6.5/20

Improve your handwriting!

NATIONAL UNIVERSITY
OF COMPUTER & EMERGING SCIENCES
LAHORE

Course Name & Section OOP Date 25-3-24

Student's Name Syed Abdullah Ijaz Shah Roll No 23L-1000 Signature [illegible]

Quiz B

```
class   ComplexNumberArray

{
        private:
①       int size;              float          never initialize
-0.5    float *real = new ~~int~~ [size];       here
         float * imaginary = new ~~int~~ float[size];

        Static  total count ;
   Public;
      ~~ComplexNumberArray (float* real, float* imaginary~~
                                              , ~~int size)~~ ;
      Number
   ComplexArray (  );                       int size
   ComplexNumberArray ( float* arr1, float *arr2, );
   ~ ComplexNumberArray ();
                    ~~size~~
        float getreal ( ~~index~~ );
        float getimaginary( ~~size~~ );         } // getters
        ~~flo~~ int get size ( );

                    float
        void set real ( ~~st~~* real arr [ ], size);         } //setter
        void setimaginary( float * imagarr[], size);
        void  setsize ( int size );


   Complex Number ~~CompArr~~
      ~~ComplexNumberArray~~ :ComplexNumberArray   } copy
      (const  ComplexNumber Array & obj);       } cons
   {        complexNumbersArray  temp;
For(int i=0,i<size,i++){temp* real [i] = obj . real [i] ;
        temp* imagi [i] = obj . imaginary [i]; } return obj; }
```

Q / Part No. Rough Work

```
(8) ComplexNumberArray operatoroverload+ (ComplexNumberArray A1,
                                          ~~ComplexNumberArray A2)~~
    {   ComplexNumberArray temp;
  {   ~~for(int i~~ ~~int~~ if(sizeof(Array1)!= sizeof(Array2))
      {   cout<< error; }
   ~~if( size of~~
          else(~~[illegible]~~
     {       ~~temp~~
              for( int i=0, i<size ; i++ )
          {     temp.real[i] = Array1.real[i] +
                               Array1.real[i];
             temp.imaginary[i] = Array1.imaginary[i]
                                 +Array2.imaginary[i];
          }
      return temp;
                    }
    void print();
  }; → class ends

      ↓
```

Overload '+' operator

continued on next page

6	-2
7	-2
8	-2
9	-2
10	-2
11	-2
12	-1

Q / Part No. | Rough Work

~~ComplexArray~~ Nuber

(2) ComplexNumberArray::ComplexArray ( )

{ ~~*~~real=null;
~~*~~ imaginary=null;
size=0; } //default const

(3) ComplexNumberArray :: ComplexNumberArray(*arr1, *arr2, size)

{ not allocated, no default values
for(int i=0, i<size, i++) //Parameterized constructor
{ *real[*i] = *arr1[i];}}
* imaginary[i]=*arr2[i];}}
totalCount++;
}

~~float~~ ~~ComplexNumberArray::get real~~

(5) ~ComplexNumberArray ( )
{ ~~for(int i=0, i<size, i++)~~ //destructor
de
delete [ ]real;
delete [ ] imaginary;
real=null;
imaginary=null;

//getters
(4) float ComplexNumberArray::getreal ( )
{ return real; };
float ComplexNumberArray::get imaginary( )
{ return imaginary; };
int ComplexNumberArray:: getsize ( )
{ return size; };

//setter Not Needed
void ~~set~~ ComplexNumberArray::setreal(*realarr[], size)
{ for(int i=0, i<size ;i++){*real[i]=*realarr[i]; }
void ComplexNumber Array::setimaginary(*imaginaryarr[], s
{ for (int i=0, i<size ;i++){*imaginary[i]=
*imaginaryarr[i]; }

Void ComplexNumber::setsize(int size)
{this size =size; }