C++ Programming - Handout 1

Konrad Wojda, 9307820244

Exercise 1

Code

```cpp
#include <iostream>

int main() {
    int x = -5, y = 7;
    double item = 1.5, MIN = -12.0;
    char DAY = 'M';
    int num = 12, power = 1024;
    int MAX = 1024, Sens = 12;

    bool condition1 = (x > y) && !y;
    std::cout << "(x > y) && !y: " << condition1 << std::endl;

    bool condition2 = (item > MIN) || (DAY != 'M');
    std::cout << "(item > MIN) || (DAY != 'M'): " << condition2 << std::endl;

    bool condition3 = ((num * 128) < power) && y;
    std::cout << "((num * 128) < power) && y: " << condition3 << std::endl;

    bool condition4 = (!(power != MAX)) && (Sens == num);
    std::cout << "(!(power != MAX)) && (Sens == num): " << condition4 <<
std::endl;

    bool condition5 = ((y + x) < num) || (DAY == 'M');
    std::cout << "((y + x) < num) || (DAY == 'M'): " << condition5 <<
std::endl;

    bool condition6 = (Sens * (!y)) != 0;
    std::cout << "(Sens * (!y)) != 0: " << condition6 << std::endl;

    bool condition7 = (!x || y) && (!y || x);
    std::cout << "(!x || y) && (!y || x): " << condition7 << std::endl;

    return 0;
}
```

Output:

```
(x > y) && !y: 0
(item > MIN) || (DAY != 'M'): 1
((num * 128) < power) && y: 0
(!(power != MAX)) && (Sens == num): 1
((y + x) < num) || (DAY == 'M'): 1
(Sens * (!y)) != 0: 0
(!x || y) && (!y || x): 1
```

Solution:

Condition	Result
(x > y) && !y	false
(item > MIN) || (DAY != 'M')	true
((num * 128) < power) && y	false
(!(power != MAX)) && (Sens == num)	true
((y + x) < num) || (DAY == 'M')	true
(Sens * (!y)) != 0	false
(!x || y) && (!y || x)	true

Exercise 2

Code

```cpp
#include <iostream>

int main() {
    int a,b,c;

    std::cout << "Enter three numbers." << std::endl;

    std::cin >> a >> b >> c;

    int max_value = a;

    if (b > max_value) {
        max_value = b;
    }

    if (c > max_value) {
        max_value = c;
    }

    std::cout << "The largest number: " << max_value << std::endl;

    return 0;

}
```

Execution

```
konradwojda@konradwojda-comp:~/studia/cpp-hyu/01-hw$ ./a.out
Enter three numbers.
-1
-100
0
The largest number: 0
```

Exercise 3

Code

```cpp
#include <iostream>

int main() {
    int num;
    int sum = 0;

    do {
        std::cout << "Enter a number. Enter 0 to exit." << std::endl;
        std::cin >> num;

        sum += num;

        std::cout << "Cumulative sum: " << sum << std::endl;

    } while (num != 0);

    return 0;
}
```

Execution

```
konradwojda@konradwojda-comp:~/studia/cpp-hyu/01-hw$ ./a.out
Enter a number. Enter 0 to exit.
1
Cumulative sum: 1
Enter a number. Enter 0 to exit.
10
Cumulative sum: 11
Enter a number. Enter 0 to exit.
15
Cumulative sum: 26
Enter a number. Enter 0 to exit.
10
Cumulative sum: 36
Enter a number. Enter 0 to exit.
0
Cumulative sum: 36
```

Exercise 4

Code

```cpp
#include <iostream>
#include <stdexcept>

int gcd(int a, int b) {
    if (a <= 0 || b <= 0) {
        throw std::invalid_argument("One of given number is not positive.");
    }

    if (a < b) {
        std::swap(a, b);
    }


    while (b != 0) {
        int r = a % b;
        a = b;
        b = r;
    }

    return a;
}

int main() {
    int a, b;

    std::cout << "Enter two positive integers: ";
    std::cin >> a >> b;

    int result = gcd(a, b);

    std::cout << "The GCD is: " << result << std::endl;

    return 0;
}
```

Execution

```
konradwojda@konradwojda-comp:~/studia/cpp-hyu/01-hw$ ./a.out
Enter two positive integers: -1
1
terminate called after throwing an instance of 'std::invalid_argument'
  what():  One of given number is not positive.
Przerwane (zrzut pamięci)
konradwojda@konradwojda-comp:~/studia/cpp-hyu/01-hw$ ./a.out
Enter two positive integers: 20
4
The GCD is: 4
```