

```
:::::::::::::
tables1.cc
:::::::::::::
#include<iostream>
using namespace std;

const int MAX_TABLE_SIZE=5;

struct Seating
{
    int table[MAX_TABLE_SIZE+1];
    Seating();
    int getTable(int num);
    void freeTable(int num);
    void printFree();
};

int Seating::getTable(int num)
{
    while(num<MAX_TABLE_SIZE+1)
    {
        if(table[num]>0)
        {
            table[num]--;
            return(num);
        }
        num++;
    }
    cout << "There was no table available for " << num << " people\n";
    return(-1); // No tables that work available!
}

void Seating::freeTable(int num)
{
    table[num]++;
}

void Seating::printFree()
{
    int i;
    for(i=0;i<MAX_TABLE_SIZE+1;i++)
    {
        cout << "There are "<< table[i]<< " tables of size: "
            << i << " unused\n";
    }
    cout << endl;
}

Seating::Seating()
{
    table[0]=0;
    table[1]=0;
    table[2]=3;
    table[3]=1;
    table[4]=4;
    table[5]=1;
}

main()
{
    Seating list;
    list.printFree();
    list.getTable(3);
    list.getTable(3);
    list.getTable(3);
    list.getTable(3);
    list.getTable(3);
    list.getTable(2);
}
```

```
        list.printFree();
    }
    ::::::::::::::
tables2.cc
    ::::::::::::::
#include<iostream>
using namespace std;

const int MAX_TABLE_SIZE=5;

struct Seating
{
    int table[MAX_TABLE_SIZE+1];
    Seating();
    Seating(int one, int two, int three, int four, int five);
    int getTable(int num);
    void freeTable(int num);
    void printFree();
};

int Seating::getTable(int num)
{
    int i=num;
    while(i<MAX_TABLE_SIZE+1)
    {
        if(table[i]>0)
        {
            table[i]--;
            return(i);
        }
        i++;
    }
    cout << "There was no table available for " << num << " people\n";
    return(-1);
}

void Seating::freeTable(int num)
{
    table[num]++;
}

void Seating::printFree()
{
    int i;
    for(i=0;i<MAX_TABLE_SIZE+1;i++)
    {
        cout << "There are "<< table[i]<< " tables of size: "
            << i << " unused\n";
    }
    cout << endl;
}

Seating::Seating(int one,int two,int three,int four,int five)
{
    table[0]=0;
    table[1]=one;
    table[2]=two;
    table[3]=three;
    table[4]=four;
    table[5]=five;
}

Seating::Seating()
{
    table[0]=0;
    table[1]=0;
    table[2]=3;
```

```
        table[3]=1;
        table[4]=4;
        table[5]=1;
    }

main()
{
    Seating smoking(4,2,2,0,0);
    Seating non(0,2,0,4,1);
    non.getTable(4);
    non.getTable(1);
    smoking.getTable(3);
    smoking.getTable(3);
    non.getTable(2);
    smoking.getTable(3);
    smoking.getTable(3);
}
::::::::::::
oo0.cc
::::::::::::

#include<iostream>
using namespace std;

struct Cmpx
{
    double real;
    double img;
    Cmpx add(Cmpx x);
};

Cmpx Cmpx::add(Cmpx x)
{
    Cmpx tmp;
    tmp.real=x.real+real;
    tmp.img=x.img+img;
    return(tmp);
};

main()
{
    Cmpx a,b,c;
    a.real=1;
    b.real=3;
    a.img=5;
    b.img=7;

    c=b.add(a);

    cout << "("<< c.real << " + "<< c.img <<"i)\n";
}
```