

Tutorial semestr 1 by słodkadiwczynna33

Wstęp

Witajcie na pierwszym semestrze informatyki na PG, znanym także jako przedszkole. Jeżeli ktoś nie umie nic z programowania to się trochę napoci, ale zdanie nie powinno sprawić problemu.

Są trzy matematyki i to one tak naprawdę sprawiają trudność, eTrapez i trzaskanie zadań to wasi przyjaciele (nie nauczycie się całek tydzień przed sesją). Na końcu dam wam ogólne tipy jak pracować efektywnie, żeby się ze wszystkim spokojnie wyrobić w kontekście programowania.

Tabela przedmiotów

Przedmiot	Skrót	Poziom trudności	Prawdopodobieństwo niepowodzenia
Algebra liniowa	AL	Średni	Średnie
Analiza matematyczna	AM	Duży	Duże
Matematyka elementarna	ME	Duży	Duże
Humanistyka dla inżynierów	HDMI	Mały	Zerowe
Hipertekst i hipermedia	HiH	Mały	Małe
Podstawy programowania	PP	Średni	Średnie
Wytwarzanie aplikacji internetowych	WAI	Mały	Średnie

Tutoriale przedmiotowe

Matematyki wszystkie razem

Prowadzącą jest Musielak, jest to jedna z lepszych wykładowczyń jakie spotkałam na PG, dobrze prowadzi też ćwiczenia.

Wykład jest w dobrej atmosferze ale w tempie Formuły 1, jak spojrzysz na sufit przez trzy sekundy to już go nie dogonisz, na szczęście są przerwy więc można trochę ułożyć myśli.

Generalnie to na wykładach jest jakieś wprowadzenie ze slajdów i potem robicie zadania z egzaminów - tutaj Musielak trochę się z was pośmiejże że ona robi szybko i macie skill issue jak tak nie umiecie.

Podsumowując, wykład obowiązkowy dla każdego kto chce zdać bez stresu i na ocenę lepszą niż 3.

Tutaj drobna uwaga na temat przydatności tych wszystkich przedmiotów - część koncepcji użyjecie, część nie. Głównym zadaniem analizy nie jest tworzenie nowych matematyków - zamiast tego ma ona wyprodukować u was regularność, która wam się przyda do końca studiów i potem życia. Nie oporujcie temu, wykorzystajcie tą okazję żeby zbudować sobie systemy nauki i pracy, zobaczycie że będzie warto.

Algebra liniowa

Zaliczenie: u nas były trzy kolokwia bez egzaminu

Najłatwiejsza z matematyk - algebra, punkty, wektory, proste, płaszczyzny, a na końcu jeszcze macierze i wyznaczniki. Dowiedziecie się dlaczego 1 może równać się 0 i w jaki sposób definiować swoje własne działania matematyczne. Przedmiot śmiesznie schematyczny i z tego powodu nie ma co się o nim rozwodzić, jak najbardziej do ogarnięcia.

Analiza matematyczna

Zaliczenie: Egzamin i dwa kolokwia

Pochodne, całki, szeregi, równania różniczkowe - te pięć słów zapadnie wam na długo w pamięć. Zadania są schematyczne tak samo jak na algebrze liniowej, ale jest ich więcej i jest szerszy materiał. Polecam sobie przeznaczyć godzinę dziennie na zadania i robić ile się da, eTrapez jest bardzo pomocny ale tylko do rzeczy których nie ogarniacie po wykładzie, bo po co podwajać naukę jak można trzaskać zadanka.

Matematyka elementarna

Zaliczenie: Egzamin

Według mnie najtrudniejsza z matematyk, zadania są pół na pół schematyczne, co oznacza dużo zadań typu zauważ, a one są zawsze trudniejsze niż odklepanie kilku kroków z listy.

Teoretycznie jest to powtórzenie wszystkiego z matury rozszerzonej minus geo/stereo/planimetrii, ale materiał jest wywindowany w górę pod kątem poziomu trudności i czasami dostaniecie zadanie wprost wredne.

Będąc po podobnym przedmiocie na innym wydziale skończyłam z 4 (a wcześniej miałam 5 bez żadnego problemu), do ogarnięcia ale tak jak wyżej polecam godzinę dziennie bo inaczej lipa, regularność jest kluczowa.

Humanistyka dla inżynierów

Przedmiot polegający na zestawie miniprzedmiotów z różnymi prowadzącymi, mi się podobało ale ogólnie mało przydatne:

Podprzedmiot	Prowadzący	Zaliczenie	Opis
Twórczość inżynierska	Adam Kaczmarek	Przepisanie formułki o twórczości wybitnej kontra niewybitnej, wypisanie minimum 20 zastosowań np. dla kabla USB (szubienica to dozwolone zastosowanie, oni tego nie sprawdzają tak naprawdę)	Coś tam mówi o byciu kreatywnym i innowacji
Filozofia	dr hab. Andrzej Lisak, prof. PG	Obecność na 4, można podejść o 5 ale on rzuci pytanie typu "wymień 20 kompozytorów baroku" i tyle z tego będzie, jak ktoś chce to można spróbować	Wykład którego nie można ominąć, wydrukujcie sobie lisakowe bingo dla najlepszego experience
Prawo patentowe	Jakaś smutna pani	Obecność, dla nieobecnych	Nie wiem o czym ona mówiła ale to

Podprzedmiot	Prowadzący	Zaliczenie	Opis
		sprawdzian ze slajdów	było nudne, mogłaby chociaż o licencjach powiedzieć tak pod programowanie a nie a jakiś quirkach prawa patentowego
Wiedza o kulturze (muzyka)	Mariusz Mróz	Kolokwium ze znanych wcześniej pytań	Nie używa mikrofonu i nie pozwala mieć otwartych komputerów, trochę się zachowuje jakby kogoś chciał walnąć krzesłem tak dla zasady, poza tym to całkiem ciekawie tłumaczy jak świadomie odbierać sztukę i puszcza muzykę z głośników
Język polski	Krzysztof Goczyła (były dziekan)	Kolokwium internetowe (w domu)	Fajnie mówi o błędach językowych, angażuje widownię i dzięki temu można się czegoś nauczyć

Hipertekst i hipermedia

Prowadząca: Wioletta Szwoch

Zaliczenie: Egzamin

Wykład nieco ciekawszy od patrzenia w zegarek przez trzy godziny, byłam raz i nic nie straciłam poza sesją gimnastyki na jednym z wykładów.

Laboratoria schematyczne i od lat te same, różni prowadzący.

Projekty debilnie proste - zrób tą rzecz za X punktów, nie zrób tego błędu bo będzie -Y punktów, ale wymagań jest dużo i mnie to zmęczyło, robicie stronę w HTMLu (bez JSa), potem XML i XSLT (tego ostatniego prawie nikt nie robi bo się nie chce nikomu i legitnie to wyszło ze standardów 'enterprise' minimum 15 lat temu).

Egzamin trudny, ale ABCD i w domu więc no.

Coś tam się Szwoch żaliła że za dobrze poszło to może was dociśnie. Przedmiot do zapomnienia, jest tu tylko po to żeby was zdenerwować jeśli będziecie chodzić na wykłady.

Podstawy programowania

Prowadzący: Dariusz Dereniowski

Zaliczenie: Kolokwium czyli tak naprawdę egzamin ale w połowie semestru a nie w sesji

Najciekawszy przedmiot na semestrze - robicie gry w C/C++, ale bez STL'a i często bez klas, więc tak naprawdę to C.

Coś wam powiem tutaj co się przyda na później - piszcie w czystym C żeby się odzwyczaić od cin/cout, jest ok. 100 wolniejszy od <stdio.h> i potem wam będzie niewygodnie przerabiać projekty na AiSD w drugim semestrze, zaufajcie mi że z 5 sekund execution time z cinem robi się 0.1 s ze scanfem.

Kolokwium to taka kompilacja w głowie i test znajomości standardu C, nic specjalnie trudnego a fajnie uczy rekurencji i jakiś edge casów językowych.

Laboratoria całkiem trudne szczególnie jeśli się trafi Pikies, wielu prowadzących, każdy ma inne zadania.

Projekty łatwe, piszcie ten pierwszy w ncurses a nie printfami, nie piszcie w conio.h bo to nie jest dobra biblioteka.

Wytwarzanie aplikacji internetowych

Prowadząca: Krystyna Dziubich

Zaliczenie: Egzamin

Fajne wprowadzenie do JSa, PHP i używania jakiejś bazy danych (u nas MongoDB).

Projekt ten sam od zawsze, łatwy. Napiszcie go uczciwie to dużo się z niego nauczycie.

Nie denerwujcie Dziubich bo ma temperament, ale ogólnie jest miła i pomocna.

Na zajęciach projektowych nie trzeba być poza podpisaniem się więc fajnie.

Egzamin nieprzyjemny, dużo pytań typu o co chodziło autorowi, na szczęście powtarza część pytań z przed roku więc łatwo wykuć.

Oddawanie projektów

Z tym jest prosta zasada - nie ważne co macie w kodzie (choć dobrze żeby działał i spełniał wymagania), ważne czy przegadacie sprawdzającego. Dosłownie każde oddawanie to jest charisma check i jak go przejdziecie to wyjdzie z dużą ilością punktów. Polecam przećwiczyć z kolegą opowiadanie o kodzie bo to nie jest nic trudnego jak się przyzwyczaić.

Antyplagiat

Na HiH i WAI nie istnieje, na PP już tak. Nie kopiujcie kodu od kolegów. Nie kopiujcie kodu z Geeks for Geeks i podobnych. Ogólnie to algorytm plagiatowy jest znany i jest wytłumaczony na naszym discordzie, ale macie tu szybkie wytłumaczenie i linki źródłowe:

“Gdyby kogoś interesowało jak działa antyplagiat to załączam taki materiał. Ten”antyplagiat” to po prostu aplikacja od Uniwersytetu Stanford, której wrzuca się kody wszystkich studentów i analizuje podobieństwa między każdym z nas. Nie jest ona w żaden sposób zintegrowana ze stosem. W skrócie, program ten analizuje kod bez względu na jego kolejność (w pliku, nie podczas wykonania), nazwy zmiennych etc. Następnie dzieli na krótkie fragmenty haszując je, a następnie porównuje z innymi te hasze - w taki sposób jest % podobieństwa :) Nie wiem, czy to wiedza powszechna, ale zapytałem u źródła. Warto przestudiować, albo prywatnie sprawdzać antyplagiat i być o krok przed nauczycielami..., ale to już chyba rady dla przyszłego rocznika.” - Mack

<https://www.youtube.com/watch?v=4fiF2YVpj8A>

<https://theory.stanford.edu/~aiken/publications/papers/sigmod03.pdf>

<https://theory.stanford.edu/~aiken/moss/>

AI

Na studiach nie będziecie robić cudów inżynierii, tylko dużo prototypów i programów typu To-Do App, które LLMy robią na prompta bo miliony studentów przed wami już je robiło więc mają je w bazie danych. Prowadzi to do tego, że jest wielka pokusa zrobić projekt w 10 minut i pora na CSa, ale to was okropnie ogłupi - po co chodzić na studia jak się nie będziecie uczyć?

Ja przyznam że korzystałam z nich przez pierwsze dwa semestry do robienia “łatwych” rzeczy typu skrypty w Bashu na drugim semestrze - w wyniku tego nic nie pamiętałam i od nowa się tego musiałam uczyć żeby robić użyteczne rzeczy dla siebie.

Nie korzystam z niczego od czerwca 2025 i czuję się dużo pewniej, widzę jak się cały czas czegoś uczę. Nie boję się swojego kodu bo go rozumiem.

Poza tym to robiłam kilka prawdziwych projektów których AI nie miało w bazie i pyk żaden chat mi nie dał rady pomóc (a to były proste frontendowe rzeczy w Javascriptcie), więc jestem przekonana że nie, nie zabierze mi pracy, a w dłuższej ramie czasowej mnie spowolni przez stopniowe ogłupianie i rozleniwienie. Pomagałam kilku kolegom którzy robili projekty na studia z LLMami i nic nie rozumieli - nie idźcie tą drogą bo nie jest przyjemna.

Podsumowując, I w LLM oznacza inteligencję.

Porady i tricki programistyczne

1. Używajcie Linuxa, może być w maszynie wirtualnej (WSL jest trochę słaby), polecam Linux Mint bo działa i podobnie wygląda jak Windows. Zobaczycie że instalacja bibliotek itd. jest dużo szybsza na Linuxie i wiele deweloperskich rzeczy zakłada że właśnie tam pracujecie
2. Używajcie Makefile żeby nie wpisywać tych samych komend po 30 razy, oto szybki tutorial jak skompilować plik main.c i uruchomić program:

```
// Makefile (w rootcie projektu)      default:      gcc -o  
program main.c      ./program
```

Tutaj to nie wygląda aż tak imponująco, ale weźcie pod uwagę że np. jeszcze chcecie dać input do programu i sprawdzić output:

```
// Makefile (w rootcie  
projektu)      default:      gcc -o program  
main.c      ./program < input.txt > output.txt  
diff output.txt expected_output.txt
```

 itd. Uruchamiamy ten skrypt wpisując make w terminalu tam gdzie mamy Makefile
3. Korzystajcie z live-reloadingu przy pracy ze stronami internetowymi, np. z live-server (`npm install -g live-server`) żeby nie klikać refresh na stronie co każdą zmianę kodu, VSCode ma fajny plugin do tego
4. Czego nie ma w wymaganiach tego nie robicie, co jest niejasne robicie jak wam się podoba (musicie tylko mieć gadkę na oddawaniu)
5. Nie komplikujcie sobie życia pięknym UI i animacjami, róbcie wymagania i tyle (tzn. to nie ma wpływu na punkty, pod naukę jak najbardziej róbcie co chcecie)
6. **Nie polecam programować o 4 rano? Potem koledzy nic nie pamiętają co pisali i zawsze to jest najgorszy kod w całym projekcie**

W każdym razie to wszystko co mam wam do powiedzenia, zapraszam do przeczytania tutorialu na semestr 2 i życzę powodzenia!