POO LAB 3

Write classes **Add, Sub, Neg, Value** and **Expression** **(modeling integer expressions)** so that the following code

```
int main() {
    Expression* e = new Add(new Neg(new Sub(new Value(10), new Value(20))), new Value(5));
    e->print();
    printf(" =  %d\n", e->compute());
    printf("%d parameter(s) => (left = %d, right = %d)\n",
          e->get_children_count(), e->get_child(0)->compute(),
          e->get_child(1)->compute());
    Expression* left = e->get_child(0);
    printf("%d parameter(s) => (value = %d)\n",
         left->get_children_count(), left->get_child(0)->compute());
    delete e;
}
```

compiles and upon execution prints the following to the screen:

```
(( - (10 - 20)) + 5) =  15
2 parameter(s) => (left = 10, right = 5)
1 parameter(s) => (value = -10)
```

Observations:

- You are not allowed to use **STL** at all (for vectors, strings, maps or any **template/object** defined in STL). The only exception is the usage of "**std::cout**" from the main function
- For the purpose of this example, you need to consider **polymorphism** (as a concept that can help you solve this problem)
- The final delete indicates that destructors should also be virtual and implemented accordingly.

Grading (informative):

G1	Constructor for Add, Sub, Neg and Value (1p / class)	4p
G2	Destructor for Add, Sub, Neg and Value (1p / class)	4p
G3	Organize your project in 10 files: **main.cpp**, **Add.h** , **Add.cpp, Sub.h** , **Sub.cpp, Neg.h** , **Neg.cpp , Value.h** , **Value.cpp , Expression.h**	1p
G4	Organize your classes Add, Sub, Neg and Value to include private and public members, the definition of a constructor and destructor, and at least one method.	2p
G5	Method: print() that uses polymorphism for Add, Sub, Neg and Value (1.5p / class)	6p
G6	Method: compute() that uses polymorphism for Add, Sub, Neg and Value (1.5p / class)	6p
G7	Method: get_children_count() that uses polymorphism for Add, Sub, Neg and Value (0.5p / class)	2p
G8	Method: get_child() that uses polymorphism for Add, Sub, Neg and Value (0.5p / class)	2p
G9	Program compiles and upon execution produces the expected results	3p