

Regulamin oceniania

Ocena wystawiana jest na podstawie regularnej pracy w trakcie semestru – oddawanych projektów i odpowiedzi przy tablicy. Ocena w indeksie wystawiana jest na podstawie oceny wyrażonej w punktach według poniższej tabeli:

Liczba punktów	Ocena
<0-50)	2
<50-60)	3
<60-70)	3.5
<70-80)	4
<80-90)	4.5
<90-100>	5

Tabela 1. Przeliczenie oceny punktowej na ocenę końcową

Ocena projektów to maksymalnie 80 punktów, suma ocen wszystkich odpowiedzi przy tablicy i kolokwium to maksymalnie 20 punktów.

Zajęcia laboratoryjne trwają 14 godzin i odbywają się w sesjach dwugodzinnych. Większość zajęć laboratoryjnych podzielona jest na dwie części.

- Część pierwsza ma charakter ćwiczeniowy - odpowiedzi przy tablicy. Dyskutowane są zagadnienia według harmonogramu. Odpowiedź osoby zgłaszającej się na ochotnika jest oceniana w skali od 0 do 10, osoby wyznaczonej przez prowadzącego w skali od –10 do 10. Prowadzący zajęcia ma obowiązek zapewnić co najmniej jedną okazję do odpowiedzi dla każdej z osób w grupie. Na drugich zajęciach zostanie przeprowadzone kolokwium, o maksymalnej ocenie 10 punktów.
- Część druga obejmuje dyskusję na temat projektu.

Harmonogram ćwiczeń i terminów rozliczenia poszczególnych etapów projektu jest przedstawiony w tabeli 2. Za opóźnienie projektu punktacja obniżana jest o 4% za każde 24 godziny upływające od końca zajęć. Zajęcia w pierwszym tygodniu dla grupy x1 są traktowane jako zajęcia organizacyjne.

Numer zajęć laboratoryjnych	Tematy prowadzonych równoległe wykładów	Rozdział w podręczniku Aho, Sethi, Ullman	Praca domowa do oddania na zajęciach	Maksymalna punktacja danych prac domowych	Tematyka ćwiczeń
1.	1. Wprowadzenie i analiza leksykalna 2. Analiza składniowa 1	1,2,3.1-3.5 4			Generacja kodu
2.	3. Analiza składniowa 2 4. Analiza semantyczna	5.1-5.6 6	1. Makefile dla przykładowego kompilatora wyrażeń arytmetycznych 2. Analizator leksykalny dla przykładowego kompilatora notacji wzrostkowej na przyrostkową wykonany przy pomocy programu flex	2 4	Kolokwium – generacja kodu
3.	5. Organizacja środowiska czasu wykonania 6. Generacja kodu	7.1-7.8 9.1-9.9	3. Przykładowy kompilator notacji wzrostkowej na przyrostkową zrealizowany przy pomocy programów flex i bison	4	Gramatyki i drzewa składniowe
4.	7. Optymalizacja kodu	10.1-10.10	4. Uzupełnienie maszyny wirtualnej o brakujące instrukcje	4	Tworzenie zbiorów first i follow
5.			5. Gramatyka przykładowego języka typu pascal w programie bison, tworzenie tablicy symboli	26	Optymalizacja

6.			6. Generator kodu dla wyrażeń	20	Rezerwa
7.			7. Instrukcje sterujące i funkcje	20	Podsumowanie

Tabela 2. Harmonogram zajęć