

## Polynomial

Generated by Doxygen 1.14.0



<b>1 Namespace Index</b>	<b>1</b>
1.1 Namespace List	1
<b>2 Class Index</b>	<b>3</b>
2.1 Class List	3
<b>3 File Index</b>	<b>5</b>
3.1 File List	5
<b>4 Namespace Documentation</b>	<b>7</b>
4.1 CLASSES Namespace Reference	7
4.1.1 Detailed Description	7
4.2 main Namespace Reference	7
4.2.1 Detailed Description	8
4.2.2 Function Documentation	8
4.2.2.1 check_size()	8
4.2.2.2 input_polynomial()	8
4.2.3 Variable Documentation	8
4.2.3.1 choice	8
4.2.3.2 pol1	8
4.2.3.3 pol2	8
4.2.3.4 pol_difference	8
4.2.3.5 pol_division	8
4.2.3.6 pol_multiplication	8
4.2.3.7 pol_sum	9
4.2.3.8 value	9
4.3 test_clasees Namespace Reference	9
4.3.1 Detailed Description	9
4.3.2 Function Documentation	9
4.3.2.1 test_count()	9
4.3.2.2 test_create()	9
4.3.2.3 test_difference()	9
4.3.2.4 test_division()	9
4.3.2.5 test_multiplication()	10
4.3.2.6 test_show()	10
4.3.2.7 test_sum()	10
<b>5 Class Documentation</b>	<b>11</b>
5.1 CLASSES.Polynomial Class Reference	11
5.1.1 Detailed Description	11
5.1.2 Constructor & Destructor Documentation	12
5.1.2.1 __init__()	12
5.1.3 Member Function Documentation	12
5.1.3.1 coefficients()	12

5.1.3.2 count_polynomial()	12
5.1.3.3 degree()	12
5.1.3.4 dictionary()	13
5.1.3.5 polynomial_division()	13
5.1.3.6 polynomial_multiplication()	13
5.1.3.7 polynomial_operations()	13
5.1.3.8 show_polynomial()	14
<b>6 File Documentation</b>	<b>15</b>
6.1 CLASSES.py File Reference	15
6.2 main.py File Reference	15
6.3 test_clasees.py File Reference	16
<b>Index</b>	<b>17</b>

# Chapter 1

## Namespace Index

### 1.1 Namespace List

Here is a list of all namespaces with brief descriptions:

<a href="#">CLASSES</a>	7
<a href="#">main</a>	7
<a href="#">test_clasees</a>	9



## Chapter 2

# Class Index

### 2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

<a href="#">CLASSES.Polynomial</a>	
<a href="#">Polynomial</a> class . . . . .	<a href="#">11</a>





## Chapter 3

# File Index

### 3.1 File List

Here is a list of all files with brief descriptions:

<a href="#">CLASSES.py</a>	15
<a href="#">main.py</a>	15
<a href="#">test_clasees.py</a>	16



## Chapter 4

# Namespace Documentation

### 4.1 CLASSES Namespace Reference

#### Classes

- class [Polynomial](#)  
*[Polynomial](#) class.*

#### 4.1.1 Detailed Description

```
@file CLASSES.py
@brief Polynomial realisation
@author Yevik A. 421702
@see Polynomial
```

### 4.2 main Namespace Reference

#### Functions

- [check\\_size](#) (int degree, list coefficients)
- [input\\_polynomial](#) (str name)

#### Variables

- [pol1](#) = [input\\_polynomial](#)("first")
- [pol2](#) = [input\\_polynomial](#)("second")
- [choice](#) = int(input("Enter your choice: "))
- [value](#) = int(input("Enter your value: "))
- [pol\\_sum](#) = [Polynomial.polynomial\\_operations](#)([pol1](#), [pol2](#), "s")
- [pol\\_difference](#) = [Polynomial.polynomial\\_operations](#)([pol1](#), [pol2](#), "d")
- [pol\\_multiplication](#) = [Polynomial.polynomial\\_multiplication](#)([pol1](#), [pol2](#))
- [pol\\_division](#) = [Polynomial.polynomial\\_division](#)([pol1](#), [pol2](#))

### 4.2.1 Detailed Description

```
@file main.py
@brief Program file that is executed
```

### 4.2.2 Function Documentation

#### 4.2.2.1 `check_size()`

```
main.check_size (
    int degree,
    list coefficients)
```

#### 4.2.2.2 `input_polynomial()`

```
main.input_polynomial (
    str name)
```

### 4.2.3 Variable Documentation

#### 4.2.3.1 `choice`

```
main.choice = int(input("Enter your choice: "))
```

#### 4.2.3.2 `pol1`

```
main.pol1 = input\_polynomial("first")
```

#### 4.2.3.3 `pol2`

```
main.pol2 = input\_polynomial("second")
```

#### 4.2.3.4 `pol_difference`

```
main.pol_difference = Polynomial.polynomial\_operations(pol1, pol2, "d")
```

#### 4.2.3.5 `pol_division`

```
main.pol_division = Polynomial.polynomial\_division(pol1, pol2)
```

#### 4.2.3.6 `pol_multiplication`

```
main.pol_multiplication = Polynomial.polynomial\_multiplication(pol1, pol2)
```

#### 4.2.3.7 pol\_sum

```
main.pol_sum = Polynomial.polynomial_operations(pol1, pol2, "s")
```

#### 4.2.3.8 value

```
main.value = int(input("Enter your value: "))
```

## 4.3 test\_clasees Namespace Reference

### Functions

- [test\\_create\(\)](#)
- [test\\_count\(\)](#)
- [test\\_show\(\)](#)
- [test\\_sum\(\)](#)
- [test\\_difference\(\)](#)
- [test\\_multiplication\(\)](#)
- [test\\_division\(\)](#)

### 4.3.1 Detailed Description

```
@file test_classes.py  
@brief Tests for Polynomial class
```

### 4.3.2 Function Documentation

#### 4.3.2.1 test\_count()

```
test_clasees.test_count()
```

#### 4.3.2.2 test\_create()

```
test_clasees.test_create()
```

#### 4.3.2.3 test\_difference()

```
test_clasees.test_difference()
```

#### 4.3.2.4 test\_division()

```
test_clasees.test_division()
```

#### 4.3.2.5 test\_multiplication()

`test_clasees.test_multiplication ()`

#### 4.3.2.6 test\_show()

`test_clasees.test_show ()`

#### 4.3.2.7 test\_sum()

`test_clasees.test_sum ()`

## Chapter 5

# Class Documentation

### 5.1 CLASSES.Polynomial Class Reference

[Polynomial](#) class.

#### Public Member Functions

- [\\_\\_init\\_\\_](#) (self, int [degree](#), list [coefficients](#))
- [degree](#) (self)
- [dictionary](#) (self)
- [coefficients](#) (self)
- [show\\_polynomial](#) (self)
- [count\\_polynomial](#) (self, number)
- [polynomial\\_operations](#) (cls, pol1, pol2, choice)
- [polynomial\\_multiplication](#) (cls, pol1, pol2)
- [polynomial\\_division](#) (cls, pol1, pol2)

#### 5.1.1 Detailed Description

[Polynomial](#) class.

Sets the degree, the list of coefficients and dictionary for operations

#### See also

[Polynomial.show\\_polynomial](#)

[Polynomial.count\\_polynomial](#)

[Polynomial.polynomial\\_operations](#)

[Polynomial.polynomial\\_multiplication](#)

[Polynomial.polynomial\\_division](#)

## 5.1.2 Constructor & Destructor Documentation

### 5.1.2.1 `__init__()`

```
CLASSES.Polynomial.__init__ (  
    self,  
    int degree,  
    list coefficients)  
  
@brief Constructor  
:param degree: Polynomial degree  
:param coefficients: Polynomial coefficients  
@see Polynomial.degree  
@see Polynomial.coefficients  
@see Polynomial.dictionary
```

## 5.1.3 Member Function Documentation

### 5.1.3.1 `coefficients()`

```
CLASSES.Polynomial.coefficients (  
    self)  
  
@brief Coefficients getter  
:return: the list of coefficients
```

### 5.1.3.2 `count_polynomial()`

```
CLASSES.Polynomial.count_polynomial (  
    self,  
    number)  
  
@brief A method to count the polynomial value  
:param number: A number to count the polynomial value with  
:return: Value of a polynomial  
@see Polynomial.show_polynomial
```

### 5.1.3.3 `degree()`

```
CLASSES.Polynomial.degree (  
    self)  
  
@brief Degree getter  
:return: polynomial degree  
@see Polynomial.coefficients  
@see Polynomial.dictionary
```



#### 5.1.3.4 dictionary()

```
CLASSES.Polynomial.dictionary (  
    self)  
  
@brief Dictionary getter  
:return: operation dictionary
```

#### 5.1.3.5 polynomial\_division()

```
CLASSES.Polynomial.polynomial_division (  
    cls,  
    pol1,  
    pol2)  
  
@brief A method that allows to divide polynomials  
:param pol1: First polynomial  
:param pol2: Second polynomial  
:return: Division  
@see Polynomial.polynomial_operations  
@see Polynomial.polynomial_multiplication
```

#### 5.1.3.6 polynomial\_multiplication()

```
CLASSES.Polynomial.polynomial_multiplication (  
    cls,  
    pol1,  
    pol2)  
  
@brief A method that allows to multiply polynomials  
:param pol1: First polynomial  
:param pol2: Second polynomial  
:return: Multiplication  
@see Polynomial.polynomial_operations  
@see Polynomial.polynomial_division
```

#### 5.1.3.7 polynomial\_operations()

```
CLASSES.Polynomial.polynomial_operations (  
    cls,  
    pol1,  
    pol2,  
    choice)  
  
@brief A method that allows to make operations sum and difference  
:param pol1: First polynomial  
:param pol2: Second polynomial  
:param choice: Str "s" or "d" depends on operation  
:return: Result of an operation  
@see Polynomial.polynomial_multiplication  
@see Polynomial.polynomial_division
```

#### 5.1.3.8 show\_polynomial()

```
CLASSES.Polynomial.show_polynomial (  
    self)
```

```
@brief A method used to show the polynomial in math form  
:return: A string that represents the polynomial  
@see Polynomial.count_polynomial  
@see Polynomial.polynomial_operations
```

The documentation for this class was generated from the following file:

- [CLASSES.py](#)

## Chapter 6

# File Documentation

### 6.1 CLASSES.py File Reference

#### Classes

- class [CLASSES.Polynomial](#)  
*Polynomial* class.

#### Namespaces

- namespace [CLASSES](#)

### 6.2 main.py File Reference

#### Namespaces

- namespace [main](#)

#### Functions

- [main.check\\_size](#) (int degree, list coefficients)
- [main.input\\_polynomial](#) (str name)

#### Variables

- [main.pol1](#) = [input\\_polynomial](#)("first")
- [main.pol2](#) = [input\\_polynomial](#)("second")
- [main.choice](#) = int(input("Enter your choice: "))
- [main.value](#) = int(input("Enter your value: "))
- [main.pol\\_sum](#) = [Polynomial.polynomial\\_operations](#)(pol1, pol2, "s")
- [main.pol\\_difference](#) = [Polynomial.polynomial\\_operations](#)(pol1, pol2, "d")
- [main.pol\\_multiplication](#) = [Polynomial.polynomial\\_multiplication](#)(pol1, pol2)
- [main.pol\\_division](#) = [Polynomial.polynomial\\_division](#)(pol1, pol2)

## 6.3 test\_clasees.py File Reference

### Namespaces

- namespace [test\\_clasees](#)

### Functions

- [test\\_clasees.test\\_create](#) ()
- [test\\_clasees.test\\_count](#) ()
- [test\\_clasees.test\\_show](#) ()
- [test\\_clasees.test\\_sum](#) ()
- [test\\_clasees.test\\_difference](#) ()
- [test\\_clasees.test\\_multiplication](#) ()
- [test\\_clasees.test\\_division](#) ()

# Index

- `__init__`
    - `CLASSES.Polynomial`, [12](#)
- `check_size`
  - `main`, [8](#)
- `choice`
  - `main`, [8](#)
- `CLASSES`, [7](#)
- `CLASSES.Polynomial`, [11](#)
  - `__init__`, [12](#)
  - `coefficients`, [12](#)
  - `count_polynomial`, [12](#)
  - `degree`, [12](#)
  - `dictionary`, [12](#)
  - `polynomial_division`, [13](#)
  - `polynomial_multiplication`, [13](#)
  - `polynomial_operations`, [13](#)
  - `show_polynomial`, [13](#)
- `CLASSES.py`, [15](#)
- `coefficients`
  - `CLASSES.Polynomial`, [12](#)
- `count_polynomial`
  - `CLASSES.Polynomial`, [12](#)
- `degree`
  - `CLASSES.Polynomial`, [12](#)
- `dictionary`
  - `CLASSES.Polynomial`, [12](#)
- `input_polynomial`
  - `main`, [8](#)
- `main`, [7](#)
  - `check_size`, [8](#)
  - `choice`, [8](#)
  - `input_polynomial`, [8](#)
  - `pol1`, [8](#)
  - `pol2`, [8](#)
  - `pol_difference`, [8](#)
  - `pol_division`, [8](#)
  - `pol_multiplication`, [8](#)
  - `pol_sum`, [8](#)
  - `value`, [9](#)
- `main.py`, [15](#)
- `pol1`
  - `main`, [8](#)
- `pol2`
  - `main`, [8](#)
- `pol_difference`
  - `main`, [8](#)
- `pol_division`
  - `main`, [8](#)
- `pol_multiplication`
  - `main`, [8](#)
- `pol_sum`
  - `main`, [8](#)
- `polynomial_division`
  - `CLASSES.Polynomial`, [13](#)
- `polynomial_multiplication`
  - `CLASSES.Polynomial`, [13](#)
- `polynomial_operations`
  - `CLASSES.Polynomial`, [13](#)
- `show_polynomial`
  - `CLASSES.Polynomial`, [13](#)
- `test_clasees`, [9](#)
  - `test_count`, [9](#)
  - `test_create`, [9](#)
  - `test_difference`, [9](#)
  - `test_division`, [9](#)
  - `test_multiplication`, [9](#)
  - `test_show`, [10](#)
  - `test_sum`, [10](#)
- `test_clasees.py`, [16](#)
- `test_count`
  - `test_clasees`, [9](#)
- `test_create`
  - `test_clasees`, [9](#)
- `test_difference`
  - `test_clasees`, [9](#)
- `test_division`
  - `test_clasees`, [9](#)
- `test_multiplication`
  - `test_clasees`, [9](#)
- `test_show`
  - `test_clasees`, [10](#)
- `test_sum`
  - `test_clasees`, [10](#)
- `value`
  - `main`, [9](#)