Rupakeerthana Vemulapalli
20778222

Design Document

- Commands to execute my program:
  Compile: g++ -std=c++11 polynomialtest.cpp Polynomial.cpp Polynomial.h

```
% g++ -std=c++11 polynomialtest.cpp Polynomial.cpp Polynomial.h
```

  Run:

```
eceTeslal:~% ./a.out <test01.txt
success
success
success
success
success
success
success
success
success
success
failure
```

- What class(es) did you design? What are the member variables and member functions for each of these classes?

  Class for my design was: Polynomial

  Member variables:

```
struct Coeff_List
{
    double coeff;
    struct Coeff_List *next_coeff;
};

// class variables
private:
int size, size_p2, size_sum;  // degree + 1
public:
Coeff_List *start, *end, *start_p2, *end_p2, *start_sum, *end_sum, *start_prod, *end_prod;
```

  The major member methods/functions and their functionalities I designed are shown below:

Member Function	**Function Description**
void init(int m)	Initializes the size of polynomial p1
void coeff_p1(const Polynomial &current_polynomial);	Initializes coeffp1 linked list
void coeff_p2(int expected_size, const double expected_coeff_p[]);	Initializes coeffp2 linked list
void get(int expected_size, const double expected_coeff_p[]) const ;	Checks if the size and the coefficients of polynomial p1 matches with the expected polynomial

void evaluate(const double x, const double expected_value) const;	Evaluates polynomial p1 at the value x and compares the result with expected_value
void add(int expected_size, const double expected_coeff_p[]) const;	Adds p1 and p2 and compares with the expected size and expected coeff[]
void mult(int expected_size, const double expected_coeff_p[]) const;	Multiplies p1 and p2 and compares with the expected size and expected coeff[]

- For each class, what are your design decisions regarding constructors?
    1) **Default Constructor**- Polynomial::Polynomial();. This initializes all the private and public variables to 0/NULL.
    2) **Polynomial given 2 parameters**- Polynomial::Polynomial (int x, const double coeff_p[]); This method initializes the size with the x value and iterates through coeffp[] array assigning values to the p1 linked list.

- For each class, what are your design decisions regarding destructors?
    - The destructor sets all the private variable back to null values.