OOPS LAB MANUAL

1.) Write a C++ program to implement stack using following functions:

Push

Pop

Display

```
#include<iostream>
using namespace std;
#define size 5
class stack
{
      int stck[size];
      int tos;
      public:
      void init();
      void push();
      int pop();
      void display();
};
void stack::init()
{
tos=0;
}
void stack::push()
{
        int i;
      if(tos==size)
      {
            cout<<"stack overflowl \n";
            return;
      }
      cout<<"enter element \n";
      cin>>i;
      stck[tos]=i;
      tos++;
}
int stack::pop()
{
      if(tos==0)
      {
            cout<<"stack underflow \n";
```

```
            return 1;
      }
      tos--;
      cout<<”popped element is”<<stck[tos];
      return 0;
}
void stack::display()
{
      if(tos==0)
      {
            cout<<"stack empty \n";
            return;
      }
      for(int i=0;i<tos;i++)
      {
            cout<<stck[i]<<"\n";
      }
}
int main()
{
      int ch,i;
            stack s;
            s.init();
      while(1)
      {
            cout<<"1.push 2.pop 3.display \n";
            cin>>ch;
            switch(ch)
            {
                  case 1:s.push();
                        break;
                  case 2:s.pop();
                        break;
                  case 3:s.display();
                        break;
                  default: return 0;
            }
      }
}
```

2(a) Write a C++ program to read the data of n employee and compute net salary of each employee using pointer. Given that an employee class contains following :-

Data members: Employee no, Employee name, Basic salary, DA, IT, Net salary, gross salary

Member functions: To read data, to calculate net salary and to print data [DA = 52% of basic salary, IT = 30% of gross salary, Gross salary = DA + Basic, Net salary = DA + Basic – IT].

```
#include<iostream>
using namespace std;
class employee
{
      int num, basic;
      long da,it,netsal,gsal;
      char name[20];
      public:
      void read();
      void disp();
      void calc();
};

void employee::read()
{
      cout<<"enter employee id,name & basic salary \n";
      cin>>num>>name>>basic;
}
void employee::calc()
{
      da=(0.52)*basic;
      gsal=da+basic;
      it=(0.3)*gsal;
      netsal=basic+da-it;
}
void employee::disp()
{
      cout<<num<<"\t"<<name<<"\t\t\t"<<basic<<"\t\t"<<da<<"\t\t"<<it<<"
          \t\t" <<gsal<<"\t"<<netsal<<"\n";
}
int main()
{
      int n;
      cout<<"enter no. of employee \n";
      cin>>n;
      employee e[n],*p;
      for(int i=0;i<n;i++)
      {
            p=&e[i];
            p->read();
            p->calc();
      }
```

```
        cout<<"sl no. eid\t    name \t\t basic salary\t \t DA\t\tgross
              salary\t  income tax\t Net salary\n";
              for(int i=0;i<n;i++)
              {
                    p=&e[i];
                    p->disp();
              }
              return 0;
           }
```

---

2)(b) Write a c++ program to find the largest of three numbers using inline function and defaullt argument concept.

```
#include<iostream>
using namespace std;
inline float lar(float a, float b ,float c=50 )
{
          return ((a>b&&a>c)?a:b>c?b:c);
}
int main()
{
          float x,y,z;
          cout<<"\nEnter three numbers:\n";
          cin>>x>>y>>z;
          cout<<"Largest= "<<lar(x,y,z);
          cout<<"\nTaking 3rd number as 50\n";
          cout<<"\nEnter two numbers:\n";
          cin>>x>>y;
          cout<<"\nLargest using default value= "<<lar(x,y)<<"\n";
          return 0;
}
```

3.) Write a C++ program to define a student class with data members usn, name and marks of 3 subjects. And member functions to read, display, and to calculate average of best 2 marks. Also find who is the topper among “n” no. of students.

```
#include<iostream>
using namespace std;
class stud
{
          char usn[20],name[30];
          float marks[3];
          int i;
public:
```

```
            float avg;
            void read();
            void calc(int);
            void disp();
};
void stud::read()
{
            cout<<"\nEnter USN no.: ";
            cin>>usn;
            cout<<"\nEnter name: ";
            cin>>name;
            for(i=0;i<3;i++)
            {
            cout<<"\nEnter the marks of subject "<<i+1<<": ";
            cin>>marks[i];
            }
}
void stud::calc(int n)
{
            int sum=0,min=marks[0];
            for(i=0;i<n;i++)
            {
            sum+=marks[i];
            if(min>=marks[i])
            min=marks[i];
            }
            avg=float(sum-min)/2;
}
void stud::disp()
{
            cout<<usn<<"\t"<<name<<"\t";
            for(i=0;i<3;i++)
            cout<<marks[i]<<"\t";
            cout<<avg<<"\n";
}
int main()
{
            int n,topper,i,z;
            float max=0;
            cout<<"\nEnter the number of students: ";
            cin>>n;
            stud s[n];
            for(i=0;i<n;i++)
            {
            s[i].read();
            s[i].calc(n);
```

```
            }
            cout<<"USN\tName\t";


            for(i=0;i<3;i++)
            {
            cout<<"Marks "<<i+1<<"\t";
            }
            cout<<"Avg.\n";
            for(i=0;i<n;i++)
            s[i].disp();
            for(i=0;i<n;i++)
            {
            if(max<s[i].avg)
            {
            max=s[i].avg;
            topper=i;
            }
            }
            for(i=0;i<n;i++)
            {
            if(s[topper].avg==s[i].avg)
            {
            cout<<"\nThe toppers is student "<<i+1<<": ";
            cout<<"\nDetails:-\n";
            s[i].disp();
            }
            }
    }
```

4.) Write a C++ program to create a class called “Complex” & Implement. Following overloading functions. ADD that returns a complex no.
ADD (a, s2) - where s2 is a complex no and ‘a’ is an integer.
ADD (s1, s2) - where s1 & s2 are complex no.

```
#include<iostream>
using namespace std;
class complex
{
            float real,img;
public:
            void read();
            void print();
            friend complex ADD(int,complex);
```

```
            friend     complex ADD(complex,complex);
};
void complex::read()
{
            cin>>real>>img;
}
void complex::print()
{
            cout<<real<<"+i"<<img;
}
complex ADD(int a,complex s2)
{
            complex c;
            c.real=a+s2.real;
            c.img=s2.img;
            return c;
}

complex ADD(complex s1,complex s2)
{
            complex c;
            c.real=s1.real+s2.real;
            c.img=s1.img+s2.img;
            return c;
      }


int main()
{
            complex c1,c2,c3;
            int a;
            cout<<"Enter Real and Imaginary components of c1 :\n";
            c1.read();
            cout<<"Enter Real and Imaginary components of c2 :\n";
            c2.read();
            cout<<"Enter the integer to be added to c2: \n";
            cin>>a;
            c3=ADD(a,c2);
            cout<<"After adding "<<a<<" to ";
            c2.print();
            cout<<"\nresult= ";
            c3.print();
            c3=ADD(c1,c2);
            cout<<"\nAfter adding:\n";
            c1.print();
            cout<<" to ";
```

```
            c2.print();
            cout<<"\nresult= ";
            c3.print();
            cout<<"\n";
            return 0;
    }
```

---

5) (a) Write a C++ program to implement locking and unlocking using static member functions.

```
#include<iostream>
using namespace std;
class resource
{
    static int res;
    public:
    static int getr();
    void free_res()
    {
       res=0;
    }
};
int resource::res;
int resource::getr()
{
    if(res)
       return 0;
    else
    {
       res=1;
       return 1;
    }
}
int main()
{
    resource a,b;
    if(resource::getr())
    cout<<"Resource under use, object a is using \n";
    if(!resource::getr())
    cout<<"Resource busy, object b access denied \n";
    a.free_res();
    if(resource::getr())
    cout<<"Resource can now be used by Object b \n";
    return 0;
}
```

---

5) (b) Write a C++ program to implement a class which accepts date in different formats (using constructor overloading).

```
#include<iostream>
#include<cstdio>
using namespace std;
class dates
{
    int dd,mm,yy;
    public:
    dates()
    {
    cout<<"Default  date: 1/12/2012 \n";
    }
    dates(char *d)
    {
       scanf("%d%d%d",&mm,&dd,&yy);
    }
    dates(int m,int d,int y)
    {
       dd=d;
       mm=m;
       yy=y;
    }
    void sd()
    {
    cout<<"Date is: \n";
    cout<<mm<<"/"<<dd<<"/"<<yy<<"\n";
    }
};

int main()
{
    int m,d,y;
    cout<<"Enter month,date and year \n";
    cin>>m>>d>>y;
    cout<<"Enter month,date & year in a single line \n";
    dates b("m,d,y"),c(m,d,y),e;
    b.sd();
    c.sd();
    return 0;

}
```

6.) Write a C++ program to create class called list with member functions to insert an element from front as well as to delete element from front of list. Demonstrate all functions by creating list object.

```
#include<iostream>
```

```
#include<new>
using namespace std;
struct nod
{
      int info;
      struct nod*next;
};

typedef struct nod node;
class list
{
      node *f;
      public:
            list()
            {
            f=NULL;
            }
            void ins(int num)
            {
                  node *p=new node;
                  p->info=num;
                  p->next=f;
                  f=p;
            }
            void del()
            {
                  node *temp=f;
                  if(f==NULL)
                  cout<<"\nNo elements to delete.\n";
                  else
                  {
                        cout<<"\n The deleted elements is :\n"<<f->info;
                        f=f->next;
                        delete temp;
                        cout<<"\n Deletion successfull \n";
                  }
                  return;
            }
            void disp()
            {
                  node *temp=f;
                  if(f==NULL)
                  cout<<"\n List is empty \n";
                  else
                  {
                        cout<<"\n Elements in the list are: ";
                        while(temp!=NULL)
                        {
                              cout<<" "<<temp->info;
                              temp=temp->next;
                        }
                  }
            }
};
```

```
int main()
{
      int num,ch=1;
      list ob;
      cout<<"\n!!!!!!!!!!! LINEAR LINK LIST !!!!!!!!!!!!!!!!\n";
      cout<<"\n1] Insert\n2] Delete\n3] Exit";
      while(ch)
      {
            cout<<"\nEnter your choice \n";
            cin>>ch;
            switch(ch)
            {
                  case 1: cout<<"\n Enter no. to be insrted\n";
                        cin>>num;
                        ob.ins(num);
                        ob.disp();
                        break;
                  case 2: ob.del();
                        ob.disp();
                        break;
                  case 3: return 0;
                  default:cout<<"Invalid choice \n";
                        break;
            }
      }
}
```

7.) Write a C++ program to create a class called student with the dada members. USN, Name and Age using inheritance. Create classes UG student and PG student having field as semester, fees and stipend. Enter the data for at least 5 students and find semester wise average, age for UG and PG students respectively.

```
#include<iostream>
using namespace std;
class student
{
            int reg,age;
            char name[50];
      public:
            void getsdata()
            {
            cout<<"\nEnter name of student: ";
            cin>>name;
            cout<<"\nEnter regestration no.: ";
            cin>>reg;
            cout<<"\nEnter age: ";
```

```
            cin>>age;
            }
            int giveage()
            {
            return age;
            }
};
class ugstudent:public student
{
            int sem;
            float fee,sti;
public:
            void getugdata()
            {
            getsdata();
            cout<<"\nEnter Sem: ";
            cin>>sem;
            cout<<"\nEnter fee: ";
            cin>>fee;
            cout<<"\nEnter stipend: ";
            cin>>sti;
            }
            int givsem()
            {
return sem;
            }
};

class pgstudent:public student
{
            int sem;
            float fee,sti;
public:
            void getpgdata()
            {
            getsdata();
            cout<<"\nEnter Sem: ";
            cin>>sem;
            cout<<"\nEnter fee: ";
            cin>>fee;
            cout<<"\nEnter stipend: ";
            cin>>sti;
            }
            int givsem()
            {
            return sem;
```

```
        }
};
int main()
{
            int m,n,i,s;
            cout<<"\nEnter the no. of UG student: ";
            cin>>m;
            ugstudent ug[m];
            for(i=0;i<n;i++)
            ug[i].getugdata();
            for(s=1;s<=8;s++)
            {
            int flag=0,count=0,sum=0;
            for(i=0;i<n;i++)
            {
            if(ug[i].givesem()==s)
            {
            sum=sum+ug[i].giveage();
            flag=1;
            count++;
            }
            }
            if(flag==1)
            {
            cout<<s<<" sem avg. age is: "<<sum/cou;
            }
            }
            cout<<"\nEnter the no. of PG student: ";
            cin>>n;
            ugstudent pg[n];
            for(i=0;i<n;i++)
            pg[i].getpgdata();
            for(s=1;s<=8;s++)
            {
            int flag=0,count=0,sum=0;
            for(i=0;i<n;i++)
            {
            if(pg[i].givesem()==s)
            {
            sum=sum+pg[i].giveage();
            flag=1;
            count++;
            }
            }
            if(flag==1)
            {
```

```
            cout<<s<<" sem avg. age is: "<<sum/cou;
            }
            }
      return 0;
}
```

---

8) (a) Write a C++ program for exception handling. Create a user defined exception classes for divide by zero and negative number input separately.

```
#include<iostream>
using namespace std;
class divide
{
            int a,b;
public:
            void compute()
            {
            cout<<"\nEnter two integers \n";
            cin>>a>>b;
            try
            {
            if(b!=0)
            cout<<a/b<<"\n";
            else
            throw b;
            }
            catch(...)
            {
            cout<<"\nDivision by zero(error) \n";
}
            }
};
class neg
{
            int a;
public:
            void scan()
            {
            cout<<"\nEnter age \n";
            cin>>a;
            try
            {
            if(a>=0)
```

```
            cout<<"\nAge of the person is "<<a<<" \n";
            else
            throw a;
            }
            catch(...)
            {
            cout<<"\nPlease enter positive value for age \n";
            }
            }
};
int main()
{
            int c;
            divide d;
            neg n;
            while(1)
            {
            cout<<"1.divide integers 2.enter age 3.exit \n";
            cin>>c;
            switch(c)
            {
            case 1:d.compute();
            break;
            case 2:n.scan();
            break;
            default:return 0;
            }
            }
            return 0;
}
```

8(b) Write a C++ program for sorting names using file handling.

```
#include<iostream>
#include<fstream>
#include<cstring>
#include<cstdlib>
using namespace std;
int main()
{
      char temp[20],name[20][20];
      int i,j,n;
      FILE *f;
      f=fopen("sort.txt","w");
      cout<<"\nEnter no. of names\n";
```

```
        cin>>n;
        cout<<"\nEnter "<<n<<" names\n";
        for(i=0;i<n;i++)
        {
              cin>>name[i];
              fprintf(f,"%s",name[i]);
        }
        fclose(f);
        f=fopen("sort.txt","r");
        if(f==NULL)
        {
              cout<<"\nFile doesnt exists\n";
              return 0;
        }
        while(!feof(f))
        {
              fscanf(f,"%s",name[i]);
              i++;
        }
        n=i-1;
        cout<<"\nNames before sorting :\n";
        for(i=0;i<n;i++)
              cout<<name[i]<<"  "<<endl;
        cout<<"\nNames after sorting :\n";
        for(i=0;i<n-1;i++)
        {
              for(j=0;j<n-i-1;j++)
              {
                    if(strcmp(name[j],name[j+1])>0)
                    {
                          strcpy(temp,name[j]);
                          strcpy(name[j],name[j+1]);
                          strcpy(name[j+1],temp);
                    }
              }
        }
        for(i=0;i<n;i++)
              cout<<name[i]<<"  "<<endl;
        return 0;
  }
```

9.) Write a C++ program to create a class string with default parameterized and copy constructor. Implement the following by overloading operation :- strcpy(),strcmp(),strcat().

```
#include<iostream>
#include<string>
using namespace std;
class strings
{
        char *str;
```

```
public:
            strings(){}
            strings(char *s)
            {
            str=s;
            }

            void operator=(strings);
            int operator==(strings);
            strings operator+(strings);
            friend ostream& operator<<(ostream&,strings);
};
void strings::operator=(strings s2)
{
            int i=0;
            while(s2.str[i]!='\0')
            {
            str[i]=s2.str[i];
            i++;
            }
            str[i]='\0';
}
strings strings::operator+(strings s2)
{
            int i=0,j=0;
            while(str[i]!='\0')
            i++;
            while(s2.str[j]!='\0')
            {
            str[i]=s2.str[j];
            i++;
            j++;
            }
            str[i]='\0';
            return *this;
}
int strings::operator==(strings s2)
{
            int i=0;

            while(str[i]!='\0'||s2.str[i]!='\0')
            {
            if(str[i]!=s2.str[i])
            return(str[i]-s2.str[i]);
            i++;
            }
```

```
            return 0;
    }
    ostream& operator<<(ostream& os,strings s)
    {
            cout<<s.str;
            return os;
    }
    int main()
    {
            char str1[80],str2[80];
            cout<<"\nEnter 2 strings:\n";
            cin>>str1>>str2;
            strings s1(str1),s2(str2);
            strings t1(s1),t2=s2;
            if((s1==s2)==0)
            cout<<"\nStrings are equal\n";
            else if((s1==s2)>0)
            cout<<"\nString 1 is greater\n";
            else
            cout<<"\nString 2 is greater\n";
            strings s3=s1+s2;
cout<<"\nAfter concatenation\n1st string= "<<s3<<"\n2nd string= "<<s2;
t1=t2;
   cout<<"\nAfter copying\n1st string= "<<t1<<"\n2nd string= "<<t2<<"\n";
            return 0;
    }
```

10.) Write a C++ program to implement following operator overloading concept using complex number. + ,- ,+ +, -, = =

```
    #include<iostream>
    using namespace std;
    class comp
    {
            int r,i;
    public:
            void scan()
            {
                  cout<<"Enter real & imaginary part of complex no. \n";
                  cin>>r>>i;
            }
            void sd()
            {
```

```
                if(i<0)
                     cout<<"complex no. is "<<r<<i<<"i \n";
                else
                     cout<<"complex no. is "<<r<<"+"<<i<<"i \n";
         }
           comp operator+(comp x);
           comp operator-(comp x);
           friend comp operator++(comp &a);
           friend comp operator+(int a,comp &x);
           friend comp operator+(comp &x,int b);
           comp operator=(comp p);
};
comp comp::operator+(comp x)
{
           comp a;
           a.r=x.r+r;
           a.i=x.i+i;
           return a;
}
comp comp::operator-(comp x)
{
           comp b;
           b.r=r-x.r;
           b.i=i-x.i;
           return b;
}
comp operator++(comp &a)
{
           a.r++;
           return a;
}
comp operator+(int a,comp &x)
{
           //    comp p;
           x.r=a+x.r;
           //    return p;
}
comp operator+(comp &x,int b)
{
           x.r=x.r+b;
//    return q;
}
comp comp::operator=(comp p)
{
           r=p.r;
           i=p.i;
```

```
            return *this;
    }
    int main()
    {
            comp a,b,c;
            int x;
            a.scan();
            b.scan();
            a.sd();
            b.sd();
            cout<<"Adding two complex no's \n";
            (a+b).sd();
            cout<<"Subtracting two complex no's \n";
            (a-b).sd();
            c=a;
            cout<<"Assign obj a to obj c \n";
            c.sd();
            operator++(a);
            cout<<"After incrementing real part of a \n";
            a.sd();
            cout<<"enter an integer \n";
            cin>>x;
            cout<<"Adding integer to complex no. \n";
            operator+(x,a);
            a.sd();
            operator+(c,x);
            c.sd();
            return 0;
    }
```

11.) Write a C++ program to sort integers and floating point numbers using template.

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

  template<class x > void sort(x *a,int size)
    {
      x t;
      for(int i=0;i<size;i++)
      for(int j=(size-1);j>=i;j--)
          {
            if(a[j-1]>a[j])
                {
                  t=a[j-1];
                  a[j-1]=a[j];
                  a[j]=t;
```

```
                            }
                        }
          }
  int main()
   {
      int i,n;
            cout<<"enter integer array size\n";
            cin>>n;
            int p[n];
            cout<<"Enter "<<n<<" integer array elements\n";
            for(i=0;i<n;i++)
            cin>>p[i];
      cout<<"Actual integer array before sorting is:\n";
            for(i=0;i<n;i++)
             cout<<p[i]<<" ";
            cout<<"\n";
        sort(p,n);
       cout<<"Sorted Integer array is:\n";

 for(i=0;i<n;i++)
        cout<<p[i]<<" ";
        cout<<"\n";
char q[n];
   cout<<"Enter "<<n<<" character  array elements\n";
     for(i=0;i<n;i++)
        {cin>>q[i];}
     cout<<"Actual character  array before sorting is:\n";
            for(i=0;i<n;i++)
        cout<<q[i]<<" ";
        cout<<"\n";
        sort(q,n);
    cout<<"Sorted character  array is:\n";
      for(i=0;i<n;i++)
        cout<<q[i]<<" ";
        cout<<"\n";
 float r[n];
  cout<<"Enter "<<n<<" float array elements\n";
     for(i=0;i<n;i++)
        cin>>r[i];
     cout<<"Actual float array before sorting is:\n";
            for(i=0;i<n;i++)
        cout<<r[i]<<" ";
        cout<<"\n";
        sort(r,n);
    cout<<"Sorted float array is:\n";
      for(i=0;i<n;i++)
```

```
        cout<<r[i]<<" ";
        cout<<"\n";
return 0;
  }
```

12.) Write a C++ program to swap integer, floating point numbers and characters by using call by value and call by reference.

```
#include<iostream>
using namespace std;
template<class t>void swv(t a,t b)
{
    cout<<"\nBefore swap(in function) a = "<<a<<" & b = "<<b<<"\n";
    t x;
    x=a;
    a=b;
    b=x;
    cout<<"\nAfter swap(in function) a = "<<a<<" & b = "<<b<<"\n";
}
template<class t,class w>void swp(t *a,w *b)
{
    cout<<"\nBefore swap (in function) a = "<<*a<<" & b = "<<*b<<"\n";
    t y;
    y=*a;
    *a=*b;
    *b=y;
    cout<<"\nAfter swap (in function) a = "<<*a<<" & b "<<*b<<"\n";
}
int main()
{
    int a,b;
    float x,y;
    char m,n;
    cout<<"\nEnter two integers\n";
    cin>>a>>b;
    swv(a,b);
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<a<<"\nb: "<<b;
    swp(&a,&b);
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<a<<"\nb: "<<b;
    cout<<"\nEnter two float nos.\n";
    cin>>x>>y;
    swv(x,y);
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<x<<"\nb: "<<y;
    swp(&x,&y);
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<x<<"\nb: "<<y;
    cout<<"\nEnter two char \n";
    cin>>m>>n;
    swv(m,n);
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<m<<"\nb: "<<n;
    swp(&m,&n);
```

```
    cout<<"\nAfter swap a & b(in main)\n"<<"a: "<<m<<"\nb: "<<n;
    return 0;
}
```

---

13) (a) Write a C++ program to create a file to store Account holder name, account number and Balance for given number of customers. Also retrieve the values from the file and print it on the standard output.

```
#include<iostream>
#include<fstream>
using namespace std;
struct bank
{
    char name[20];
    double bal;
    long acc;
};
int main()
{
    int n,i;
    cout<<"Enter no. of entries \n";
    cin>>n;
    struct bank b[n];
    ofstream op("bank.txt",ios::out);
    if(!op)
    {
       cout<<"Cannot open file \n";
       return 1;
    }
    for(i=0;i<n;i++)
    {
       cout<<"enter a/c no.,name & bal of customer : "<<i+1<<"\n";
       cin>>b[i].acc>>b[i].name>>b[i].bal;
       op.write((char *)&b[i],sizeof(struct bank));
    }
    op.close();
    ifstream ip("bank.txt",ios::in);
    if(!ip)
    {
       cout<<"File doesn't exists \n";
       return 1;
    }
    for(i=0;i<n;i++)
    {
       if(ip)
       {
            ip.read((char *)&b[i],sizeof(struct bank));
            cout<<"A/c no : "<<b[i].acc<<"\n";
            cout<<"Name : "<<b[i].name<<"\n";
            cout<<"Balance (in INR) : "<<b[i].bal<<"\n";
       }
    }
```

```
        ip.close();
        return 0;
    }
```

---

13 (b).Write a C++ program to convert dollar to rupees, euro to rupees and pound to rupees using pure virtual functions.

1 dollar = 54.3 Rs
1 pound = 81.1Rs
1 euro = 70. Rs

```
#include<iostream>
using namespace std;
class rs
{
    public:
    float rs;
    virtual void conv()=0;
    void disp()
    {
        cout<<" is eqvivalent to "<<rs<<" INR \n";
    }
};

class doll:public rs
{
    float dol;
    public:
    void conv()
    {
        cout<<"Enter currncy in dollar \n";
        cin>>dol;
        rs=54.3*dol;
        cout<<" "<<dol<<" in dollar ";
        disp();
    }
};

class euro:public rs
{
    float er;
    public:
    void conv()
    {
        cout<<"Enter currency in Euro \n";
        cin>>er;
        rs=70.2*er;
        cout<<" "<<er<<" in euro ";
        disp();
    }
};
class pd:public rs
{
```

```
    float pnd;
    public:
    void conv()
    {
       cout<<"Enter currency in pound \n";
       cin>>pnd;
       rs=81.1*pnd;
       cout<<" "<<pnd<<" in pound ";
       disp();
    }
};
int main()
{
    int c;
    doll d;
    euro e;
               pd p;
    cout<<"\t\t\t Currency conversion \n";
    while(1)
    {
              cout<<"1.$ to Rs.    2.Euro to Rs.  3.Pound to Rs. 4.Exit
\n";
         cin>>c;
     switch(c)
    {
             case 1:d.conv();
        break;
             case 2:e.conv();
            break;
              case 3:p.conv();
       break;
    default:return 0;
    }
    }
    return 0;
}
```

---

14. (a) Write a C++ program to perform ( ) overloading.

```
#include<iostream>
using namespace std;
class  loc
 {
      int longitude, latitude;
public:
     loc ( )  {}
     loc(int lg, int lt)
      {
     longitude = lg;
      latitude = lt;
     }
void show ()
```

```
{
      cout<< longitude<<" ";
      cout<< latitude<<"\n ";
}
        loc operator+ (loc op2);
        loc operator()( int i, int j);
};
// overload () for loc
loc loc::operator()(int i, int j)
{
      longitude = i;
      latitude = j;
      return *this;
}
// overload + for loc
loc loc::operator+(loc op2)
{
      loc temp;
      temp.longitude= op2.longitude+longitude;
      temp.latitude= op2.latitude+latitude;
      return temp;
}
int main()
{
      loc ob1(10,20), ob2(1,1);
      ob1.show();
      ob1(7,8);
      ob1.show();
      ob1=ob2+ob1(10,10);
      ob1.show();
      return 0;
}
```

---

14. (b) Write a C++ program to perform -> overloading.

```
#include<iostream>
using namespace std;
class  myclass{
public:
 int i;
myclass *operator->() {return this;}
};


int main()
{
myclass ob;
ob->i=10;
cout<<ob.i <<" "<<ob->i;
return 0;
}
```

15.) Sorting using Genric classes

```
#include<iostream>
#include<cstdlib>
using namespace std;
const int SIZE=10;
template<class atype>class sort
{
public:  atype a[SIZE];
           atype &operator[](int i)
           {
           if(i<0||i>SIZE-1)
           {
           cout<<"\nIndex value of ";
           cout<<i<<" is out-of-bonds.\n";
           exit(1);
           }
           return a[i];
           }
sort(){}
void read(int z)
{
           for(int i=0;i<z;i++)
cin>>a[i];
}


void disp(int z)
{
           for(int i=0;i<z;i++)
           cout<<a[i]<<" ";
           cout<<"\n";
}
template<class X> void bubble(X *items,int count)
{
           register int a,b;
           X t;
           for(a=1;a<count;a++)
           {
           for(b=count-1;b>=a;b--)
           {
           if(items[b-1]>items[b])
           {
           t=items[b-1];
```

```
            items[b-1]=items[b];
            items[b]=t;
            }
            }
            }
    }
    };
    int main()
    {
            sort <int> intob;
            sort <double> doubleob;
            int i,m,n;
            cout<<"\nEnter the size of integer array: -\n";
            cin>>n;
            cout<<"\nEnter the size of double array: -\n";
            cin>>m;
            intob[n-1]=0;
            doubleob[m-1]=0;
            cout<<"\nEnter the integer array:-\n";
            intob.read(n);
            cout<<"\nEnter the double array:-\n";
            doubleob.read(m);
            cout<<"\nUnsorted integer array is:-\n";
            intob.disp(n);
            cout<<"\nUnsorted double array is:-\n";
            doubleob.disp(m);
            intob.bubble(intob.a,n);
            doubleob.bubble(doubleob.a,m);
            cout<<"\nSorted integer array is:-\n";
            intob.disp(n);
            cout<<"\nSorted double array is:-\n";
            doubleob.disp(m);
            return 0;
        }
```
