

Kursanalys av Programmeringsparadigm, DD1360 2023

Marcus Dicander^{a,1}

^aEECS, KTH

1. Generativ AI

ChatGPT är förmodligen en av de mest intressanta tjänster som någonsin har lanserats och lanseringen skedde i november 2022, precis innan kursstart. I den här kursen valde vi ett strängt förbud mot att använda generativ AI i lösningen av samtliga laborationer och mästarpöv. Med fri användning av ChatGPT så skulle kursens laborationer kunna lösas med en totalt sett enklare och väldigt annorlunda problemapproach där studenterna mer skulle vara kritiska mot AI:n, testa och utvärdera dess output och lära sig prompt engineering. Anledningen till att vi inte valde den approachen är flera: (1) Generativ AI kan inte göra de mest avancerade programmen och för att människor ska kunna göra dem så måste de behärska grunderna ordentligt. Det kommer de inte att göra om de får fritt fram att låta AI tänka åt dem. (2) ChatGPT producerar en hel del nonsens. Även som studenterna ska använda AI så måste de vara kritiska mot vad den ger ifrån sig och för att vara det så behöver studenterna kunna grunderna själva. (3) Prompt engineering kommer att förändras. De trick som används för att få en viss version av ChatGPT att göra rätt kan i framtiden komma att förändras rejält och då är kunskaperna lite mer av en kylskåpsvara än den långsiktigt hållbara utbildning som vi vill leverera. Med detta vill vi inte ha sagt att detta är onödiga kunskaper, men de kommer inte att gagna studenternas lärande i den här kursen så vi valde att förbjuda den helt. Året efter 2024 luckrades dessa regler upp pyttelitet och det blev tillåtet att använda generativ AI för att slå upp LaTeX-kommandon, men fortfarande inte lösa själva uppgifterna utöver det. Jag sneglade på Grudat på teknisk fysik för att se om de är för generösa eller lagom när de tillåter generativ AI att användas i alla fall där man hade kunnat kolla upp lösningen i en manual eller bok (utom förstås mönstermatching och liknande). Att hitta den exakta balansen till nästa gång blir en intressant utmaning.

2. Svåra mästarpöv

Examinationsgraden för kursen sjönk markant i och med införandet av mästarpöv. Mer specifikt har examinationsgraden för mästarpövet i Syntax blivit såpass lågt att vi fick hålla möten med PA och förklara och försvara hur de har gått till och varför det är så många underkända. Min förklaring är att studenterna inte vill dopaminfasta och läsa de krävande kursböckerna långsamt och noggrant på biblioteket utspritt över lång tid. Studentrepresentanterna har under åren föreslagit att vi ska ge mer detaljerade läsanvisningar (vilket har gjorts), rekommendera datum för att starta med uppgifter i kursen (vilket också har gjorts) men till syvende och sist så hänger det på studenterna att faktiskt ta den här delen av kursen på allvar och använda ordentliga källor som böcker istället för Youtube, Wikipedia, Stack Overflow och oförberedda, oplanerade angrepp på laborationerna. Den förklaringen köpte inte programledningen dock utan syntaxmästarpövet ska skalas ner till att bestå av tre delar som är strängt begränsade i om-

fattning, delresultat ska kunna återanvändas så att det inte blir såpass hårt straff för studenter som överlag gör bra mästarpöv men som underpresterar på ett av lärandemålen. Vi får se hur det går under 2024 (där vi just nu är mitt uppe i mästarpövsredovisningar då detta skrivs.) En sak som sågs som förvirrande i mästarpövet i Syntax var formuleringen “känner igen” om ett programmeringsspråk. Många studenter tolkade detta som att falska positiver var tillåtna och att man därför inte behöver göra en noggrann lösning som använder teorin för syntax i approachen. Trots generösa möjligheter att komplettera vid den muntliga redovisningen så misslyckades många på den uppgiften. Kanske formuleringen “känner igen exakt” hade fått färre att misstolka, men samtidigt är “känner igen” en etablerad term i syntax och den är uppenbar för alla som har läst boken ordentligt.

3. Många studenter från äldre kursomgångar

De vanligaste examinationsmomenten som studenter har kvar från tidigare kursomgångar är Inet, S2 och mästarpövet i Syntax. Ingen av dessa är särskilt svår egentligen, men många studenter vill inte läsa kurslitteraturen i Syntax utan tycker att i en såhär liten kurs så borde en liten arbetsinsats med Wikipedia, Youtube och Stack Overflow räcka. Det gör det inte. Uppifrån och nedifrån kommer då en önskan om att göra kursen enklare.

3.1. Labb S2

Specifikt för labb S2 så står det numera i labblydelsen att man ska göra en grammatik innan man börjar... och innan man gör grammatiken så ska man ha läst i boken... och under programmeringen så ska man göra egen testning. Tyvärr vill många studenter mot alla dessa instruktioner hoppa raka vägen på att implementera och vill att Kattistesterna ska berätta för dem vad de gör fel, men det är inte Kattis jobb och inte heller en approach som lär ut syntax!

3.2. Inet

Inet är en av de mest förekommande restlabbarna i prog. Det är väldigt märkligt att vi har haft nätverksprotokoll som en del av prog:n eftersom nätverksprotokoll vanligtvis räknas som en egen kurs, men tacksamt nog så är den biten numera bortplockad från programmeringsparadigmen vilket gör kursen enklare att förklara för studenterna. I år och bara i år kunde jag säga att kursen handlar om programmeringsspråk. Först och främst deras paradigm och sedan deras implementation med syntax. Precis som med S2 så är problemet att kvarvarande studenter inte vill läsa på teorin om nätverksprotokoll och skriva ner sitt protokoll utan bara hoppa rakt på implementationen vilket leder till att det tar ännu längre tid än om de hade läst på innan.

4. Kurslitteratur

Det finns en bra men aningen torr bok i Haskell som heter Graham Huttons Programming Haskell. De som väljer den boken och läser den ordentligt klarar i allmänhet kursen i programmeringsparadigm. Det finns också en uppskattad, enklare bok som heter Learn You a Haskell". Den senare boken är mer av en tutorial och går inte lika mycket på djupet, även där det vore önskvärt. De som har läst den ordentligt brukar dock klara kursen så den får jobbet gjort och den finns också gratis på nätet. Boken i syntax är alldeles utmärkt, men vice PA kommenterade att boken var aningen svår för en grundkurs.

5. Föreläsningar

Föreläsningarna är välbesökta och studenterna där är pigga och aktiva. Jag brukar göra snabba omröstningar där jag låter studenter peka på olika väggar eller taket i flervalfrågor. Tanken med detta är att (1) Det är ett lätt sätt att göra studenter engagerade. (2) Det är mycket smidigare än elektroniska omröstningar och jag är ändå inte intresserad av exakta resultat. (3) De som kan saker kan efteråt lära de som inte kunde. Kursens lärandemål presenteras mycket enligt principen om differentierad inläring, dvs att varje faktoid ska beskrivas mot bakgrund av sin historiska bakgrund, särskilt från vanliga missförstånd och liknande begrepp. Föreläsningarna är uppskattade och underhållande och åtminstone de i kursens första två delar får applåder.

6. Youtube

Alla föreläsningar från pandemin finns på Youtube. Detta har kursnämnderna lyft fram som positivt. Youtubeföreläsningarna har fått många views, över 1000 på de mest besökta men tyvärr infaller många av dessa strax innan deadline för mätarprov vilket säger mig att de kan få effekten att studenter sprider ut sin inläring för dåligt. Jag kommer dock att behålla dem eftersom de är bra för duktiga och ordentliga studenter.

7. Laborationer

Ett fåtal assistenter rättar laborationer utifrån issues utanför laborationstid och bokar in redovisningstider med studenter som sedan redovisar på den bokade tiden. Detta system är utmärkt när det väl fungerar, men det är tyvärr få assistenter som vill använda detta system så det blir tyvärr fallback till redovisning på plats med drop-in med långa köer som resultat. Incitamentet för assistenter att rätta issues utanför labbtid är att de får lön för det, men många föredrar det sociala. En idé till kommande år är att vi schemalägger issue-möten där assistenter sitter och rättar issues inför labbtillfällen. Det borde om inte lösa så iallafall lindra problemet.

8. Anonyma kursutvärderingar har brister

Kursnämnden sköter enkäten till studenterna om hur det har gått. Anledningen till detta är att de anonyma LEQ:erna har missbrukats av ett fåtal studenter som har missbrukat anonymiteten till att skriva fullständigt oacceptabla kommentarer genom åren.

9. Få studenter använder datorerna i datasalarerna

Vilka språk vi tar upp i kursen är anpassat efter vad som finns installerat på skolans datorer. Det här är för att de studenter som inte har egen dator ska kunna följa med i kursen, men i praktiken så kör de allra flesta studenter på sin egen dator. De berättar att de har bättre kontroll där över vad som är installerat och de behöver inte använda gamla versioner av exempelvis Python och C++ som finns på skolans datorer trots att vi påpekar detta för de ansvariga varje år.

10. Inrapportering av resultat har shadow-IT automatisering

I slutet av kursen så ska vi räkna ut vilka studenter som har gjort vad. Jag har överläggare med delresultat utspridda på över 10 år och här är våra IT-system mycket bristfälliga på nivån att jag blir alldeles förtvivlad. Exempel på uppenbara brister i Canvas som denna kurs lider av:

1. Det går inte att automatiskt slå ihop delresultat av flera omgångar av samma kurs. Detta problem är därför löst genom ett Shadow-IT-system som hämtar informationen ur Canvas. Detta system har dock ett aningen hackigt API eftersom det inte finns något bra officiellt API. Det fungerar alltså, men det kräver programmering från varje enskild lärare på varje kurs trots att detta borde hanteras centralt.
2. När ett betyg ska läggas in så förväntas man gå in på en sida som heter rapportera betyg. Först behöver man logga in på Canvas (vilket är en separat inloggning från Kth, och sessionen behålls inte så man behöver göra detta varje gång. Det finns ett trick där man kan öppna inloggningen i ett separat fönster så att man åtminstone inte behöver leta rätt på kursen igen. Nästa steg är att öppna bedömnings-sidan. Här går det tyvärr inte att spara separata view:s så att dessa blir lättåtkomliga och det går inte heller att ange view:s i URL-fönstret så att man kunde ha dessa i ett separat dokument med länkar. När man sedan lägger in sina betyg så måste alla lärare komma ihåg att Canvas glömmer bort det sista inmatade betyget (vilket i fallet ett betyg innebär det enda) om man inte klickar någon annanstans. Om det blir något fel här så blir det extra stress från studentens sida, mail fram och tillbaka tills vi har kommit fram till att en redovisning faktiskt har ägt rum. Jag bedömer att denna sida inte kan ha genomgått någon som helst kvalitetskontroll.
3. Canvas stödjer inte att ange en tidpunkt då en inrapportering skedde. Detta leder till att Canvas inte automatiskt hanterar bonusdatum. Vi har löst det i kursen genom att ha en separat uppgift för bonusdatum men det vore smidigare om det gick att ange datum för en inrapportering i ett separat fält. Detta stöddes av det gamla betygrapporteringssystemet Rapp.
4. Det finns inget enkelt sätt att ta reda på vilka studenter som är förförstagångsregistrerade och från vilket program de är. Detta gör det svårt att ta reda på vilka program som det går bra och dåligt för och att se hur det går för

de förförstagångenregistrerade studenterna över huvud taget. Det här borde vara one-click.