**Podstawy Programowania  
Computer Programming 1**

**Sylabus zajęć   
Course Syllabus**

Prowadzący przedmiot  
Course leader  
  
**Janusz Stal**

Kraków

1. **Opis przedmiotu / Course details**

Liczba godzin / No of hours: **45** (studia stacjonarne / full-time study)

Kierunek studiów / Field of study : Informatyka Stosowana / Computer Science

Semestr / Semester: 1

1. **Prowadzący zajęcia / Course leader**

**Prof. UEK dr hab. Janusz Stal**

Gabinet / Room: budynek biblioteki, pokój 438 / library building, room 438

Email: <https://e-uczelnia.uek.krakow.pl/mod/page/view.php?id=455578>

1. **Cele zajęć / Class goals**

* wyjaśnienie roli, jaką obliczenia mogą odgrywać w rozwiązywaniu problemów  
  provide an understanding of the role computation can play in solving problems
* przegląd podstawowych pojęć związanych z programowaniem imperatywnym, proceduralnym i obiektowym  
  overview of the basic concepts related to imperative, procedural and object-oriented programming
* nabycie umiejętności tworzenia programów komputerowych  
  acquiring the ability to create computer programs

1. **Spis zagadnień / List of issues**

|  |  |
| --- | --- |
| **Tematy Topics** | |
|  | Zajęcia organizacyjne Course Introduction |
| CZĘŚĆ 1 – PODSTAWY PROGRAMOWANIA PART 1 – PROGRAMMING BASICS | |
|  | Typy danych, zmienne i operatory Types, variables and operators |
|  | Sterowanie przebiegiem wykonania programu Control structures |
|  | Podprogramy Subroutines |
|  | Test próbny Mock test |
|  | Kolokwium 1 Test 1 |
| CZĘŚĆ 2 – STRUKTURY DANYCH PART 2 – DATA STRUCTURES | |
|  | Tablice Arrays |
|  | Obsługa plików File handling |
|  | Słowniki, zbiory, stosy, kolejki, listy Dictionaries, Sets, stacks, queues, lists |
|  | Kolokwium 2 Test 2 |
| CZĘŚĆ 3 – PARADYGMATY PROGRAMOWANIA PART 3 – PROGRAMMING PARADIGMS | |
|  | Programowanie zorientowane obiektowo Object Oriented Programming |
|  | Programowanie funkcyjne Functional programming |
|  | Kolokwium 3 Test 3 |
|  |  |
|  | Kolokwium dla nieobecnych studentów Test for absent students |
|  | Oceny końcowe Final grades |

1. **Materiały do zajęć / Class materials**

Mark Lutz, Python. Wprowadzenie. Wydanie IV, Helion, Gliwice 2010.

The Python Tutorial  
<https://docs.python.org/3/tutorial/index.html>

Python Programming Tutorials (Computer Science): <https://www.youtube.com/playlist?list=PLi01XoE8jYohWFPpC17Z-wWhPOSuh8Er->

MIT’s Introduction to Computer Science and Programming in Python:  
<https://www.youtube.com/playlist?list=PLUl4u3cNGP63WbdFxL8giv4yhgdMGaZNA>

<https://ocw.mit.edu/courses/6-0001-introduction-to-computer-science-and-programming-in-python-fall-2016/>

Praktyczny kurs Git: <https://www.youtube.com/playlist?list=PLjHmWifVUNMKIGHmaGPVqSD-L6i1Zw-MH>

1. **System oceniania / Assessment rules**
   1. **Warunki zaliczenia przedmiotu / Conditions for completing the course**

|  |  |  |
| --- | --- | --- |
| **Nazwa aktywności Activity name** | **Forma kolokwium Test form** | **Punkty Points** |
| Kolokwium 1 Test 1 | Zadania praktyczne Practical assignments | 0 – 10 |
| Kolokwium 2 Test 2 | Zadania praktyczne Practical assignments | 0 – 10 |
| Kolokwium 3 Test 3 | Zadania praktyczne Practical assignments | 0 – 10 |
| **Suma / Total** | | 0 - 30 |
| *Obecność na zajęciach Attendance at classes* | *Punkty bonusowe za wysoką frekwencję Bonus points for high attendance* | *0 - 3* |

Możesz uzupełnić kolokwium tylko w przypadku twojej nieobecności oraz jeśli posiadasz usprawiedliwienie swojej nieobecności (zwolnienie lekarskie).

You can additionally take part in any test only if you were absent and you have an excuse for your absence (sick leave).

* 1. **Skala ocen / Grading scale**

Aby zaliczyć przedmiot, musisz uzyskać co najmniej 15 punktów (50%).

You need at least 15 points (50%) to complete the course.

|  |  |
| --- | --- |
| **Liczba punktów No of points** | **Ocena Grade** |
| ≥ 27 | bardzo dobry |
| ≥ 24 | + dobry |
| ≥ 21 | dobry |
| ≥ 18 | + dostateczny |
| ≥ 15 | Dostateczny |
| poniżej 15 | Niedostateczny |