TAMAL DATTA

23/05/2020

1. Write a C++ program to Overload unary ' - ' operator.
Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
   int a;
   float b;
   public:

   abc(){}
   abc(int x,float y):a(x),b(y){}

   abc operator - ()
   {
      abc t;
      t.a=-this->a;
      t.b=-this->b;
      return t;
   }

   void print()
   {
      cout<<a<<"\t"<<b<<endl;
   }
};

int main()
{
   abc ob(5,5.6),ob1;
   ob1=-ob;
   ob.print();
   ob1.print();
}
```

2. Write a C++ program to implement Private Constructor.

Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
   int a;
   float b;

   abc(){}     //private default constructor
   abc(int x,float y):a(x),b(y){}      //private parameterized constructor

   public:
   friend int main();
   void print()
   {
      cout<<a<<"d\t"<<b<<endl;
   }
};

int main()
{
   abc ob(5,5.6);
   ob.print();

}
```

3. Write a C++ program to create class and read and add two distance.
Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
   int distance;
   public:
   abc(){}
   abc(int x):distance(x){}

   abc operator +(abc t)
   {
      return t.distance+this->distance;
   }
   void print()
   {
      cout<<distance<<endl;
```

```
    }
};

int main()
{
    int n1,n2;
    cout<<"distance of ob1?"<<endl;
    cin>>n1;
    cout<<"distance of ob2?"<<endl;
    cin>>n2;

    abc ob1(n1),ob2(n2),ob;
    cout<<"ob1=";
    ob1.print();
    cout<<"ob2=";
    ob2.print();
    cout<<"before addition :ob=";
    ob.print();
    ob=ob1+ob2;
    cout<<"after addition :ob=";
    ob.print();

}
```

4. Write a C++ program to demonstrate the example of friend functions with class.
Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
    int a;
    public:
    friend void print(abc);
    abc();
};
abc::abc()
{
    cout<<"enter objeect value"<<endl;
    cin>>a;
}

void print(abc x)
{
    cout<<"value of object is "<<x.a<<endl;
    cout<<"above line is printed by friend function"<<endl;
```

```
}

int main()
{
   abc ob;
   print(ob);
}
```

5. Create an object of a class inside another class declaration in C++(Nested class).
Soln:

```
#include<iostream>
using namespace std;
class A{
   int a;
   public:
   A(){}
   A(int x):a(x){}

   friend ostream& operator <<(ostream& out,A x);
};


ostream& operator <<(ostream& out,A x){
   out<<x.a<<endl;
   return out;
}


class B{
   int b;
   A aa;
   public:
   B(){}
   B(int b,A c):b(b),aa(c){}

   void print()
   {
      cout<<"member of class A="<<aa<<"exclusive member of class B="<<b<<endl;
   }

};

int main()
```

```
{
   B obj(5,6);
   obj.print();
}
```

6. Write a C++ program passing an object to a non member function in C++.
Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
   int a;
   public:
   abc(int x):a(x){}
   abc(){}
   friend void print(abc);
};

void print(abc t)
{
   cout<<"it's non-member function"<<endl;
   cout<<"value is "<<t.a<<endl;
}

int main()
{
   abc ob(5);
   print(ob);

}
```

7. Write a C++ proram to Access the address of an object using 'this' pointer in C++.
Soln:

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

class abc{
   int a;
   float b;
   public:
   abc(){}
   abc(int x,float y):a(x),b(y){}
   void addr()
```

```
    {
        cout<<"address of object is "<<this<<endl;
    }

};

int main()
{
    abc ob(5,6.6);
    ob.addr();
}
```

8. Write a C++ program to add two class objects using binary plus(+) operator overloading.
Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
    int a;
    public:
    abc(int x):a(x){}
    abc(){}
    abc operator + (abc t)
    {
        return t.a+this->a;
    }

    void print()
    {
        cout<<a<<endl;
    }
};

int main()
{
    abc ob1(5),ob2(6),ob;
    ob1.print();
    ob2.print();
    ob=ob1+ob2;
    ob.print();
}
```

9. Write a C++ program for unary increment (++) and decrement(--) operator overloaded.

Soln:

```
#include <iostream>
#include<string>
using namespace std;
class abc{
   int a;
   int b;
   public:
   abc(int x,int y):a(x),b(y){}
   abc(){}
   abc operator ++ ()
   {
      a=a+1;b=b+1;
      return *this;
   }

   abc operator ++(int t)
   {
      abc temp;
      temp.a=a;
      temp.b=b;
      a=a+1;b=b+1;
      return temp;
   }

      abc operator -- ()
   {
      a=a-1;b=b-1;
      return *this;
   }

   abc operator --(int t)
   {
      abc temp;
      temp.a=a;
      temp.b=b;
      a=a-1;b=b-1;
      return temp;
   }

   void print()
   {
      cout<<a<<"\t"<<b<<endl;
   }
};
```

```
int main()
{
   abc ob(5,6),ob1,ob2,ob3,ob4;
   cout<<"ob=";
   ob.print();

   ob1=ob++;
   cout<<"ob1=";
   ob1.print();
   cout<<"ob=";
   ob.print();

   ob2=++ob;
   cout<<"ob2=";
   ob2.print();
   cout<<"ob=";
   ob.print();

   ob3=ob--;
   cout<<"ob3=";
   ob3.print();
   cout<<"ob=";
   ob.print();

   ob4=--ob;
   cout<<"ob4=";
   ob4.print();
   cout<<"ob=";
   ob.print();
}
```

10. write a C++ to Generate random numbers.
Soln:

```
#include <iostream>
#include<string>
#include<time.h>
using namespace std;
int main()
{
   srand(time(0));
   int n,n1,n2,temp1;
   cout<<"how many random numbers to generate?"<<endl;
   cin>>n;
   cout<<"lower range?"<<endl;
```

```
   cin>>n1;
   cout<<"upper range?"<<endl;
   cin>>n2;
   cout<<n<<" random no.s between "<<n1<<" & "<<n2<<" are:"<<endl;
   for(int i=0;i<n;i++)
      {
      temp1=rand()%(n2-n1);
      cout<<temp1+n1<<"\t";
      }
}
```

11. Write a C++ program to read and print employee information with depratment and pf information using hierarchical inheratiance.
Soln:

```
#include<iostream>
using namespace std;
class details;
class emp{          //base class
   public:
   char name[20];
   emp(){
      cout<<"enter name"<<endl;
      cin.getline(name,20);
   }
};
////////// hiararchical inheritance//
class dept:virtual public emp{  //derived1
   public:
   char dep[20];
   dept()
   {
      cout<<"enter dept"<<endl;
      cin.getline(dep,20);
   }
};

class pf:virtual public emp{    //derived2
   public:
   int p;
   pf()
   {
      cout<<"enter pf"<<endl;
      cin>>p;
   }
```

```
};

// this class is to print the contents of two hiararchically    inherited class//
class details:public dept,public pf{
   public:
   void print()
   {
      cout<<"employee
name="<<name<<endl<<"department="<<dep<<endl<<"pf="<<p<<endl;
   }
};

int main()
{
  details obj;
  obj.print();
}
```

12. Write a C++ Program to exchange values of two variables b/w two classes using friend function.
Soln:

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

class abc{
   int a,b;
   public:
   abc(){
      cout<<"enter two integers"<<endl;
      cin>>a>>b;
   }
   void print(){
      cout<<"a="<<a<<"\tb="<<b<<endl;
   }

   friend void swap(abc&,abc&);
};

void swap(abc&t1,abc&t2)
{
   t1.a=t2.a-t1.a+(t2.a=t1.a);
   t1.b=t2.b-t1.b+(t2.b=t1.b);
}
```

```
int main()
{
cout<<"ob1:"<<endl;
abc ob1;
cout<<"ob2:"<<endl;
abc ob2;
ob1.print();
ob2.print();
swap(ob1,ob2);
cout<<"after swaping"<<endl;
ob1.print();
ob2.print();
}
```

13. Write a C++ program to create a class complex with real and imaginary parts perform addition and subtraction of two complex objects.
Soln:

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

static int flag=1;
class complex{
   float x;
   float y;
   public:
   complex()
   {
      if(flag){
      char op;
      cout<<"Want to enter value of object? Y/N"<<endl;
      cin>>op;
      if((op=='y' || op=='Y') )
      {
      cout<<"enter real part"<<endl;
      cin>>x;
      cout<<"enter imag part"<<endl;
      cin>>y;
      }
      }
   }
   void print()
   {
      cout<<x<<"+j"<<y<<endl;
   }
```

```
    complex operator +(complex t)
    {
       complex temp;
       temp.x=t.x+x;
       temp.y=t.y+y;
       return temp;
    }

    complex operator -(complex t)
    {
       complex temp;
       temp.x=x-t.x;
       temp.y=y-t.y;
       return temp;
    }

};

int main()
{
    cout<<"object1 "<<endl;
    complex ob1;
    cout<<"object2 "<<endl;
    complex ob2;
    flag=0;
    complex ob;
    cout<<"ob1: ";
    ob1.print();
    cout<<"ob2: ";
    ob2.print();
    ob=ob1+ob2;
    cout<<"summation= ";
    ob.print();
    cout<<"subtraction= ";
    ob=ob1-ob2;
    ob.print();
  }
```

14. Write a C++ program to sort the given five strings from the keyboard and print it in the sorted order. (Use C++'s DMA).

Soln:

```
#include<iostream>
#include<cstring>
using namespace std;
class str{
   char **a=new char*[5];
   public:
   str()
   {
      int i=0;
      cout<<"enter 5 strings"<<endl;
      while(i<5)
      {
         a[i]=new char[50];
         cin.getline(a[i],50);
         i++;
      }
   }

   /*
   ~str()
   {
      cout<<"destructor"<<endl;
      int i=0;
      while(i<5);
         {
            delete[] a[i];
            i++;
         }
      delete[] a;
   }
   */

   void print()
   {
      int i=0;
      while(i<5)
      {
         cout<<this->a[i]<<endl;
         i++;
      }
   }

   void sort()
   {
```

```
      int i=0,j=0;
      for(i=0;i<5-1;i++)
         for(j=i;j<5;j++)
            if(strcmp(this->a[i],this->a[j])>0)
               {
                  char temp[50];
                  strcpy(temp,this->a[i]);
                  strcpy(this->a[i],this->a[j]);
                  strcpy(this->a[j],temp);
               }


   }
   friend void sort(str&);
};

  void sort(str& abc)
   {
      int i=0,j=0;
      for(i=0;i<5-1;i++)
         for(j=i;j<5;j++)
            if(strcmp(abc.a[i],abc.a[j])>0)
               {
                  char temp[50];
                  strcpy(temp,abc.a[i]);
                  strcpy(abc.a[i],abc.a[j]);
                  strcpy(abc.a[j],temp);
               }
   }
int main()
{
   str obj;
   cout<<"before sorting"<<endl;
   obj.print();

   obj.sort();
   cout<<"after sorting using member fun"<<endl;
   obj.print();

   sort(obj);
   cout<<"after sorting using friend fun"<<endl;
   obj.print();
}
```

15. Write a C++ program to define virtual destructors with example ?
Soln:

```
#include<iostream>
using namespace std;
class b{
   char *s;
   public:
   b(){
      s=new char[10];
   }

   virtual ~b(){
      delete []s;
      cout<<"dist. of base"<<endl;
   }
};

class d1:public b{
   char *p1;
   public:
   d1(){
      p1=new char[10];
      }
   ~d1()
   {
      delete []p1;
      cout<<"dest. of d1"<<endl;
   }
};

class d2:public d1{
   char *p2;
   public:
   d2(){
      p2=new char[10];
      }
   ~d2()
   {
      delete []p2;
      cout<<"dest. of d2"<<endl;
   }
};


int main(){
   b *obj;
   obj=new d2;
   delete obj;
```

```
}
```

Here, as base class (b) destructor is virtual therefore here destructor of d1,d1,b all are called,, otherwise only base(b) destructor would be called.

16. Write a C++ program to convert data in a file to opposite case in same file?
Soln:

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

int main()
{
   fstream ifs("def");   // def is the name of file
   if(ifs.fail())
      {
         cout<<"file not present"<<endl;
         return -1;
      }
   char c;
   int i=0,j=0;
   ifs.seekg(0,ios::end);
   char arr[(int)ifs.tellg()];
   ifs.seekg(0,ios::beg);
   while((c=ifs.get())!=-1)
      if((c>='a' && c<='z')||(c>='A' && c<='Z'))
         arr[i++]=c^(1<<5);
      else
         arr[i++]=c;
   ifs.close();

   ofstream ofs("def");
   while(j<i)
      {
         ofs<<arr[j];
         j++;
      }
   ofs.close();
}
```

17. Write a Program to merge two files data character by character into third file(check file1 > file2, file1== file2,file1 <file2)
Soln:

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

int main()
{
   ifstream ifs1("abc"),ifs2("def");   // abc, def are two i/p files
    if(ifs1.fail())
      {
         cout<<"abc not present"<<endl;
         return -1;
      }
    if(ifs2.fail())
      {
         cout<<"def not present"<<endl;
         return -1;
      }



   ofstream ofs("destination");
   char c1,c2;
   while(1)
      {
         if((c1=ifs1.get())!=-1)
            ofs<<c1<<" ";
         if((c2=ifs2.get())!=-1)
            ofs<<c2<<" ";

         if(c1==-1 && c2==-1 )
            break;
      }
   ifs1.clear();
   ifs2.clear();
   ifs1.seekg(0,ios::end);
   ifs2.seekg(0,ios::end);
   if(ifs1.tellg()>ifs2.tellg())
      cout<<"len. of file abc > len. of file def";
   else if(ifs1.tellg()<ifs2.tellg())
      cout<<"len. of file abc < len. of file def";
   else
      cout<<"len. of file abc = len. of file def";
   ifs1.close();
   ifs2.close();
   ofs.close();
```

```
}
```

18. Write a Program to merge two files data word by word into third file(check file1 > file2, file1== file2,file1 < file2)
Soln:

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

int main()
{
   ifstream ifs1("abc"),ifs2("def");   // abc, def are two i/p files
   if(ifs1.fail())
      {
         cout<<"abc not present"<<endl;
         return -1;
      }
    if(ifs2.fail())
      {
         cout<<"def not present"<<endl;
         return -1;
      }

   ofstream ofs("destination");
   char s1[10],s2[10];
   while(1)
      {
         ifs1>>s1;
         ifs2>>s2;
         if(ifs1.tellg()!=-1)
            ofs<<s1<<" ";
         if(ifs2.tellg()!=-1)
            ofs<<s2<<" ";

         if(ifs1.tellg()==-1 && ifs2.tellg()==-1 )
            break;
      }
   ifs1.clear();
   ifs2.clear();
   ifs1.seekg(0,ios::end);
   ifs2.seekg(0,ios::end);
   if(ifs1.tellg()>ifs2.tellg())
      cout<<"len. of file abc > len. of file def";
   else if(ifs1.tellg()<ifs2.tellg())
```

```
    cout<<"len. of file abc < len. of file def";
  else
    cout<<"len. of file abc = len. of file def";
  ifs1.close();
  ifs2.close();
  ofs.close();
}
```

19. Write a Program to merge two files data line by line into third file(check file1 > file2, file1== file2,file1 < file2).
Soln:

```
#include<iostream>
#include<fstream>
#include<cstring>
using namespace std;
int main()
{
  ifstream ifs1("abc");          // input file1 ->  abc
  if(!ifs1.is_open())
     {
     cout<<"abc file not present"<<endl;
     return -1;
     }
  ifstream ifs2("def");          // input file2 ->  def
  if(!ifs2.is_open())
     {
     cout<<"def file not present"<<endl;
     return -1;
     }

  ofstream ofs("destination");
  char s1[50],s2[50];
  int c1=0,c2=0;
  while(1)
     {
        ifs1.getline(s1,50);
        ifs2.getline(s2,50);
        if(ifs1)
           {
           ofs<<s1<<endl;
           c1+=strlen(s1);
           }
        if(ifs2)
           {
           ofs<<s2<<endl;
```

```
            c2+=strlen(s2);
            }
          if(!(ifs1||ifs2))
            break;
       }
   if(c1>c2)
      cout<<"len of file abc >len of  file def";
   else if(c2>c1)
      cout<<"len of file abc < len of file def";
   else
      cout<<"len of file abc = len of  file def";
ifs1.close();
ifs2.close();
ofs.close();
}
```

20. WAP to make perticular member function of a class as friend to another class.
Soln:

```
#include<iostream>
using namespace std;
class A;
class B;
class A{
   int a;
   public:
   void get(B& t);
   void set(B& t);
};
class B{
   int b;
   public:
   friend void A::get(B& t);
   friend void A::set(B& t);
};

void A::get(B& t)
   {
      cout<<t.b<<endl;
   }

   void A::set(B& t)
   {
      cout<<"enter data"<<endl;
      cin>>t.b;
   }
```

```
int main()
{
  A obj1;
  B obj2;
  obj1.set(obj2);
  obj1.get(obj2);
}
```