16.5/20

NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE

FAST

Course Name & Section OOP Date

Student's Name Hafiz Sachal Roll No 23L-0973 Signature Sachal

```
#include<iostream>
using namespace std;
                                  Assay
class ComplexNumbers{
private:
double *real;
double *imaginary;
int size;
static totalCount;
                                  Assay
ComplexNumber(double r, double i)
{
    real = nullptr;
    imaginary = nullptr;
    size = 0;
}
                                  Assay
ComplexNumber(ComplexNumber *arr1, ComplexNumber *arr2){
    arr1.real = nullptr;
    arr1.imaginary = nullptr;
    size
for(int i=0; i<size; i++){
    arr1.real = nullptr;
    arr2[i].real = nullptr;
    arr1.imaginary = nullptr;
    arr2.real = nullptr;
    arr2.imaginary = nullptr;
    size = 0;
}
```

Q / Part No.

Rough Work

(5) -0.5

```
~ComplexNumberArray(){
    // no deallocation
    totalCount--;
}
```

(4) ✓

```
double *getReal(){
    return *real;
}
double *getImaginary(){
    return *imaginary;
}
~~double~~ int getSize(){
    return size;}
static get totalCount(){
    return totalCount;}
```

(6) ✓

```
ComplexNumberArray(const ComplexNumber &obj){
    // memory leakage   // size =?
    *real = *obj.real
    *imaginary = *obj.imaginary
    totalCount++;
}
```

(7) -0.5

```
ComplexNumber Array operator=(const ComplexNumber && obj)
{
    // deep copy?
    real = obj.real;
    imaginary = obj.imaginary;
    totalCount++;
}
```

Q / Part No.

Rough Work

(8) -1

```
ComplexNumberArray &operator*
(ComplexNumberArray &obj){
ComplexNumberArray result(real * obj.imaginary +
obj.real * imaginary);
return result;}
```

(9) ✓ -0.5

```
ComplexNumberArray &operator-=
(ComplexNumberArray &obj){
if(sizeof(ComplexNumberArray) !=
sizeof(this->ComplexNumberArray)){
cout << "Error";
}
else{
obj(real + obj.real, imaginary
+obj.imaginary);
return obj;}
```

(10) ✓

```
double * operator[](int index)
const{
double *n = new double[2];
n[0] = *real[index];
n[1] = *imaginary[index];
return *n;
}
```

Part No.

Rough Work

(11) -0.5

~~complexNumberArray~~ double *Mag(const){

ComplexNumber result(
sqrt(real*real)+(imm*imm))

return result;

}

~~void complexNumberPrint(){~~

~~cout <<~~

(12) ✓ void Print() const {

Use a loop for printing

cout << real << '+' <<
imaginary << 'i';

}