PART B

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;
}
```