# PART A

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<<"\nr
esult= ";
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;
}
```

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();
}
}
}
```