Dr. Roxana Dumitrescu
Department of Mathematics
King's College London
email: roxana.dumitrescu@kcl.ac.uk

Exercises in C++
SHEET 4

Classes and overloaded operators (Part II)

Problem 1. Class Polynomial

Write a class **Polynomial** that represents the polynomial

$$a_0 + a_1 x + a_2 x^2 + ... + x^n. \tag{0.1}$$

Here the coefficients a_i are doubles which are associated with the polynomial itself, but x is an unknown.

The class **Polynomial** should have the following features.

- It should store the coefficients a_i in a vector.
- It should have a function **evaluate** which takes as a parameter x and evaluates the polynomial at x.
- It should have **a constructor** which takes a vector of coefficients.
- It should have a function **add** which can be used to add two polynomials.
- It should have **a default constructor** which generates the constant zero polynomial.
- It should have a **constructor** which takes a single double c as a parameter and generates the constant polynomial with $a_0 = c$.

Problem 3. Class YourVector

Write a class **YourVector** which has as **private members** the size of the vector (the number of values stored in the vector), an int pointer to the first element and the capacity (the maximum number of values that the vector can store).

The class should contain the following:

(1) Constructors

- Default Constructor .

  ```
  YourVector();
  ```

- Constructor which creates a vector with *size* elements which are initialised with 0.

  ```
  YourVector(int size);
  ```

- Copy constructor

  ```
  YourVector(const YourVector&);
  ```

(2) Destructor

```
~YourVector();
```

(3) Public member functions

- A member function **max()** which returns the largest value of YourVector ;
- A member function **sum()** which computes the sum of the elements of YourVector;
- A member function **get_size()** which returns the size of YourVector;
- A member function **push_back()** which adds a new element to YourVector;
- A member function **extract(int i, int j)** which returns a vector containing the elements between the position i and j;

(4) Overloaded operators (as members of the class).

- The assignment operator =;
- The operator +=;
- The operator *= which multiplies the elements of the vector by a real number;

  ```
  YourVector& operator*=(double);
  ```

- An operator which gives the value of the vector at index i.

  ```
  int operator [](int i) const;
  ```

- An operator which gives the value of the vector at index i and returns a modifiable value.

  ```
  int& operator [](int i);
  ```

- The unary operator -.

  ```
  YourVector& operator-();
  ```

(5) Non-member functions and operators

* A function **sum2Vectors()** which computes the sum of 2 vectors a and b which have the same size (element by element sum: $a.ptr[i] + b.ptr[i]$, $\forall i < size$);
* A function **product2Vectors()** which computes the product of 2 vectors a and b which have the same size (element by element multiplication: $a.ptr[i] * b.ptr[i]$, $\forall i < size$);
* The operator $==$ which returns 1 if two vectors coincide and 0 otherwise;

```
bool operator==(const YourVector&, const YourVector&);
```

* The operator **!=** which returns 0 if two vectors coincide and 1 otherwise;

```
bool operator!=(const YourVector&, const YourVector&);
```

* The operator $<<$ for *cout*.
* A function which returns the scalar product in $\mathbf{R}^n$ of two vectors which have the same size;

Problem 3. Exercises with <vector> from the STL

Problem 3.1. Write a function **mean** that computes the mean of a vector of doubles.

Problem 3.2. Write a function **standardDeviation** that computes the standard deviation of a vector of doubles.

Problem 3.3. Write a function **min** and a function **max** which each take a vector of doubles and return the max and the min, respectively.