Presidency College
Reaccredited 'A+' by NAAC
NIRF Top 100 Ranked College
Kempapura, Hebbal, Bengaluru – 560024
www.presidencycollege.ac.in

3BCA C++ LAB MANUAL -2020

PART – B

During practical examination the External and Internal examiners may prepare exam question paper related to theory syllabus apart from Part-A. (A minimum of 8 Programs has to be prepared).

Note :

a) The candidate has to write two the programs One from Part-A and other from Part-B and execute one program as of External examiner choice.

b) A minimum of 8 Programs has to be done in Part-B and has to be maintained in the Practical Record.

c) Scheme of Evaluation is as follows:

Writing two programs - 10 Marks
Execution of one program - 10 Marks
Formatting the Output - 05 Marks
Viva - 05 Marks
Record - 05 Marks
Total - 35 Marks

Part-A

1. W.A.P to display student's avg of two better marks in 3 tests of subjects by creating student objects.
2. W.A.P to Deposit amount and withdraw options in bank transactions for SB and CB account and display the total balance using friend functions.
3. W.A.P to find the area of right angle, equilateral and scalene triangle using function overloading.
4. W.A.P to perform addition of real and complex numbers.
5. Write a C++ Program to compare two strings by overloading = = operator.
6. Write a C++ Program to perform addition of two matrices by overloading + operator.
7. W.A.P to create class STUDENT and SPORTs. with data members. Using inheritance display student details, percentage and sport details.
8. Write a C++ Program to sort elements using bubble sort technique applying function templates.
9. Write a C++ Program to calculate area and volume of various figures using function overriding.
10. Program to calculate the area and perimeter of rectangles using concept of inheritance.

PART - B

11. Write a program to prepare shopping list.
12.program to calculate area and circumference of circle using inline function
13. Program to find maximum of two numbers using friend function
14.program to add two distance variables.

15. C++ program to calculate volume of cube using constructor
16. Program to add two time variables.
17. Program to show returning current object, accessing member data of current object and returning values of object using this pointer
18. Program to display largest among two numbers using function templates.
19. Program to illustrate the concept of virtual function
20. Program to read & write file operation (convert lowercase to uppercase)

1. PART – A Define a STUDENT class with USN, Name, and Marks in 3 tests of a subject. Declare an array of 20 STUDENT objects. Using appropriate functions, find the average of the two better marks for each student. Print the USN, Name and the average marks of all the students.

```
#include <iostream.h>
#include <conio.h>
class STUDENT
{
    private: char usn[10], name[10];
                float marks1, marks2, marks3;
                float avg;

    public: void read()
                {       cout << "Enter the name and USN : " ;
                        cin >> name >> usn;
                    cout << "Enter marks1, marks2, marks3 : " ;
                    cin >> marks1 >> marks2 >> marks3;
                }

                void calculate()
                {
                   int smallest;
                if( (marks1 < marks2) && (marks1 < marks3) )
                          avg = (marks2+ marks3)/2.0;
                else if (marks2 < marks3)
                          avg = (marks1 + marks3)/2.0;
                          else avg = (marks1 + marks2)/2.0;
                }

                void display()
                {
                   cout <<"USN  "  << usn<<endl;
                   cout<<" Name " << name<<endl;
                   cout<<" Average Marks  "<<avg<<endl;

                }
};
```

```
void main()
{
	STUDENT s[10];
	clrscr();

	for(int i=0;i<2;i++)
	{
		s[i].read();
	}
	for(i=0;i<2;i++)
	{
		s[i].calculate();
		s[i].display();
	}
}
```

//PART - A PROGRAM2. Write a C++ Program to Deposit amount and withdraw options in bank transactions for saving and current account and display the total balance using friend functions.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class current;
class saving
{
	char name[15];
	int accno;
	float balance;
	public:
		void getdata()
		{
			cout<<"\n Savings account details \n";
			cout<<"\n Name";
			cin>>name;
			cout<<"\n Account number";
			 cin>>accno;
			cout<<"\n Balance ";
			 cin>>balance;
		}
	friend float totbalance(saving, current);
  };


class current
```

```
{
        char name[15];
        int accno;
        float balance;
        public:
                void getdata()
                {
                        cout<<"\n Current account details \n";
                        cout<<"\n Name ";
                        cin>>name;
                        cout<<"\n Account number ";
                        cin>>accno;
                        cout<<"\n Balance ";
                        cin>>balance;
                }
        friend float totbalance(saving,current);
 };

 float totbalance(saving s, current c)
 {
        cout<<"\n Saving's Account Banalce = Rs"<<s.balance<<endl;
        cout<<"\n Current Account Balance = Rs"<<c.balance<<endl;
        return(s.balance + c.balance);
 }

void main()
 {
        saving  s;
        current c;
        clrscr();
        s.getdata();
        c.getdata();
     cout<<endl<<"Total Balance          Rs"<<setw(6)<<totbalance(s,c)<<endl;
 }
```

3. PART-A Write a C++ Program to find the area of right angle, equilateral and scalene triangle using function overloading.

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
const float PI=3.1415;
```

```
float area(float b,float h)
{
float area1;
area1=0.5*b*h;
return area1;
}

float area(float s1)
{
float area2;
area2=sqrt(3)/4*(s1*s1);
return area2;
}

float area(float s1,float s2, float ang)
{
float area3;
area3=(s1 * s2 * sin((PI/180)*ang))/2;
return area3;
}

void main()
{
  float b,h,s1,s2,res1,res2,res3,ang;
  clrscr();
  cout<<"\nEnter value of base and height";
  cin>>b>>h;
  res1=area(b,h);
  cout<<"\nArea of right angle Triangle is "<<res1<<endl;

  cout<<"\n Input the value of the side of the equilateral triangle: ";
  cin>>s1;
  res2=area(s1);
  cout<<"\nArea of equilateral Triangle is "<<res2<<endl;


  cout<<"\n Input side1,side2,angle values for scalene triangle";
  cin>>s1>>s2>>ang;
  res3=area(s1,s2,ang);
  cout<<"\nArea of scalene Triangle is "<<res3<<endl;
}
```

/* 4. C++ program to create a class called COMPLEX and implement the following overloading functions ADD that return a complex number:
(i) ADD(a, s2) where a is an integer (real part) and s2 is a complex number
(ii) ADD(s1, s2) where s1 and s2 are complex numbers
*/

```
#include<iostream.h>
#include<conio.h>
class complex
{
  int r,i;
  public:
  void read()
  {
  cout<<"Enter real and imaginary\n";
  cin>>r>>i;;
  }

  void print()
  {
   cout<<r<<"+i"<<i<<endl;
  }

  friend complex add(int a,complex c);
  friend complex add(complex c1,complex c2);
};

complex add(int a,complex c)
{
  complex t;
  t.r = a + c.r;
  t.i = c.i;
  return t;
}


complex add(complex c1,complex c2)
{
  complex t;
  t.r = c1.r +  c2.r;
  t.i = c1.i +  c2.i;
```

```
  return t;

}
void main()
{
   int a=2;
   clrscr();
   complex s1,s2,s3;
   s1.read();
   cout<<"\ns1 : ";
   s1.print();
   s2=add(a,s1);
   cout<<"s2 : 2+s1\n";
   cout<<" : ";
   s2.print();
   s3=add(s1,s2);
   cout<<"s3=s1+s2\n";
   cout<<"s1 : ";
   s1.print();
   cout<<"s2 : ";
   s2.print();
   cout<<"s3 : ";
   s3.print();
   getch();
}
```

```
Enter real and imaginary
4
7

s1 : 4+i7
s2 : 2+s1
   : 6+i7
s3=s1+s2
s1 : 4+i7
s2 : 6+i7
s3 : 10+i14
```

// Program 5. PART-A Write a C++ Program to compare two strings by overloading = = operator.

```
#include<iostream.h>
#include<conio.h>
#include<string.h>

class string
{
```

```
 private:
            char str[40];
 public:
            void show()
            {
              cout<<str;
            }
            void getdata()
            {
              cin>>str;
            }
            friend int operator==(string, string);
};

int operator==(string t1, string t2)
{
          if(strcmp(t1.str,t2.str)==0)
          return 1;
          else
          return 0;
}

void main()
{
          clrscr();
          string s1,s2,s3;
          cout<<endl<<"Enter the first string:";
          s1.getdata();
          cout<<endl<<"Enter the second string:";
          s2.getdata();
          cout<<endl<<"String1:";s1.show();
          cout<<endl<<"String2:";s2.show();
          if(s1==s2)
          cout<<endl<<"Strings are equal.";
          else
          cout<<endl<<"Strings are unequal.";
          getch();
}
```

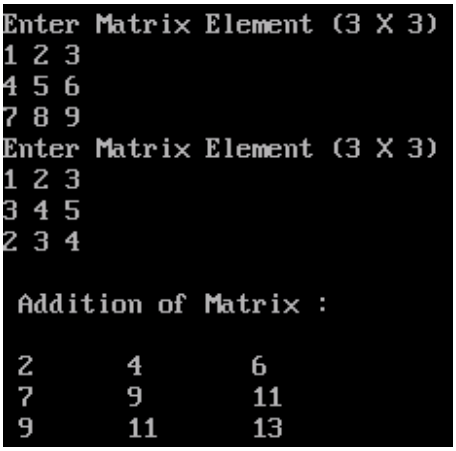
INPUT:
Enter the first string: bca
Enter the second string: bca
OUTPUT: Strings are equal

Enter the first string: bca
Enter the second string: bca1
Output: Strings are unequal

//Program 6.write c++ program to perform addition of 2 matrices using operator overloading.

```
#include<iostream>
class Matrix
{
      int a[3][3];
   public:
      void accept();
      void operator +(Matrix x);
};
void Matrix::accept()
{
      cout<<"Enter Matrix Element (3 X 3) \n";
      for(int i=0; i<3; i++)
      {
            for(int j=0; j<3; j++)
            {
                  cout<<" ";
                  cin>>a[i][j];
            }
      }
}
void Matrix::operator +(Matrix x)
{
      int mat[3][3];
      for(int i=0; i<3; i++)
      {
            for(int j=0; j<3; j++)
            {
                  mat[i][j]=a[i][j]+x.a[i][j];
            }
      }
      cout<<"\n Addition of Matrix : \n\n";
      for(int i=0; i<3; i++)
      {
            cout<<" ";
            for(int j=0; j<3; j++)
            {
                  cout<<mat[i][j]<<"\t";
            }
```

```
            cout<<"\n";
        }
}
void main()
{
        Matrix m,n;
        m.accept();       // Accepting Rows
        n.accept();      // Accepting Columns
        m+n;        // Addition of Two Matrices. Overloaded '+' Operator
 }
```



//7. Write a C++ Program to create a class called STUDENT with data members USN, Name and Age. Using inheritance, create the class MARKS containing data members for 3 subjects,percentage and create another class called SPORTS having data members name of the sport, achievements. Enter the data for at least 5 students. Display student details, percentage and sport details for all the students separately.

```
#include<iostream.h>
#include<iomanip.h>
#include<conio.h>
#include<stdio.h>
#include<string.h>
class STUDENT
{
        protected:
                int regno;
                int age;
                char name[20];

        public:
                void read()
                {
```

```
cout<<"Register Number,name and age ";
```

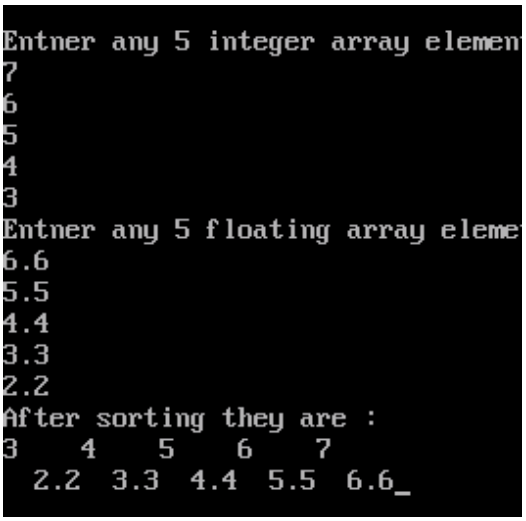
```
                cin>>regno;
                gets(name);
                cin>>age;
        }

};
class MARKS
{
    protected:
        int m1,m2,m3;
        float perc;
        char result[10];
    public:
        void readmarks()
        {
            cout<<"\nMarks in C++ := "; cin>>m1;
            cout<<"\nMarks in OS := "; cin>>m2;
            cout<<"\nMarks in FAM := "; cin>>m3;
            perc = (m1+m2+m3)/300.0*100;
            if(m1<40 || m2<40 || m3<40)
            strcpy(result,"fail");
            else
              strcpy(result,"pass");

        }

};
class SPORTS:public STUDENT , public MARKS
{
  private: char s_name[30];
           char s_ach[20];

  public: void readsports()
          {
            cout<<"enter sports name and achievement";
            gets(s_name);
            gets(s_ach);
          }
            void display()
            {

                cout<<regno<<setw(10);
                cout<<name<<setw(10);
                cout<<age<<setw(10);
```

```
                        cout<<perc<<setw(10);
                        cout<<result<<setw(10);
                        cout<<s_name<<setw(10);
                        cout<<s_ach;

                }
 };

void main()
{       int i;
          clrscr();
          SPORTS s[5];
          for(i=0;i<2;i++)
          {
            s[i].read();
            s[i].readmarks();
            s[i].readsports();
          }
          clrscr();
          cout<<"\n details of students \n"  ;
          cout<<" ---------------------------------------\n";
          cout<<"regno    name    age     percen result sportname  achievement\n";

          for(i=0;i<2;i++)
          {
            s[i].display();
            cout<<endl;

          }
}
```

//Program8. Write a C++ Program to sort elements using bubble sort technique applying function templates.

```
 #include<iostream.h>
#include<conio.h>
 #include<iomanip.h>
 #define n 5
template <class T>
void sort(T arr[])
{
          for (int i = 0; i < n-1; i++)
          {
                    for (int j = 0; j < n-i-1; j++)
                    {
```

```
                        if (arr[j] > arr[j+1])
                        {
                                T temp;
                                temp = arr[j];
                                arr[j] = arr[j+1];
                                arr[j+1] = temp;
                        }
                }
        }
}


void main()
{
        int x[5];
        float y[5];

        clrscr();
        cout<<"Entner any 5 integer array elements:"<<endl;
        for (int i = 0; i < 5; i++)
        {
                cin>>x[i];
        }
        cout<<"Entner any 5 floating array elements:"<<endl;
        for (i = 0; i < 5; i++)
        {
                cin>>y[i];
        }
        sort(x);
        sort(y);
        cout<<"After sorting they are :"<<endl;
        for ( i= 0; i < 5; i++)
        {
                cout<< x[i] << setw(5);
        }
        cout<<endl;
        for (i = 0; i < 5; i++)
        {
                cout<< y[i] << setw(5);
        }
        getch();
}
```



//9. Write a C++ Program to calculate area and volume of various figures using function overriding.

//Program to calculate area of geometrical figure

```
#include<iostream.h>
#include<conio.h>
#include<math.h>
#include<iomanip.h>

const float pi=3.1416;
class shape
{
        public:
                virtual void getdata()=0; //pure virtual fn
                virtual void area()=0;
};

class square: public shape
{
        private:
                float side;
        public:
                void getdata()
                {
                cout<<"\n ENTER SIDE VALUE OF SQUARE  \n";
                cin>>side;
                }

                void area()
                {
                   cout<<side*side;
                }
```

```
};

class triangle:public shape
{
        private:
                float a,b,c;
        public:
                void getdata()
                {
                cout<<"\n ENTER a,b,c VALUE OF TRIANGLE \n";
                cin>>a>>b>>c;
                }


        void area()
        {
        float s=(a+b+c)/2.0;
        cout<<sqrt(s*(s-a)*(s-b)*(s-c));
        }
};

class circle:public shape
{
        private:
                float r;
        public:
                void getdata()
                {
                cout<<"\n ENTER VALUE OF r OF CIRCLE \n";
                cin>>r;
                }

                void area()
                {
                cout<<pi*r*r;
                }
};

void main()
{
square s;
triangle t;
circle c;
```

```
clrscr();

s.getdata();
s.area();
t.getdata();
t.area();
c.getdata();
c.area();
getch();
}
```

// 10. Program to calculate the area and perimeter of rectangles using concept of inheritance.

```
// 10. Program to calculate the area and perimeter of rectangles
// using concept of inheritance.
//area of rectangle = l * w
//perimeter of rect = 2 * (l+w)

#include <iostream.h>
#include<conio.h>
#include<iomanip.h>
class RECT
{
   protected:
              float length, width;
   public:
             RECT()
              {
               length=0.0;
               width=0.0;
               }
};

class AREA : public RECT //AREA is derived from RECT
{
   public:
          void read()
          {
             cout<<"Enter length: ";
             cin>>length;
```

```
            cout<<"Enter width: ";
            cin>>width;
          }
           float calArea()
           {
              return length*width;
           }

};


class PERI : public RECT //PERI is derived from RECT
{
   public:

           void read()
          {
            cout<<"Enter length: ";
            cin>>length;
            cout<<"Enter width: ";
            cin>>width;
          }
            float calPeri()
           {

              return (2*(length+width));
           }
};

void main()
{
    clrscr();
    AREA a;
    PERI p;

    a.read();
    cout<<"Area of Rectangle is "<<setw(5) << a.calArea();
    cout<<endl;
    p.read();
    cout<<" \n Perimeter of Rectangle is"<<setw(5)<< p.calPeri();
    getch();

}
```

Part B:

Program 11. Write a program to prepare shopping list.

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
//#include<stdio.h>
const int size = 3;
class item
{
        private:
                char name[25];
                float rate, amount;
                int qty;
        public:
                void getdata()
                {
                        cout<<"Enter item name := "; cin>>name;
                        cout<<"Enter Quantity := "; cin>>qty;
                        cout<<"Enter item rate := "; cin>>rate;
                        cout<<" ---------------------------- "<<endl;
                }
                void printdata()
                {
                        cout<<setw(7)<<name;
                        cout<<setw(8)<<qty;
                        cout<<setw(8)<<rate;
                        amount = rate * qty;
                        cout<<setw(8)<<amount<<endl;
                }
};
void main()
{
        clrscr();
        item shop[size];
        for(int i=0; i<size; i++)
                shop[i].getdata();
        cout<<"\n\n\n Details of Shopping List\n\n";
        cout<<" Name Qty Rate Amount"<<endl;
        cout<<" ------------------------------------"<<endl;
        for(i=0; i<size; i++)
                shop[i].printdata();
        cout<<" ------------------------------------"<<endl;

}
```

12. PART –B program to calculate area and circumference of circle using inline function

```
#include<iostream.h>
#define pi  3.1412

inline float circum(float r)
{
        return (2 * pi * r);
}

inline float area (float r)
{
         return (pi * r * r);
}

void main()
{
          float rad;
         cout<<"enter radius "<<endl;
         cin>>rad;
        cout<<"circumference ="<<circum(rad);
         cout<<"\n Area is    ="<<area(rad);

}
```

//part- B 13. PROGRAM TO FIND MAXIMUM OF TWO NUMBERS USING FRIEND FUNCTION

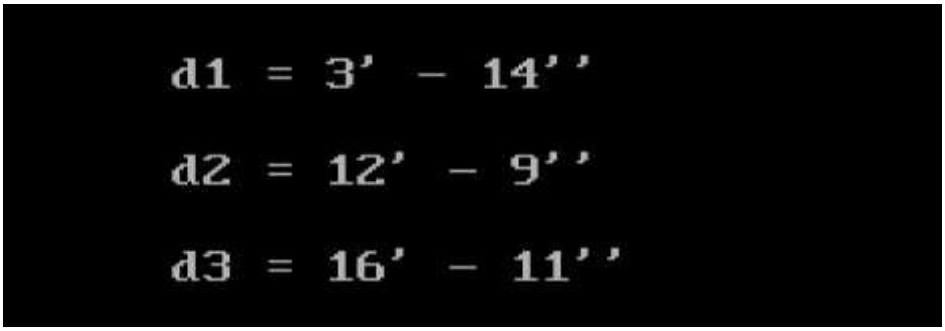
```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>

class max
{
  private:
          int x;
          int y;
  public:
          void getdata()
          {
            cout<<endl<<"Enter a number";
```

```
  cin>>x;
  cout<<endl<<"Enter another number";
  cin>>y;
 }

 void showdata()
 {
  cout<<endl<<"x is"<<x;
  cout<<endl<<"y is"<<y;
 }

 friend int large(max m);
};

int large(max m)
{
  if(m.x > m.y)
     return m.x;
  else
     return m.y;
}

void main()
{
  max m;
  clrscr();
  m.getdata();
  m.showdata();
  cout<<endl<<"Largest is"<<large(m);
}
```

//14.PROGRAM TO ADD TWO DISTANCE VARIABLES USING CONSTRUCTORS

```
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
#include<math.h>
class dist
{
        int feet;
        float inches;
        public:
```

```
        dist()    //default constructor
        {
                feet = 0; inches = 0.0;
        }

        dist(int ft, float in)     //parameterized
        {
                feet = ft; inches = in;
        }


        void showdist()
        {
                cout<<feet<<"'"<<" - "<<inches<<"'"<<"\n";
        }

        void sumdist(dist d1, dist d2)
        {

                int i1, i2;
                i1 = d1.inches;i2 = d2.inches;
                inches = (i1 + i2) % 12;
        feet = d1.feet + d2.feet + (d1.inches + d2.inches) / 12;
    }
};

void main()
{
        clrscr();
        dist d3;
        dist d1(3, 14) ;
        dist  d2(12,  9);  //paramterized
        d3.sumdist(d1,d2);
        cout<<"\n d1 = "; d1.showdist();
        cout<<"\n d2 = "; d2.showdist();
        cout<<"\n d3 = "; d3.showdist();
}
```



//15 PART –B C++ Program to calculate Volume of Cube using constructor

```
#include<iomanip.h>
#include<conio.h>

class cube
{
public: double side;

        cube(double side1)
        {
                cout << "\nparameterized constructor is called" << endl;
                side=side1;
        }

      cube()
        {
                cout<< "\nA default constructor is called " << endl;
        }

        ~cube()
        {
                cout << "\nDestructing " << side << endl;
        }

        double volume()
        {
                return(side*side*side);
        }
};

void main()
{
clrscr();
cube c1(2.34);
cube c2;
cout << "\nThe side of the cube is: " << c1.side << endl;
cout << "\nThe volume of the first cube is : ";
cout << c1.volume() <<endl;
cout << "\nEnter the length of the second cube : " ;
cin >> c2.side;
cout << "\nThe volume of second cube is : " << c2.volume() <<endl;
```

```
getch();
}
```

```
parameterized constructor is called
A default constructor is called
The side of the cube is: 2
The volume of the first cube is : 8
Enter the length of the second cube : 3
The volume of second cube is : 27
Destructing 3
Destructing 2
```

```
//16. PROGRAM TO ADD TWO TIME VARIABLES.
#include<iostream.h>
#include<conio.h>
#include<iomanip.h>
class CLOCK
{
        int hr, min;
        public:
                void gettime(int hh, int mm)
                {
                        hr = hh;
                        min = mm;
                }

                void showtime()
                {
                        cout<<hr<<" hours and "<<min<<" minutes"<<endl;
                }
                CLOCK operator + (CLOCK t )
                {
                        int i1;
                        CLOCK temp;
                        temp.min = (min + t.min) % 60;
                        i1 = (hr + t.hr);
                        if((min+t.min)>=60)
                                i1++;
                        temp.hr = i1;
                        return temp;
                }
```

```
};

void main()
{
        clrscr();
        CLOCK t1, t2, t3;
        t1.gettime(1, 25);
        t2.gettime(3, 50);
        t3 = t1 + t2;
        cout<<endl<<"Time 1        :"; t1.showtime();
        cout<<endl<<"Time 2        :"; t2.showtime();
        cout<<endl<<"Total Duration :"; t3.showtime();
        getch();
}
```

Output

1 hours and 25 minutes
3 hours and 50 minutes
5 hours and 15 minutes

//17. PROGRAM TO SHOW RETURNING CURRENT OBJECT, ACCESSING MEMBER DATA OF CURRENT OBJECT AND RETURNING VALUES OF OBJECT USING THIS POINTER

```
#include <iostream.h>
#include <conio.h>
class Box
{
  private:
    double  l;
    double b;
    double h;
  public:
    // Constructor definition
    Box(double l, double b, double h)
    {
        cout <<"Constructor called." << endl;
        this->l = l;
        this->b = b;
        this->h = h;
    }

    double Volume()
    {
        return l*b*h;
    }


    int compare(Box box)
    {
        return this->Volume() > box.Volume();
```

```
    }

};

void main(void)
{
  Box Box2(2.0, 2.0, 2.0);    // Declare box1
  Box Box1(3.0, 3.0, 3.0);    // Declare box2

  clrscr();
  if(Box1.compare(Box2))
  {
    cout << "Box2 is smaller" <<endl;
  }
  else
  {
    cout << "Box1 is smaller" <<endl;
  }
  getch();
  }
```

//program 18 part-b Program to display largest among two numbers using function templates.

```
#include<iostream.h>
#include<conio.h>

template <class T>
T LARGE(T n1, T n2)
{
        return (n1 > n2) ? n1 : n2;
}

void main()
{
        int i1, i2;
        float f1, f2;
        char c1, c2;
        clrscr();
        cout << "Enter two integers:\n";
        cin >> i1 >> i2;
        cout << LARGE(i1, i2) <<" is larger." << endl;

        cout << "\nEnter two floating-point numbers:\n";
        cin >> f1 >> f2;
        cout << LARGE(f1, f2) <<" is larger." << endl;

        cout << "\nEnter two characters:\n";
        cin >> c1 >> c2;
        cout << LARGE(c1, c2) << " has larger ASCII value.";

        getch();
```

}

```
Enter two integers:
2 3
3 is larger.

Enter two floating-point numbers:
2.3 6.7
6.7 is larger.

Enter two characters:
a z
z has larger ASCII value.
```

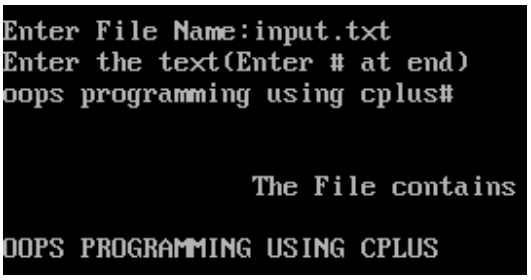
//19. PROGRAM TO ILLUSTRATE THE CONCEPT OF VIRTUAL FUNCTION

```
#include<iostream.h>
#include<conio.h>
class base
{
   public:
     virtual void show()
     {
            cout<<"\n Base class show:";
     }
     void display()
     {
           cout<<"\n Base class display:" ;
     }
};
class derive:public base
{
  public:
     void display()
     {
           cout<<"\n Drive class display:";
     }
     void show()
     {
           cout<<"\n Drive class show:";
     }
};
void main()
{
  clrscr();
  base obj1;
  base *p;
  cout<<"\n\t P points to base:\n"  ;
```

```
  p=&obj1;
  p->display();
  p->show();
  cout<<"\n\n\t P points to derive:\n";
  derive obj2;
  p=&obj2;
  p->display();
  p->show();
  getch();
}
```

//20. Program to Read & Write File Operation (Convert lowercase to uppercase)

```
#include<fstream.h>
#include<stdio.h>
#include<ctype.h>
#include<string.h>
#include<iostream.h>
#include<conio.h>
void main()
{
char c,u;
char fname[10];
clrscr();
ofstream out;
cout<<"Enter File Name:";
cin>>fname;
out.open(fname);
cout<<"Enter the text(Enter # at end)\n"; //write contents to file
while((c=getchar())!='#')
{
u=c-32;
out<<u;
}
out.close();
ifstream in(fname); //read the contents of file
cout<<"\n\n\t\tThe File contains\n\n";
while(in.eof()==0)
{
in.get(c);
cout<<c;
}
getch();

}
```


Enter File Name:input.txt
Enter the text(Enter # at end)
oops programming using cplus#
The File contains
OOPS PROGRAMMING USING CPLUS