Object-Oriented Programming

C++

Q1: Operator Overloading for String Class Your goal is to overload the operators for \String" class that you have implemented in the last assignment. You will need to write three les (string.h, string.cpp and stringMain.cpp). Your implemented class must fully provide the de nitions of following class (interface) functions . Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class String{
// think about the private data members...
public:
// provide definitions of following functions...
String();// default constructor
String(char *str);// initializes the string with constant cstring
String(const String &);// copy constructor to initialize the string fromexisting string
String(int x);// initializes a string of pre-defined size
char operator[](int i);// returns the character at index [x]
String& operator+(String str );// append a String at the end of string
String& operator-(String substr);//removes the substr from the string
bool operator!();// returns true if string is empty..
String& operator=(const String&);// Copy one string to another ...
bool operator==(const String&);//returns true if two strings are equal
int operator()(char);// returns the index of character being searched.
String operator*(int a);//multiples the string by i times and return the string. Remember the Python functionality for *
int length();// returns the length of string
~String();// destructor...
};
istream& operator<<(istream& input,const String);//Inputs the string
ostream& operator>>(ostream& ouput,const String); //Outputs the string
```

Q2: Implementation of Array Class Your goal is to overload the operators for \Array" class that you have implemented in the last assignment. You will need to write three les (array.h, array.cpp and arrayMain.cpp). Your implemented class must fully provide the de nitions of following class (interface) functions . Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Array{
// think about the private data members...
public:
// provide definitions of following functions...
Array();// a default constructor
Array(int size);// a parametrized constructor initializing an Array of predefined ,! size

Array(int *arr, int size);// initializes the Array with an existing Array 8 Array(const Array &);// copy constructor
int operator[](int i);// returns the integer at index [i] after checking the out of ,! range error

void operator=(const Array);//copy the array
Array operator+(const Array);//adds two Array
Array operator-(const Array);//subtracts two Array
void operator++();//adds one to each element of Array
Array& operator--(int);//subtracts one from each element of array
bool operator==(const Array);//returns true if two arrays are same
bool operator!();// returns true if the Array is empty
void operator+=(const Array&);//adds two Array
void operator-=(const Array&);//subtracts two Array
  int operator()(int idx, int val);// erases the value val at idx. Returns 1 for a ,! successful deletion and -1 if idx
  does not exists or is invalid. Shift the ,! elements after idx to the left.
~Array();// destructor...
};
istream& operator<<(istream& input,const Array);//Inputs the Array
ostream& operator>>(ostream& ouput,const Array); //Outputs the Array
```

Q3: Implementation of Matrix Class Your goal is to overload the operators for \Matrix" class that you have implemented in the last assignment. You will need to write three les (matrix.h, matrix.cpp and matrixMain.cpp). Your implemented class must fully provide the de nitions of following class (interface) functions . Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Matrix{
// think about the private data members...
// the matrix should store real numbers
public:
//include all the necessary checks before performing the operations in the functions 6 Matrix();// a default constructor
Matrix(int, int);// a parametrized constructor 8 Matrix(const Matrix
&);// copy constructor
void operator()(int i, int j, float val);//set value at (i,j)
Matrix& operator=(const Matrix &);//assigns (copies) a Matrix. Returns the same
bool operator==(const Matrix &);//Compares two matrices
Matrix operator+(const Matrix &);//adds two Matrices and returns the result
Matrix operator-(const Matrix &);//subtracts two Matrices and returns the result

  Matrix operator*(const Matrix &);//multiplies two Matrices elementwise and returns ,! the result
Matrix& operator++(int);//add one to every element
void operator+=(const Matrix&);//adds two Matrices
void operator-=(const Matrix&);//subtracts two Matrices
~Matrix();
}
istream& operator<<(istream& input,const Matrix);//Inputs the Array
ostream& operator>>(ostream& ouput,const Matrix); //Outputs the Array
```

Q4: Implementation of Polynomial Class Your goal is to overload the operators for a generic \Polynomial" class. The internal representation of a polynomial consists of two integers i.e. coe cient and as exponent e.g the term $2x^4$ has the coe cient 2 and exponent 4. You will need to write three les (polynomial.h, polynomial.cpp and polynomialMain.cpp). Your implemented class must fully provide the de nitions of following class (interface) functions . Please also write down the test code to drive your class implementation. Please note that we will be running your code against our test code and any segmentation faults or incorrect result will result in loss of marks.

```
class Polynomial{
// think about the private data members...
// the matrix should store real numbers
public:
//include all the necessary checks before performing the operations in the functions Polynomial();// a default constructor
Polynomial(int, int);// a parametrized constructor Polynomial(const
Polynomial &);// copy constructor
Polynomial& operator=(const Polynomial &);//assigns (copies) a Polynomial. Returns ,! the same

bool operator==(const Polynomial &);//Compare and return true if equal
Polynomial operator+(const Polynomial &);//adds two Polynomial and returns the result
Polynomial operator-(const Polynomial &);//subtracts two Polynomial and returns the
 ,!    result

  Polynomial operator*(const Polynomial &);//multiplies two Polynomial and returns the ,! result
void operator+=(const Polynomial&);//adds two polynomials
void operator-=(const Polynomial&);//subtracts two Matrices
~Polynomial();
}
istream& operator<<(istream& input,const Polynomial);//Inputs the Array
ostream& operator>>(ostream& ouput,const Polynomial); //Outputs the Array
```