****

Assignment No-1

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

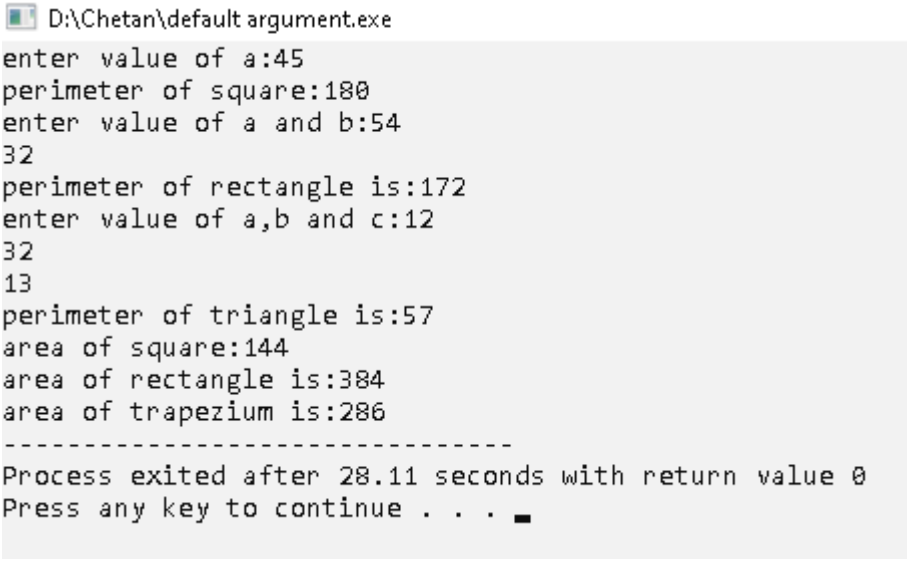
Q-1). Function Default argument.

i)To calculate perimeter of square(4*r),rectangle(2*l+2*b),triangle(a+b+c)
ii)To calculate area of square(r*r),rectangle (l*b),trapezium (1/2*h*(s1+s2))

```
#include<iostream>
using namespace std;
perimeter(double a,double b=0,double c=0)
{
	if(b==0&&c==0)
	{
		return(4*a);
	}
	if(c==0)
	{
		return(a*2+b*2);
	}
	return(a+b+c);
}
area(double a,double b=0,double c=0)
{
	if(b==0&&c==0)
	{
		return(a*a);
	}
	if(c==0)
	{
		return(a*b);
	}
	return(c/2*(a+b));
}
int main()
{
	doublea,b,c;
	cout<<"enter value of a:";
	cin>>a;
	cout<<"perimeter of square:"<<perimeter(a)<<endl;
	cout<<"enter value of a and b:";
	cin>>a>>b;
	cout<<"perimeter of rectangle is:"<<perimeter(a,b)<<endl;
	cout<<"enter value of a,b and c:";
	cin>>a>>b>>c;
```

```
        cout<<"perimeter of triangle is:"<<perimeter(a,b,c)<<endl;
        cout<<"area of square:"<<area(a)<<endl;
        cout<<"area of rectangle is:"<<area(a,b)<<endl;
        cout<<"area of trapezium is:"<<area(a,b,c);
        return 0;
}
```

**********Output**********



****

****

Assignment No-2

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

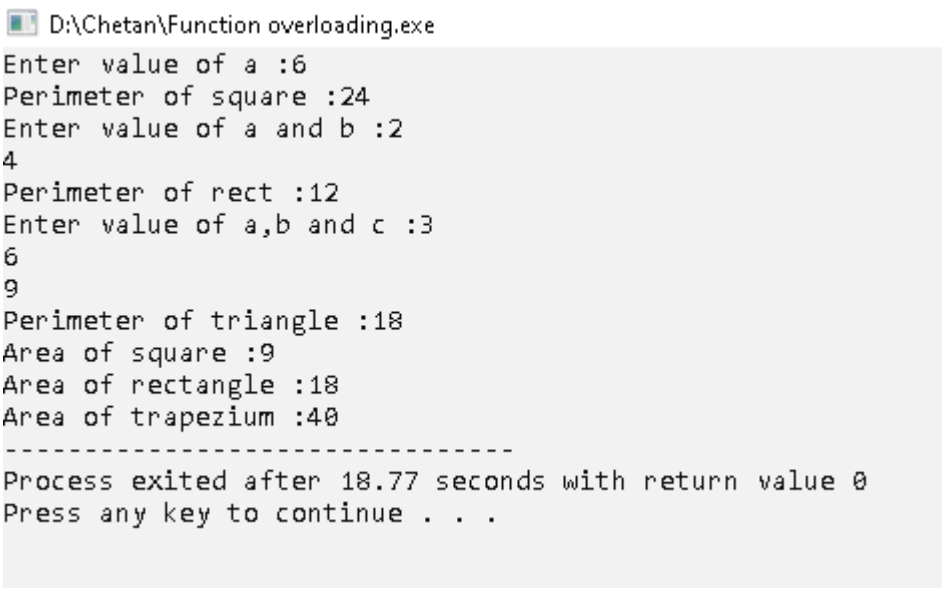
Q-1). Function Overloading.

i)To calculate perimeter of square(4*r),rectangle(2*l+2*b),triangle(a+b+c)
ii)To calculate area of square(r*r),rectangle (l*b),trapezium (1/2*h*(s1+s2))

```
#include<iostream>
using namespace std;
perimeter (double a)
{
	return(4*a);
}
perimeter (double a,double b)
{
	return(a*2+b*2);
}
perimeter (double a,double b,double c)
{
	return(a+b+c);
}
area (double a)
{
	return(a*a);
}
area (double a,double b)
{
	return(a*b);
}
area (double a,double b,double c)
{
	return(c/2*(a+b));
}
int main()
{
	double a,b,c;
	cout<<"Enter value of a :";cin>>a;
	cout<<"Perimeter of square :"<<perimeter(a)<<endl;cout<<"Enter value of a and b
:";cin>>a>>b;
	cout<<"Perimeter of rect :"<<perimeter(a,b)<<endl;cout<<"Enter value of a,b and c
:";cin>>a>>b>>c;
	cout<<"Perimeter of triangle :"<<perimeter(a,b,c)<<endl;cout<<"Area of square
:"<<area(a)<<endl;
```

```
	cout<<"Area of rectangle :"<<area(a,b)<<endl;cout<<"Area of trapezium
:"<<area(a,b,c);
	return 0;
}
```

**********Output**********



****

****

Assignment No-3

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

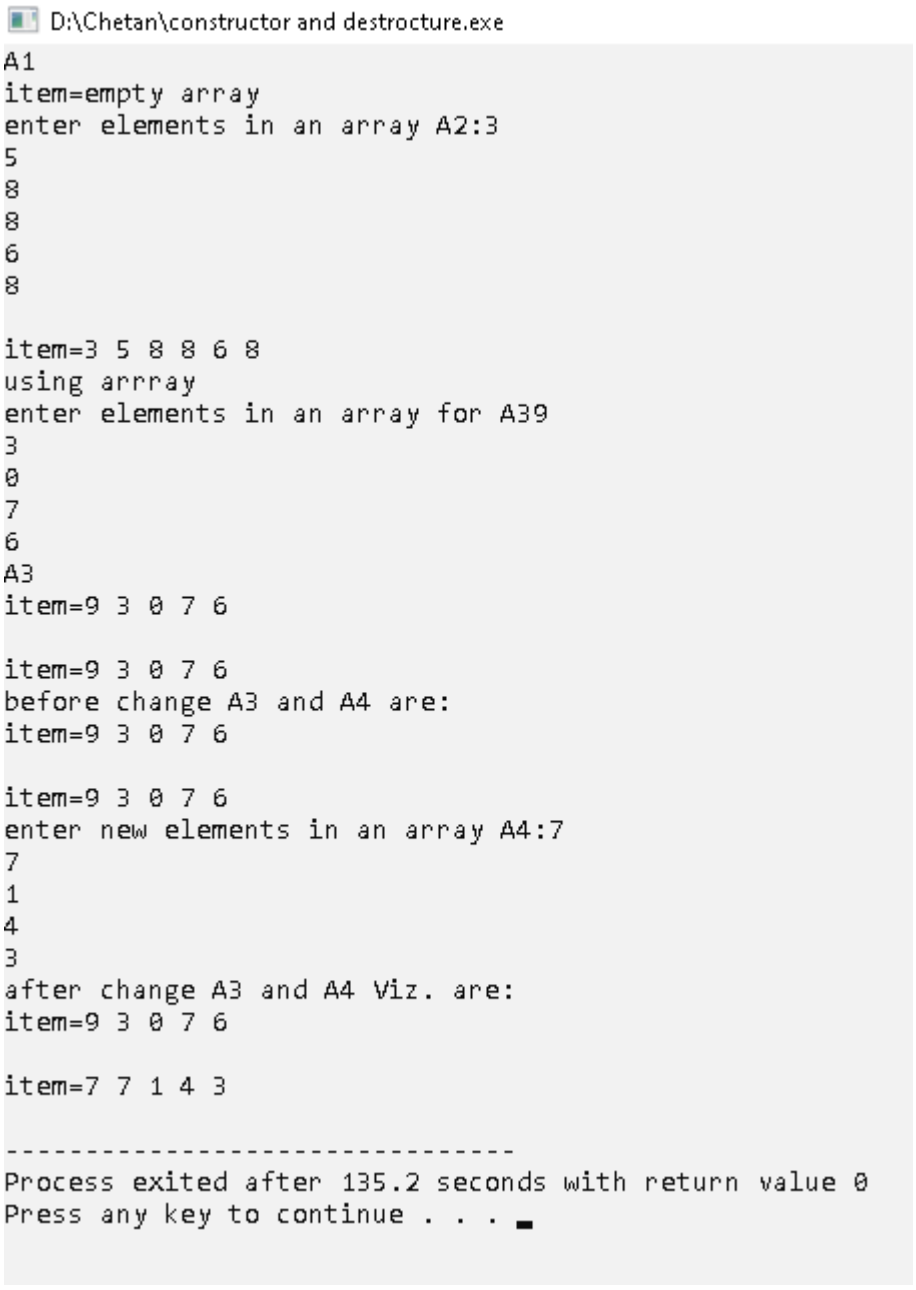
Q-1). Constructor And Destructor.

Q)Demonstrate working of constructor (default, parameterized, copy) and destructor to allocate and de-allocate memory to or from an array of integers using DMA operators new and delete.

```
#include<iostream>
using namespace std;
classmyarray
{
int *arr,l; public:
myarray()
{
l=0;
arr=NULL;
}
myarray(intll)
{
l=ll;
arr=new int[ll];
}
~ myarray()
{
if(l!=0) delete []arr; arr=NULL;
}
myarray(int [],int); myarray(myarray&); int&item(int index)
{
return(arr[index]);
}
voidinsertitem(); void displayitem();
};
myarray::myarray(int A[],intll)
{
l=ll;
arr=new int [l];
for(int index=0;index<l;index++) arr[index]=A[index];
}
myarray::myarray(myarray&m)
{
l=m.l;
arr=new int [l];
for(int index=0;index<l;index++) arr[index]=m.arr[index];
```

```
}
voidmyarray::insertitem()
{
for(int index=0;index<l;index++) cin>>arr[index];
}
voidmyarray::displayitem()
{
cout<<endl<<"item="; if(l==0)
cout<<"empty array";
for(int index=0;index<l;index++) cout<<arr[index]<<" "; cout<<endl;
}
int main(intargc,char *argv[])
{
myarray A1; cout<<"A1"; A1.displayitem(); myarray A2(6);
cout<<"enter elements in an array A2:"; for(int index=0;index<6;index++)
cin>>A2.item(index);
intarr[5];
A2.displayitem();
cout<<"using arrray"<<endl;
cout<<"enter elements in an array for A3"; for(int index=0;index<5;index++)
cin>>arr[index];
myarray A3(arr,sizeof(arr)/sizeof(arr[0])); cout<<"A3";
A3.displayitem(); myarray A4=A3; A4.displayitem();
cout<<"before change A3 and A4 are:"; A3.displayitem();
A4.displayitem();
cout<<"enter new elements in an array A4:"; A4.insertitem();
cout<<"after change A3 and A4 Viz. are:"; A3.displayitem();
A4.displayitem();
return 0;
}
```

**********Output**********


D:\Chetan\constructor and destrocture.exe
A1
item=empty array
enter elements in an array A2:3
5
8
8
6
8
item=3 5 8 8 6 8
using arrray
enter elements in an array for A39
3
0
7
6
A3
item=9 3 0 7 6
item=9 3 0 7 6
before change A3 and A4 are:
item=9 3 0 7 6
item=9 3 0 7 6
enter new elements in an array A4:7
7
1
4
3
after change A3 and A4 Viz. are:
item=9 3 0 7 6
item=7 7 1 4 3

Process exited after 135.2 seconds with return value 0
Press any key to continue . . .

****

****

Assignment No-4

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

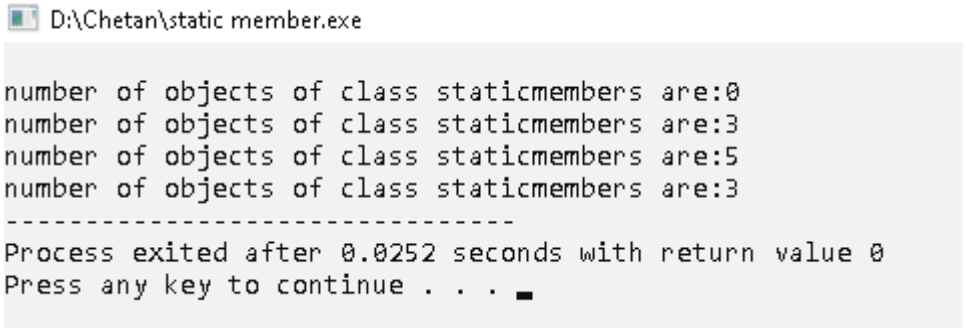
****

Q-1)Static members.

Q)Display counter which counts numbers of objects of class, counter is incremented in constructor and decremented in destructor.

```
#include<iostream>
using namespace std;
classstaticmembers
{
staticintcnt; public:
staticmembers()
{
cnt++;
}
~staticmembers()
{
cnt--;
}
static void displaycounter()
{
cout<<endl<<"number of objects of class staticmembers are:"<<cnt;
}
};
intstaticmembers::cnt=0;
int main(intargc,char *argv[])
{
staticmembers::displaycounter(); staticmembers A,B,C; staticmembers::displaycounter();
{
staticmembers D,E; staticmembers::displaycounter();
}
staticmembers::displaycounter();
return 0;
}
```

**********Output**********


D:\Chetan\static member.exe
number of objects of class staticmembers are:0
number of objects of class staticmembers are:3
number of objects of class staticmembers are:5
number of objects of class staticmembers are:3

Process exited after 0.0252 seconds with return value 0
Press any key to continue . . .

****

****

Assignment No-5

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

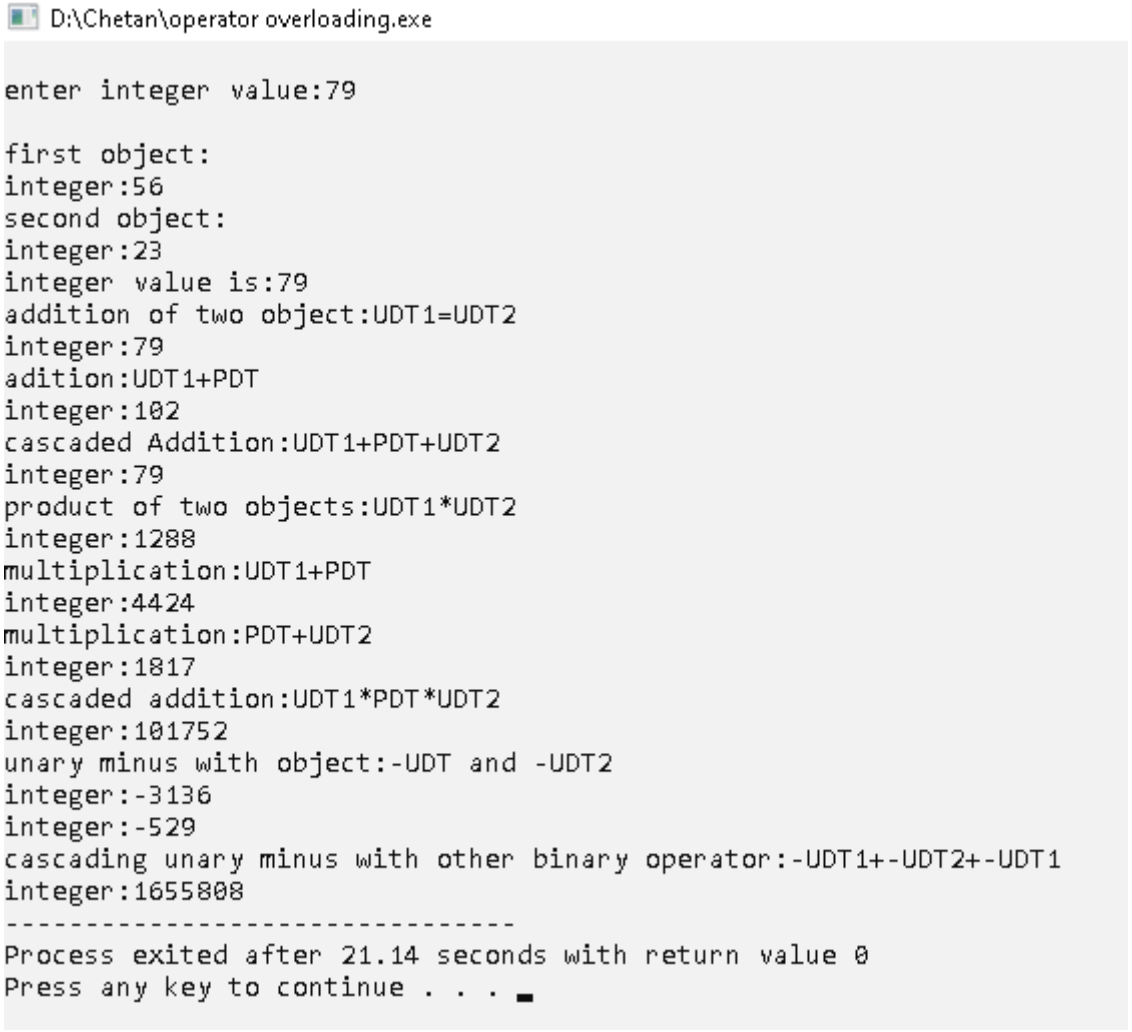
Q-1). Operator Overloading.

Q)To overload addition, multiplication, unary minus operator on class Integer

```
#include<iostream>
using namespace std;
class integer
{
int I; public:
integer()
{
I=0;
}
integer(int i)
{
I=i;
}
voidgetdata(int i)
{
I=i;
}
voidputdata()
{
cout<<endl<<"integer:"<<I;
}
integer operator -()
{
integer R;
R.I=I*-I;
return R;
}
integer operator +(integer RHO)
{
integer R;
R.I=I+RHO.I;
return R;
}
integer operator +(int i)
{
integer RHO(i); return ((*this)+RHO);
}
```

```
friend integer operator +(int,integer); integer operator *(integer RHO)
{
integer R;
R.I=I*RHO.I;
return R;
}
integer operator *(int i)
{
integer RHO(i); return ((*this)*RHO);
}
friend integer operator *(int,integer);
};
integer operator +(inti,integer RHO)
{
return(RHO+i);
}
integer operator *(inti,integer RHO)
{
return(RHO*i);
}
int main(intargc,char *argv[])
{
int a;
cout<<endl<<"enter integer value for first object:"; cin>>a;
integer objI1(a);
cout<<endl<<"enter integer value for second object:";
cin>>a;
integer objI2(a); cout<<endl<<"enter integer value:"; cin>>a;
cout<<endl<<"first object:"; objI1.putdata(); cout<<endl<<"second object:"; objI2.putdata();
cout<<endl<<"integer value is:"<<a;
cout<<endl<<"addition of two object:UDT1=UDT2"; integer objI3=objI1+objI2;
objI3.putdata(); cout<<endl<<"adition:UDT1+PDT"; objI3=a+objI2;
objI3.putdata();
cout<<endl<<"cascaded Addition:UDT1+PDT+UDT2"; objI3=objI1+objI2;
objI3.putdata();
cout<<endl<<"product of two objects:UDT1*UDT2"; objI3=objI1*objI2;
objI3.putdata(); cout<<endl<<"multiplication:UDT1+PDT"; objI3=objI1*a;
objI3.putdata(); cout<<endl<<"multiplication:PDT+UDT2"; objI3=a*objI2;
objI3.putdata();
cout<<endl<<"cascaded addition:UDT1*PDT*UDT2"; objI3=objI1*a*objI2;
objI3.putdata();
cout<<endl<<"unary minus with object:-UDT and -UDT2"; objI3=-objI1;
objI3.putdata(); objI3=-objI2; objI3.putdata();
cout<<endl<<"cascading unary minus with other binary operator:-UDT1+-UDT2+-UDT1";
objI3=-objI1+-objI2*-objI1;
objI3.putdata();
return 0;
}
```

**********Output**********


D:\Chetan\operator overloading.exe
enter integer value:79
first object:
integer:56
second object:
integer:23
integer value is:79
addition of two object:UDT1=UDT2
integer:79
adition:UDT1+PDT
integer:102
cascaded Addition:UDT1+PDT+UDT2
integer:79
product of two objects:UDT1*UDT2
integer:1288
multiplication:UDT1+PDT
integer:4424
multiplication:PDT+UDT2
integer:1817
cascaded addition:UDT1*PDT*UDT2
integer:101752
unary minus with object:-UDT and -UDT2
integer:-3136
integer:-529
cascading unary minus with other binary operator:-UDT1+-UDT2+-UDT1
integer:1655808
Process exited after 21.14 seconds with return value 0
Press any key to continue . . .

****

****

Assignment No-6

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

Q.1) Operator Overloading.

Q)To overload Type Cast operator to convert temperature in Degree Celsius To Degree Fahrenheit and Degree Fahrenheit To Degree Celsius using classes Celsius and Fahrenheit.

F=9/5^ * c + 32

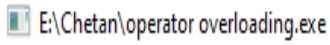
```
#include<iostream>
using namespace std;
class fahrenheit;
class celcius
{
double c;
public:
celcius()
{
c=0;
}
celcius(fahrenheit); operator fahrenheit();

friend istream& operator>>(istream&,celcius&);
friend ostream& operator<<(ostream&,celcius);
};
istream& operator>>(istream& din,celcius& objc)
{
din>>objc.c;
return din;
}
ostream& operator<<(ostream& dout,celcius objc)
{
dout<<endl<<"temperature in degree celcius is:"<<objc.c;
return dout;
}
class fahrenheit
{
double f;
public:
fahrenheit()
{
f=0;
}
```

```
friend class celcius;
friend istream& operator>>(istream&,fahrenheit&);
friend ostream& operator<<(ostream&,fahrenheit);
};
istream& operator>>(istream&din,fahrenheit&objf)
{
din>>objf.f;
return din;
}
ostream& operator <<(ostream& dout,fahrenheit objf)
{
cout<<endl<<"temerature in degree fahrenheit is:"<<objf.f;
return dout;
}
celcius::celcius(fahrenheit objf)
{
c=(objf.f-32.0)*5.0/9.0;
}
celcius::operator fahrenheit()
{
fahrenheit objf; objf.f=c*9.0/5.0+32.0;
return objf;
}
int main(int argc,char *argv[])
{
celcius objc1;

cout<<endl<<"enter temperature in degree celcius:";
cin>>objc1;
fahrenheit objf1;
cout<<endl<<"enter temperature in degree fahrenheit:";
cin>>objf1;
cout<<"temperatures are:"<<objc1<<objf1;
cout<<endl<<endl<<"tyecastoperator:conversion from fahrenheit to celcius"<<"using
constructor i.e. for destination object";
celcius objc2=objf1; cout<<objc2;
cout<<endl<<endl<<"typecast operator:conversion from celcius to fahrenheit"
<<"using operator function i.e. for source object";
fahrenheit objf2=objc2;
fahrenheit(); cout<<objf2;
return 0;
}
```

**********Output**********




```
enter temperature in degree celcius:25

enter temperature in degree fahrenheit:98
temperatures are:
temperature in degree celcius is:25
temerature in degree fahrenheit is:98

tyecastoperator:conversion from fahrenheit to celciususing constructor i.e. for destination object
temperature in degree celcius is:36.6667

typecast operator:conversion from celcius to fahrenheitusing operator function i.e. for source object
temerature in degree fahrenheit is:98
--------------------------------
Process exited after 17.24 seconds with return value 0
Press any key to continue . . .
```

****

****

Assignment No-7

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

Q.1 Pure virtual function and inheritance.

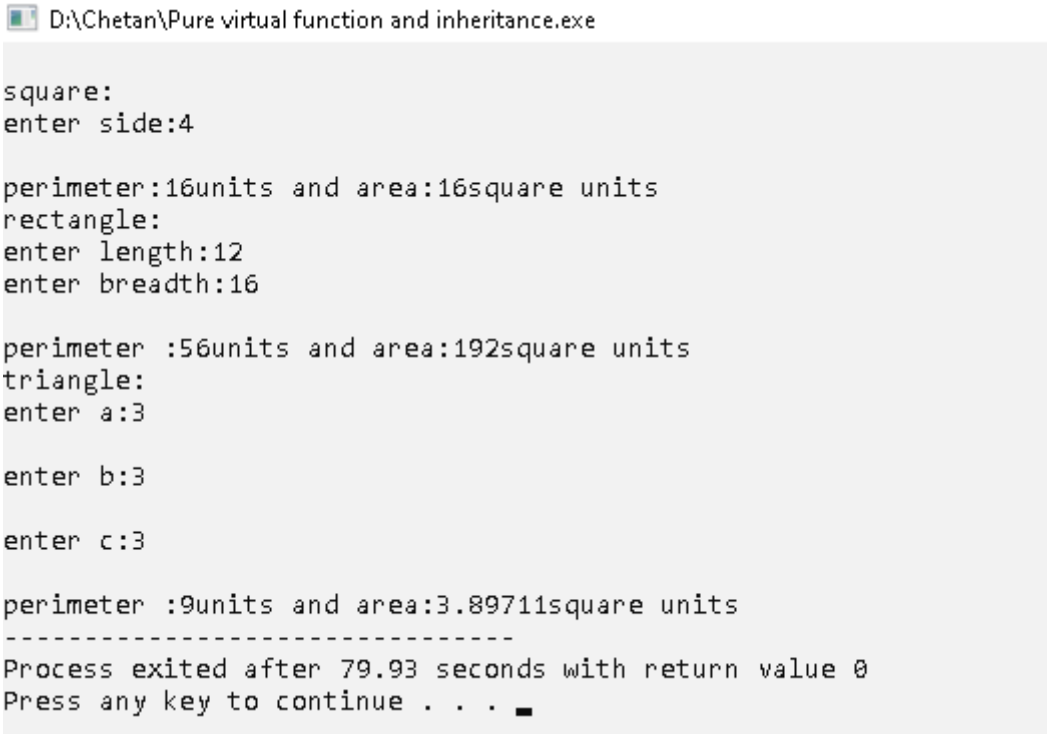
Q) To specify base class Shape with pure virtual methods Input(), Perimeter() and Area(). Inherit three classes Square, Rectangle and Triangle from class Shape with appropriate data members and override methods Input(), Perimeter() and Arca(). Use Pointer of class Shape to access objects of Three classes and Demonstrate working in "main" function**.**

```
#include<iostream>
#include<math.h>
using namespace std;
class shape
{
protected:
double *sides; unsigned n; public:
shape()
{
n=0;
sides=NULL;
}
shape(unsigned nn)
{
n=nn;
sides=new double[n];
}
virtual ~shape()
{
if(n>0)
delete []sides;
}
virtual void input()=0;
virtual double perimeter()=0; virtual double area()=0;
};
classsquare:public shape
{
public:
square():shape(1){
}
~square()
{
}
void input()
```

```
{
cout<<endl<<"enter side:"; cin>>sides[0];
}
double perimeter()
{
return 4*sides[0];
}
double area()
{
return sides[0]*sides[0];
}
};
classrectangle:public shape
{
public:
rectangle():shape(2)
{
}
~rectangle()
{
}
void input()
{
cout<<endl<<"enter length:"; cin>>sides[0];
cout<<"enter breadth:"; cin>>sides[1];
}
double perimeter()
{
return 2*sides[0]+2*sides[1];
}
double area()
{
return sides[0]*sides[1];
}
};
classtriangle:public shape
{
public:
triangle():shape(3)
{
}
~triangle()
{
}
void input()
{
cout<<endl<<"enter a:"; cin>>sides[0]; cout<<endl<<"enter b:";
cin>>sides[1]; cout<<endl<<"enter c:"; cin>>sides[2];
}
double perimeter()
```

```
{
return sides[0]+sides[1]+sides[2];
}
double area()
{
double s=(sides[0]+sides[1]+sides[2])/2;
double A2=s*(s-sides[0])*(s-sides[1])*(s-sides[2]); if(A2<0)
{
cout<<endl<<"invalid triangle sides:"<<endl; return 0.0;
}
double A; A=sqrt(A2); return A;
}
};
int main(intargc,char *argv[])
{
        shape *ptrshp;
        cout<<endl<<"square:";
        ptrshp=new square();
        ptrshp->input();
        cout<<endl<<"perimeter:"<<ptrshp->perimeter()<<"units and area:"<<ptrshp-
>area()<<"square units";
        deleteptrshp;
        cout<<endl<<"rectangle:";
        ptrshp=new rectangle();
        ptrshp->input();
        cout<<endl<<"perimeter :"<<ptrshp->perimeter()<<"units and area:"<<ptrshp-
>area()<<"square units";
        deleteptrshp;
        cout<<endl<<"triangle:";
        ptrshp=new triangle();
        ptrshp->input();
        cout<<endl<<"perimeter :"<<ptrshp->perimeter()<<"units and area:"<<ptrshp-
>area()<<"square units";
return 0;
}
```

**********Output**********


D:\Chetan\Pure virtual function and inheritance.exe
square:
enter side:4
perimeter:16units and area:16square units
rectangle:
enter length:12
enter breadth:16
perimeter :56units and area:192square units
triangle:
enter a:3
enter b:3
enter c:3
perimeter :9units and area:3.89711square units
Process exited after 79.93 seconds with return value 0
Press any key to continue . . .

****

****

Assignment No-8

NAME – Chetan Gundurao Jagatap **Roll No -**
Class –B.Sc II **Date - / /**
Signature -

****

Q1) friend function.

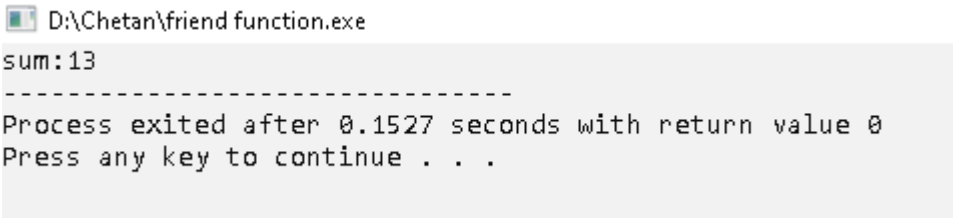
Q)Create two classes Celcius and Fahrenheit and define friend function to add and to compare two temperatures.

```
#include<iostream>
using namespace std;
class fahrenheit;
class celsius
{
        public:
                celsius():numA(12){}

                private:
                        int numA;
                        friend int add(class celsius,class fahrenheit);
};
 class fahrenheit
{
        public:
                fahrenheit():numB(1){}

                private:
                        int numB;
                        friend int add(class celsius,class fahrenheit);
};
int add(celsius objectA, fahrenheit objectB)
{
        return (objectA.numA+objectB.numB);
}
int main()
{
        celsius objectA;
        fahrenheit objectB;
        cout<<"sum:"<<add(objectA,objectB);
        return 0;
}
```

**********Output**********


D:\Chetan\friend function.exe
sum:13

Process exited after 0.1527 seconds with return value 0
Press any key to continue . . .

****