    bptr =
&base;    bptr->b
= 100;
   cout<<"bptr poins to base objects"<<endl;
bptr->show();    DC derived;    bptr =
&derived;    bptr->b = 200;
   /*bptr->b = 300;*/ // wont work
   cout<<"bptr now points to derived object"<<endl;
bptr->show();    DC *dptr;    dptr=&derived;    dptr-
>d=300;
```

```
    cout<<"Dptr is derived type pointer"<<endl;
    dptr->show();
    return 0;
}
```

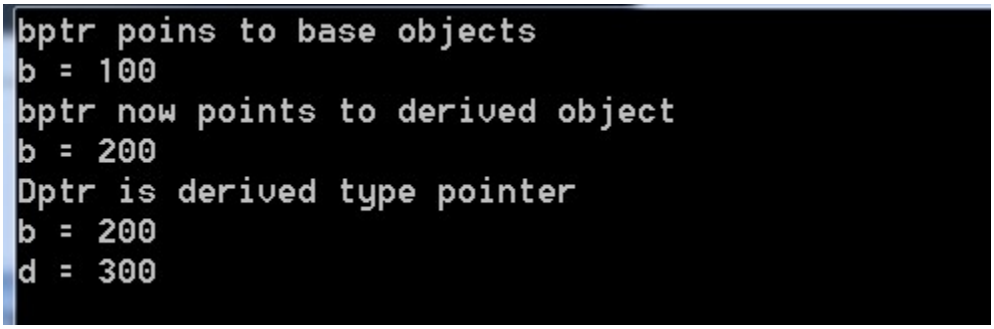
Output:(Screenshot)

(a) Pointer to Objects



(b) this pointer

(c) Pointers to Derived Objects



P37	**Create a class Media that stores the title (a string) and price (float). Class Mediahas two argument constructor which initializes data members of class Media. Also declare a virtual function display () in Class Media. From the class Media derive two classes: Class book, which contains data member page count (int): and Class tape, which contains data member playing time in minutes (float). Both Class book and Class tape should have a constructor which initializes base class constructor as well as its own data members and display ( ) function which displays book details and tape details respectively. Write a main ( ) to test book and tape classes by creating instances of them, asking the user to fill data and displaying them. Use the concept of Virtual function and Constructor in Derived Class.**

Source Code:

```
#include<iostream>
using namespace std;

class media
{    string title;    float
price; public:
media(string a,float b)
   {
      title=a;
      price=b;
   }
   virtual void display()
   {
      cout<<"title = "<<title<<endl;
      cout<<"price = "<<price<<endl;
   }

};
class book : public media
{    int
page_count;
public:
   book(int x,string y,float z):media(y,z)
   {
      page_count=x;
   }
   void display()
   {
      cout<<"page count = "<<page_count<<endl;
   }
};
class tape:public media
```

```
{    float playing_time; public:
tape(float x,string y,float z):media(y,z)
   {
      playing_time=x;
   }
   void display()
   {
      cout<<"playing time = "<<playing_time<<endl;
   }
};
int main()
{    int a;
float b,d;
   string c;
   cout<<"Enter title of book : ";
cin>>c;
   cout<<"Enter price of the book : ";
cin>>b;
   cout<<"Enter page count : ";
cin>>a;
   cout<<"Enter playing time : ";
cin>>d;

   media m(c,b),*bptr=&m;

   book        b1(a,c,b);
tape t1(d,c,b);
    cout<<"Data of book"<<endl;
   bptr->display();
bptr = &b1;
   bptr->display();

   media *tptr=&m;
   cout<<"Data of tape"<<endl;

   tptr->display();
tptr=&t1;    tptr-
>display();    return 0;
}
```

Error :
NO

OutPut:(Screenshot)

```
Enter title of book : F.R.I.E.N.D.S
Enter price of the book : 100
Enter page count : 250
Enter playing time : 30
Data of book
title = F.R.I.E.N.D.S
price = 100
page count = 250
Data of tape
title = F.R.I.E.N.D.S
price = 100
playing time = 30
```

What You have learned from the Program:
I learnt concept of Virtual function and Constructor in Derived Class.

P38

Create an Abstract class vehicle having average as data and pure virtual function getdata() and putdata(). Derive class car and truck from class vehicle having data members: fuel type (petrol, diesel, CNG) and no of wheels respectively. Write a main ( ) that enters the data of two cars and a truck and display the details of them. Use the concept of Abstract Base class and Pure Virtual functions.

Source Code:

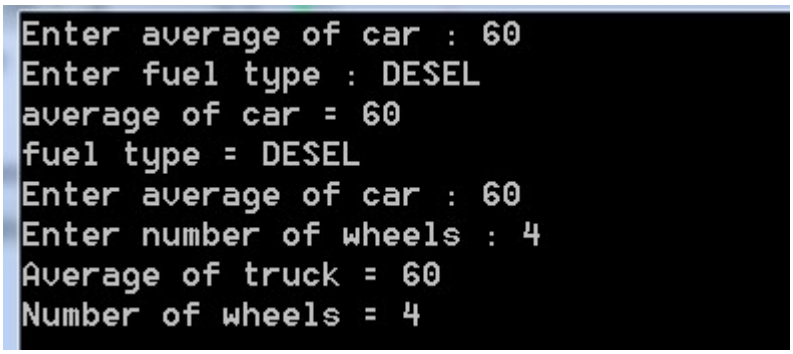
```
#include<iostream>
using namespace std;

class vehicle
{
protected:
   float avg;
public:
   virtual void getdata()=0;
   virtual void putdata()=0;
};
class car : public vehicle
{
```

```
   string
fuel_type; public:
void getdata()
   {
      cout<<"Enter average of car : ";
cin>>avg;
      cout<<"Enter fuel type : ";
cin>>fuel_type;
   }
   void putdata()
   {
      cout<<"average of car = "<<avg<<endl;
cout<<"fuel type = "<<fuel_type<<endl;
   } };
class truck : public vehicle
{    int
no_of_wheels;
public:    void
getdata()
   {
      cout<<"Enter average of car : ";
cin>>avg;
      cout<<"Enter number of wheels : ";
cin>>no_of_wheels;
   }
   void putdata()
   {
      cout<<"Average of truck = "<<avg<<endl;
      cout<<"Number of wheels = "<<no_of_wheels<<endl;
   }
};
int main()
{
   vehicle *vptr;
   car c;
   truck t;
vptr=&c;    vptr-
>getdata();    vptr-
>putdata();
vptr=&t;    vptr-
>getdata();    vptr-
>putdata();
   return 0;
}
```

Error :

No

Output:(Screenshot)



What You have learned from the Program:

I learnt the concept of Abstract Base class and Pure Virtual functions.

P39

Write a program that creates a text file that contains ABC…Z. A program should print the file in reverse order on the screen. i.e. ZYX…BA. Use concept of Opening the file using constructor and open() function. Use all error handling functions like eof() , fail() , bad() , good() and functions for manipulation of file pointer like seekg() and tellg().

Source Code:

```
#include<iostream>
#include<fstream>
using namespace std;
int main()
{
   char a='A';
ofstream abc("abc.txt");
for(int i=0;i<26;i++)
{
   abc<<a; a++;
}
abc.close();
ifstream file1("abc.txt");
ofstream file2("cba.txt");
char b[1];
file1.seekg(1,ios::end);
for(int i=0;i<26;i++)
{
```

d

```
   char ch;

file1.seekg(-2L,ios::cur);

file1.get(ch);file2<<ch;
}
file2.close();
file1.close();
ifstream file4("abc.txt");
ifstream file3("cba.txt");
char p[100];
file3.getline(p,100);
cout<<p;
 return 0;
}
```

Error :

No

Output:(Screenshot)

What You have learned from the Program:

I learnt error handling functions like eof() , fail() , bad() , good() and functions for manipulation of file pointer like seekg() and tellg().