



# Programowanie w środowiskach zintegrowanych

No to zabieramy się do roboty

Let's get to work!

# Ćwiczenie I

- Napisz prostą aplikację testującą znajomość programowania w C++
- Program ma zawierać od 3-10 pytań.
- Musi zawierać pytania zamknięte, o jednej możliwej odpowiedzi albo wielu odpowiedziach
- Może zawierać także pytania otwarte
- Po zatwierdzeniu testu, należy podliczyć punkty użytkownika i zaznaczyć na formie złe odpowiedzi.

# Exercise I

- Implement a simple testing application with questions about C++.
- There should be 3-10 questions.
- There should be both multiple-choice questions with one or many answers correct.
- There could be questions where user has to submit answer in text field.
- After submitting the test, points should be awarded and wrong answers marked.

## Ćwiczenie 2

- Napisz humorystyczny program, w którym po najechaniu na przycisk „OK”, przycisk ten losowo zmienia swoje miejsce na formie
- Pamiętaj o wielkości formy i jak daleko możesz „wyjechać”.
- Zrób program tak, aby można go było rozszerzać i dalej działał.

## Exercise 2

- Implement a program where hovering over „OK” button makes it move to a random place on a form.
- Remember how to find out the actual size of the form.
- Make the program expandable.

# Ćwiczenie 3

- Wykonaj szyfrator/deszyfrator rot13.
-

# Exercise 3

- Make a ROT-13 scrambler

# Ćwiczenie 4

- Zaimplementuj algorytm Luhna do walidacji kart kredytowych.
- Dodawaj poprawne numery karty do listy
- Program powinien się rozszerzać.
- Dodaj kontrolki do filtrowania tzw. Major Industry Numbers:

[http://en.wikipedia.org/wiki/Bank\\_card\\_number](http://en.wikipedia.org/wiki/Bank_card_number)



# Exercise 4

- Implement Luhn algorithm to validate credit card number
- Add proper credit cards to the list.
- Make the program expandable.
- Add some controls to filter Major Industry Numbers:

[http://en.wikipedia.org/wiki/Bank\\_card\\_number](http://en.wikipedia.org/wiki/Bank_card_number)