Lab 3 – Functional objects

Exercise 1. Function Pointers

Change the file function_pointers.cpp so that it will compile with -std=c++03 flag.
Replace auto and decltype with appropriate function pointers.

Exercise 2. Lambda Expressions

Implement lambda expressions in the lines denoted by // (n) in the file lamba.cpp:

(1) Function that compares two elements a and b:
a<b iff a is closer to center than b.

(2) Function that compares two elements a and b such that
- negative numbers are after positive ones, with zero in the middle,
- if numbers have the same sign then odd numbers are after even ones.

e.g. 4 < 8 < 1 < 3 < 0 < -8 < -2 < -7 < -3

(3) Function without parameters that return random integer number from interval [a, b] (a and b both included). Changing values of a or b should change the interval used by generator.
Use `std::rand` to generate random integer.

(4) Function that generates arithmetic sequence with given *start* value and *step*.
Changing start does not change the beginning of the sequence but the change of *step* influences the function output.

(5) Function that for given standard container (vector, list, deque) computes l_1 norm i.e. the sum of the absolute values of elements in the container.
Try to use std::accumulate algorithm with another lambda expression to implement it.

(6) Function that for given array a and integer n returns a function with one parameter x that computes a value of a polynomial of degree n with coefficients a at the point x.

Exercise 3. Function Objects (Functors)

Implement class Printer that for given std::ostream, prefix and postfix will define unary functor that for argument x (of any type) will output to stream x surrounded by prefix na postfix.
e.g.

```
 Printer printer(std::cout,  "[ ", " ] " );
 printer("hello");  // [ hello ]
printer(5);           // [ 5 ]
```