

Program_Project1

vectorDouble
<ul style="list-style-type: none"> - list: double* - maxCount: int - count: int
<ul style="list-style-type: none"> + ~VectorDouble() + VectorDouble() + VectorDouble(numberElements: int) + VectorDouble(object: VectorDouble& const) + push_back(newElement: double): void + capacity():int const + size():int const + reserve(newMaxCount: int): void + resize(newCount: int): void + valueAt(int i): double + changeValueAt(d: double, i: int): void + output(ostream& outs): void

Optional 2 points extra credit)

Program_Project2

Rational
<ul style="list-style-type: none"> - numbers[2] :int
<ul style="list-style-type: none"> + Rational() + Rational(wholeNumber: int) + setNumerator(newNumerator: int): void + getNumerator(): int const + setDenominator(newDenominator: int): void + getDenominator(): int const + normalization(): void + operator +(const Rational& r1 ,const Rational& r2): Rational <<friend>> + operator -(const Rational& r1, const Rational& r2): Rational << friend>> + operator *(const Rational& r1, const Rational& r2): Rational << friend>> + operator /(const Rational& r1, const Rational& r2): Rational << friend>> +operator -(): void + bool operator <(const Rational& r2) + bool operator <=(const Rational& r2) + bool operator ==(const Rational& r2) + bool operator >=(const Rational& r2) + operator <<(ostream& outs, const Rational& object): ostream& <<friend>> + operator >>(istream& ins, Rational& object): istream& <<friend>>

Program_Project3

Complex
- real: double - imaginary: double
+ Complex() + Complex(realPart: double) + Complex(realPart: double, imaginaryPart: double) + setReal(newReal: double): void + getReal(): double const + setImaginary(newImaginary: double): void + getImaginary(): double const + operator +(const Complex& c1 ,const Complex& c2): Complex <<friend>> + operator -(const Complex& c1, const Complex& c2): Complex << friend>> + operator *(const Complex& c1, const Complex& c2): Complex << friend>> + bool operator ==(const Complex& c2) + operator <<(ostream& outs, const Complex& object): ostream& <<friend>> + operator >>(istream& ins, Complex& object): istream& <<friend>>

Program_Project4

StringVar
- value: char* - maxLength: int
+ ~StringVar() + StringVar() + StringVar(size: int) + StringVar(const char a[]) + StringVar(const StringVar& sObject) + length():int const + inputLine(istream& ins): void + copyPiece(startIndex: int, length: int) : StringVar + oneChar(index: int) : char + operator +(const StringVar& s1 , const StringVar& s2): StringVar <<friend>> + bool operator ==(const StringVar& s2) + operator >>(istream& ins, StringVar & object): istream& <<friend>> + operator =(const StringVar& rightSide): void + operator <<(ostream& outs, const StringVar& object): ostream& <<friend>>

(Optional 3 points extra credit)

Program_Project11

You will design the UML, implement, and test.

(Optional 5 points extra credit for drawing the UML)