

Kursutvärderingar	2
Formativ kursutvärdering	2
Åtgärder och lärarnas svar på det som togs upp i den formativa kursutvärderingen	3
Åtgärder och förbättringar	4
Kompletterande enkät sista veckan före tentan (50 svaranden)	5
Analys av extra enkäten	9
Ordinarie kursvärdering	13
Jag har under kursens gång känt till och förstått kursens lärandemål.	13
Jag har fått tillräcklig information gällande kursens innehåll och upplägg för att kunna göra mitt bästa för att leva upp till kursens lärandemål.	13
Kursens undervisningsformer (som t ex föreläsningar, seminarier och övningar) har varit relevanta för kursens lärandemål.	14
Respondentkommentarer:	14
Vad är ditt allmänna omdöme om relevans och innehåll i föreläsningar, seminarier, övningar/laborationer, litteratur och examination?	15
Vad tycker du om kursens svårighetsgrad?	16
Det var tydligt vad som förväntades av mig i kursens examinerande moment (till exempel tentor, inlämningsuppgifter och projekt).	17
Jag har enligt min egen bedömning lagt tillräckligt med tid och engagemang på kursen för att klara av den.	17
Kursen har gett mig kunskaper och/eller färdigheter som jag tror jag kommer att ha nytta av i mitt framtida yrkesliv.	17
Fritextsvar:	17
Kommentarer till ordinarie kursvärdering	21
Tentamensresultat	21
lakttagelser från rättning av tentamen	24
lakttagelser vid rättning av Inlämningsuppgiften	27

Kursanalys - Programmeringsdelen

TIG015 HT2018

Kursutvärderingar

Formativ kursutvärdering

Under kursen utsågs tre kursrepresentanter och vi träffades vid två tillfällen och diskuterade kursens upplägg och allmänna synpunkter.

Följande kom fram (urval):

- Bra att det finns online (kursmaterial och filmer)
- Ok miljö i klassrummet
- Pedagogiskt, mycket lättare än de trodde först
- Övningarna känns svåra, men det verkar som att poängen gått fram att det är genom övning och diskussion med handledare som man lär sig
- Bra att vi får öva på att läsa kod
- Förklaring, kanske med pilar eller bilder, av kod vore bra
- Kanske vore studiegrupper bra
- Ibland blir det för snabba svar från handledarna (man vill hellre ha ledtrådar och försöka själv)
- Tog tid att hitta övningarna (och att hitta runt i allmänhet) på Wikin
- Rubriksnivåerna i schemat på wikin skulle vara bra om de på något sätt visade en vecka
- Vore bra med kodexempel i början av ett kapitel och en introduktion som sätter sammanhanget (syfte, mål, ...)
- Mer live-kodning (lärarna programmerar på projektorn "live" framför klassen)
- Presentationerna (pdf) kan man läsa själv, behöver inte presenteras
- Bättre om litteraturen är på svenska
- Begrepp (teori) tas upp blandat i pdf, wiki, övningar - Svårt få överblick
- "Det här kommer inte på tentan" sägs ofta
- Problematiska (eller felaktiga) formuleringar i wikin
- Gärna mer grafisk layout
- "Wiki-format" kanske inte det enklaste
- Ordningen på filmer förvirrande i wikin (ibland)
- Hade varit bra med två veckors repetition innan tentan
- Metainformationen i Wikin (mål, syfte, sammanhang för ett kapitel) hade varit bra om den var synlig direkt (så man slipper expandera den för att läsa den)
- Grafiska grejer - Varför får vi inte skriva mer grafiska program?

Åtgärder och lärarnas svar på det som togs upp i den formativa kursutvärderingen

Förklaring, kanske med pilar eller bilder, av kod vore bra

Vi har faktiskt lite bilder längre fram i kursen men vi ska se över hur vi ska kunna ta fram mer pedagogiska bilder som komplement till texterna i wikin. Vi tror dock att man får se kursmaterialet som en helhet som utgörs av Wiki med text och övningar, filmer med föreläsningar, klassrumsundervisning där vi ritar bilder på whiteboarden och handledning där vi går runt och diskuterar och ofta ritar bilder som en del av handledningen. Vissa filmer har mer bilder än andra. Vissa filmer har mer bilder än andra. Men vi ser över hur vi kan ha mer bilder i det pedagogiska materialet.

Ibland blir det för snabba svar från handledarna

Vi ska lägga till detta i handledarinstruktionerna att inte ge lösningen direkt. Dock så är handledarna inte pedagoger utan en extra resurs (som inte finns i varje kurs) som mer ska fungera som ett sätt att brygga över studenternas årskullar samt ett sätt att fråga någon som läst kursen själv och minns hur det är att lära sig och vara student.

Tog tid att hitta övningarna (och att hitta runt i allmänhet) på Wikin

Rubriksnivåerna i schemat på wikin skulle vara bra om de på något sätt visade en vecka

Detta är något vi ska adressera i början av nästa kurs och lägga mer tid på att förklara hur man hittar i wikin. Vår omedelbara åtgärd var att förtydliga navigationslänkarna längst ned på varje sida så att det är tydligare vad "Next" betyder (t ex tala om ifall det är övningar på samma ämne som aktuell sida eller om det är ett nytt ämne). Dessutom gjorde vi om schemat så att varje vecka var en huvudrubrik och veckans dagar var underrubriker.

Vore bra med kodexempel i början av ett kapitel och en introduktion som sätter sammanhanget (syfte, mål, ...)

Detta har vi tagit till oss. Nu har nästan alla kapitel (om inte alla till och med) ett kort kodexempel som sätter tonen och exemplifierar kod för kapitlets ämne. Vad gäller introduktion, syfte och mål, så finns i varje teorikapitel en meta-sektion som man kan expandera (den är kollapsad till en början) och där finns syfte, mål, introduktion och lärarinstruktioner (t ex vanliga frågor och vanliga missuppfattningar).

Vi överväger att göra delar av meta-informationen synlig direkt (syfte och mål). Varje kapitel har en synlig introduktionstext av varierande längd. Vi tror dock att vi kan upplysa om Meta-sektionen när vi introducerar till hur man ska använda wikin.

Mer live-kodning

Detta är absolut något vi planerar för. Vi märkte dock att när vi lade in extra live-kodning för dem som ville under vissa handledningspass, så var det inte många som kom.

Åtgärder och förbättringar

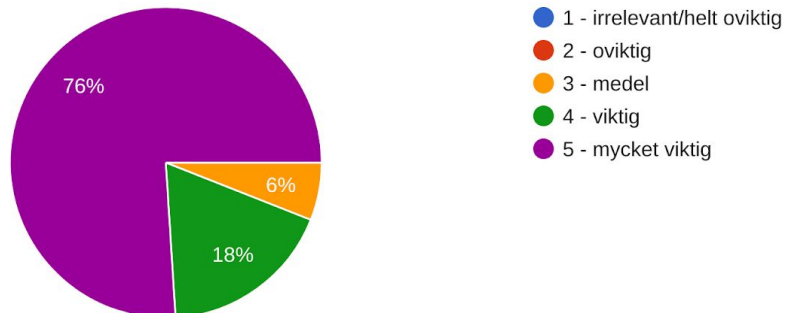
Nya övningar:

- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Defining_methods_-_Exercises#Q9 (Att ändra en String-variabel för födelsedatum i en klass till att vara av typen LocalDate i stället)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Defining_methods_-_Exercises#Q10 (Att lägga till en metod `boolean hasAdultHolder()` i en klass för att visa att man kan räkna ut ålder utifrån födelseår)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Declaring_variables_-_Exercises#Concluding_remarks (Nytt stycke som problematiserar valet av String som typ för födelsedatum.)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Defining_constructors_-_Exercises#Q20 (En konstruktor som tar emot tre int-värden för år-månad-dag och använder dessa för att sätta födelsedatumsvariabeln som är en LocalDate)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_main_-_Exercises#Q14. (Lägga till att programmet kan ta emot argument, så att programmet inte gör samma sak varje gång man kör det)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Packages_-_Exercise_s#Q9. [Download the Student and StudentDemo classes and put them in a package](#) (Ladda ned två paketlösa klasser, skapa paket-katalogstruktur, flytta ned klasserna i varsin katalog och lägg till paketdeklarationer i klasserna som reflekterar detta, lägg till en import-sats i main-klassen, kompilera och kör programmet.)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Declaration_-_Exercises (En övning i att ladda ned och köra två färdiga klasser, lägg till skapandet av ett Student-objekt som representerar dig själv, kompilera och kör igen)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Objects_-_Relation_to_classes_-_Exercises (Skriv av och skapa två klasser, kompilera och kör programmet. Studera hur indenteringen ser ut, hur krullparenteser sätts ut.)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Variables_and_types_-_Exercises#Java_primitive_types (5 nya övningar om Javas primitiva typer och literaler där studenten får ta hjälp av kompilatorn för att dra slutsatser om vilka typer olika literaler har)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Inheritance_-_Every_class_extends_Object_-_Exercises (Vi modifierade övningarna så att Team-klassen får en `numberOfMembers()`-metod, så att klientkod inte behöver hämta ut hela listan på members och fråga listan efter storleken - mer objektorienterat)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Classes_-_Other_keywords_-_Exercises (Övning på att läsa Java Language Specification och identifiera keywords/reserved words, samt en övning i att felaktigt försöka använda ett keyword som ett variabelnamn för att undersöka vad kompilatorn har att säga om det)
- http://wiki.juneday.se/mediawiki/index.php/Chapter:Interfaces_-_Creating_an_anonymous_class_-_Exercises (Övning på att skapa anonym lokal class för en Comparator)

Kompletterande enkät sista veckan före tentan (50 svaranden)

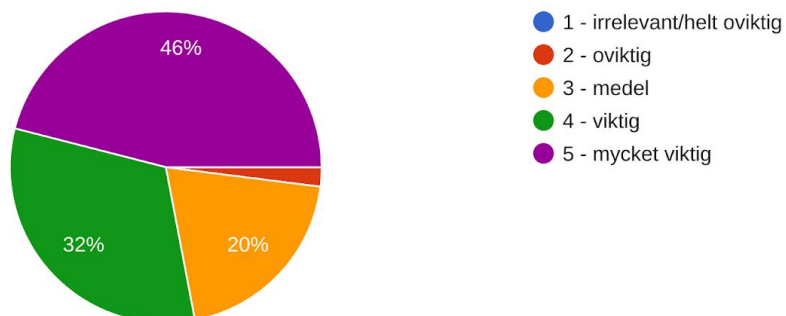
Hur viktig anser du att Juneday-wikin varit för ditt lärande

50 responses



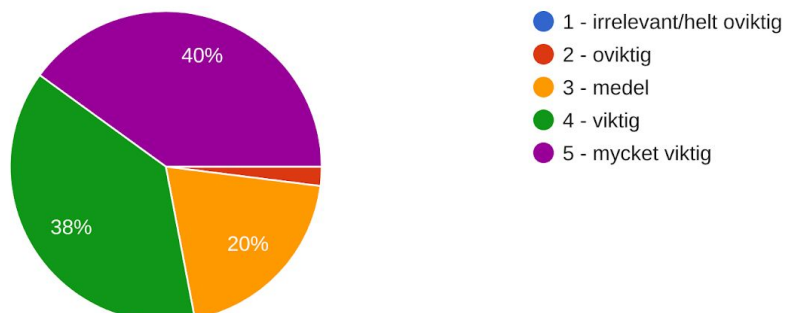
Hur viktig anser du att våra videofilmer varit för ditt lärande

50 responses



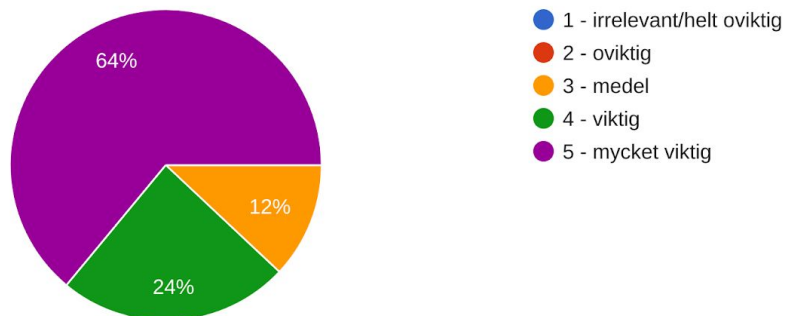
Hur viktig anser du att våra föreläsningar varit för ditt lärande

50 responses



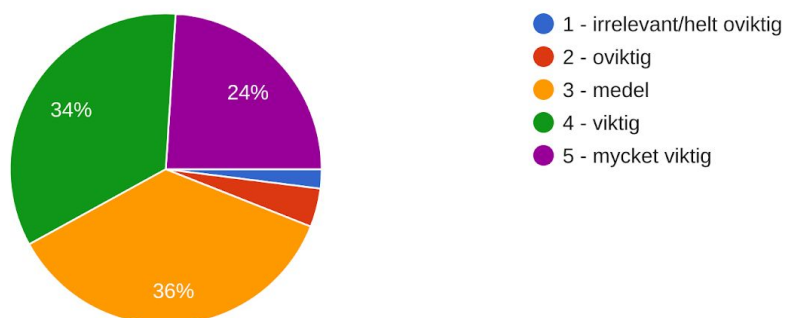
Hur viktig anser du att övningsuppgifter (inkl. lösningsförslag) varit för ditt lärande

50 responses



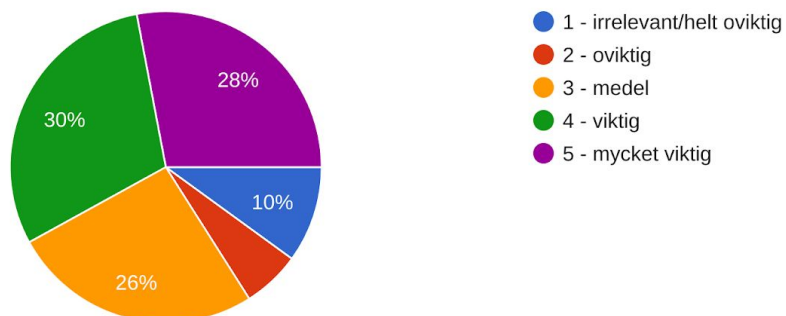
Hur viktig anser du att handledning (i klassrum) varit för ditt lärande

50 responses



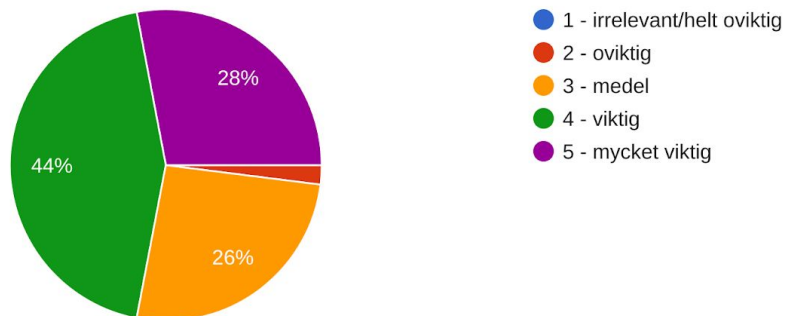
Hur viktig anser du att handledning utanför schemat (t ex genom att besöka lärarna på deras rum) varit för ditt lärande

50 responses



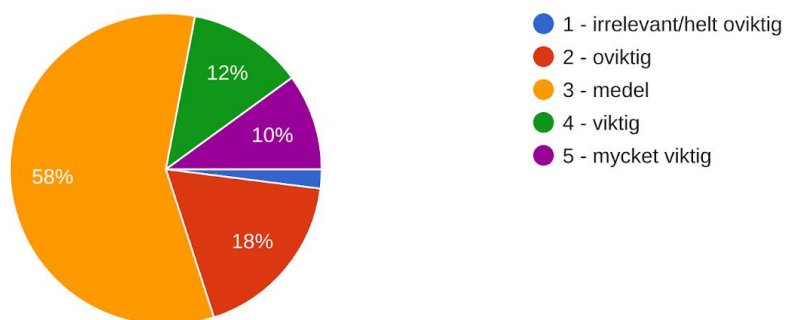
Hur viktig anser du att inlämningsuppgiften (Adressbook) varit för ditt lärande

50 responses



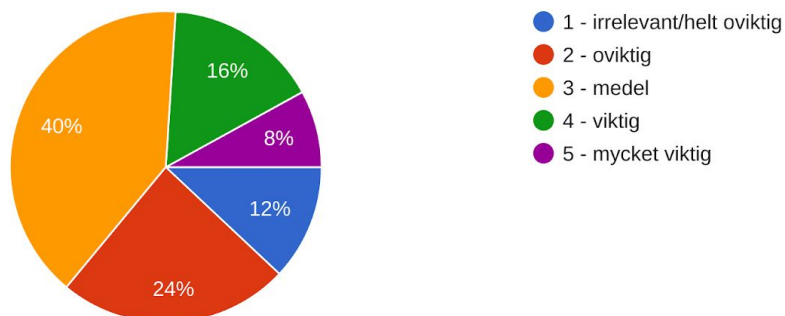
Hur viktig anser du att externa källor länkade från wikin varit för ditt lärande

50 responses



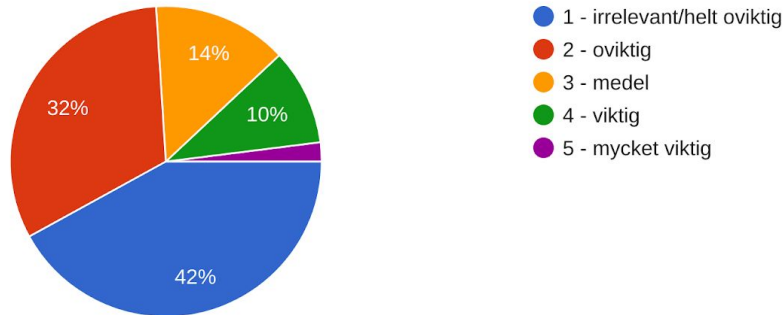
Hur viktig anser du att externa källor du hittat själv (t ex Youtube) varit för ditt lärande

50 responses



Hur viktig anser du att en kursbok i pappersform hade varit för ditt lärande

50 responses



Har du kommentarer på frågorna om ditt lärande?

12 responses

mer LIVE kodning

Mer liveprogrammering, man förstår bättre då

Pluggar java i pyjamas i soffan. 10/10 would learn again.

Nej

Mer liveprogrammering där ni snackar om vad ni gör, istället för bare snack där man ej hänger med tillslut för det blir för mkt termer

Wikin är underbar. Dock något mycket text ibland... men övningarna e grymma!

Skulle ha varit bättre att ha mer övningar (I alla delar av kursen), annars hade jag inte så mycket att göra på handledning. Mer övningar som AddressBook (kanske kan man ha en obligatorisk inlämning, och tre extra stora övningar)

När man får lite större program som innehåller saker vi inte lärt oss än i kursen, hade det varit till stor hjälp om info kring dessa kommandon etc hade haft förklaring direkt i uppgiften så man kan kolla direkt utan att gå i ovisshet tills det kommer senare i kursen eller rentav en helt annan kurs.

Jag tycker tiden för föreläsningar är för kort. Ofta måste vi hasta genom delar.

En kursbok hade erbjudigt en skärmfri stund av pluggande.

Jag har faktiskt lärt mig något nytt och vill lära mig mer om programmeringen

Har du tips på annan resurs som vi (Rikard och Henrik) kan kika på

13 responses

Mer live programmerings videor samt gärna en video där man ser era ansiktet. Tufft att stirra blint på text.

wiki på svenska för en introduktionskurs många nya begrepp

Istället för att skapa Members 10 ggr kan man kanske öva på att skriva något helt nytt som studenterna bestämmer? Göra ihop och diskutera fram

Nej tyckte upplägget för kursen har varit bra och har haft tillgång till allt jag behöver för att klara av kursen.

Mer livecoding så man kan koda med er (tex. en gång efter varje huvud kapitel, klasser t.ex) niet

Ge exempel på andra hemsidor där man kan lära sig java. Tex CodeAcademy o om det fungerar lika bra som er wiki.

astrologi

Skulle vilja ha en lista på ord som referensvariabel och objekt med en kort beskrivning. Det är så många uttryck att allt bara blir en röra inne i pallet.

Arbeta med interaktivt med oracles egna tutorials då de beskriver saker mer konsist. Wikin har varit väldigt ingående utan konsis sammanfattning av begrepp.

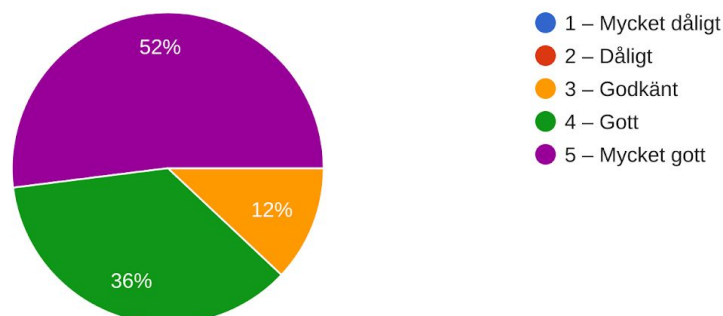
Jag har använt mig att boken Head first java, som har lärt mig en hel del igenom deras övningar. Enkla övningar som sammanfattat varje sak dom gått igenom. Även deras exemplar och bilder har hjälp mig en hel del att fatta hur objekt och array list fungerar. Bra bok, kolla där!

Lite gemensamma programmeringstillfällen där vi alla programmerar tillsammans.

Jag har använt mig av en app som heter SoloLearn under kursens gång. Denna ger översiktliga förklaringar om allt ni gått igenom.

Vad är ditt sammanfattande intryck av programmeringsdelen av kursen

50 responses



Analys av extra enkäten

Enkäten hölls sista veckan då det var repetition inför tentan. Av 50 närvarande studenter så svarade samtliga. Detta ger en 100% svarsfrekvens men urvalet var lite under 50% av registrerade studenter. Även om vi tolkar svarsfrekvensen som 50% (dock gavs inte alla chansen att svara men faktum är att vi fick in svar från nästan 50% av våra studenter), så är detta betydligt högre än vad den obligatoriska officiella kursvärderingsenkäten brukar ge.

Kursmaterialet på vår Wiki ges stor betydelse för studenternas lärande. 76% anger att den är mycket viktig, 18% att den är viktig och 6% att den är "medel". Ingen sätter den som mindre än medelviktig. Detta betyder att vi kommer fortsätta arbeta med wikin.

På frågan om hur viktiga våra **föreläsningsfilmer** var för studenternas lärande blev svaret 46% mycket viktigt, 32% viktigt och 20% medel. Endast 2% uppgav att videorna var oviktiga

och ingen uppgav att de var irrelevanta. Slutsatsen blir att vi ska fortsätta arbeta med föreläsningssvideor.

När samma fråga ställdes angående våra **föreläsningar** (i sal) uppger 40% att de var mycket viktiga för deras lärande, 38% att de var viktiga och 20% att de var medel. Endast 2% uppger att de var oviktiga. Slutsatsen blir att Wikin upplevs som mycket viktigare än våra föreläsningar och videofilmerna ungefär lika viktiga som våra föreläsningar. Av detta, kombinerat med vad som sagts angående önskemål om mer "liveprogrammering" i klassrummet drar vi slutsatsen att med förbättrad wiki, fler och bättre videor, så kan vi dra ned på föreläsningarna i sal om vi byter ut dem mot liveprogrammering i klassrummet motsvarande undervisningstid. Vi kommer införa detta på försök, möjligtvis redan i databas-kursen (halva TIG058) VT2019, i den mån vi hinner förbereda manus för liveprogrammering. Vi anser att liveprogrammering i sig inte har något större värde ifall man inte har en pedagogisk idé och ett manus att följa.

Samma fråga angående **övningsuppgifternas** betydelse gav resultatet 64% mycket viktigt, 24% viktigt och 12% medel. Ingen uppgav att övningarna hade mindre än medelbetydelse för deras lärande. Att hela 64% uppger att övningarna hade mycket viktig betydelse för lärandet är uppmuntrande då vi genomgående betonat att övningarna är det viktigaste verktyget för att nå målen för kursen. Vår analys ger att vi ska förbättra och öka antalet övningar så att de når upp till samma grad av betydelse som Wikin. Det är vår övertygelse att programmeringskunskaper kräver en stor mängd praktisk övning.

Nu finns det en svaghet med frågorna i enkäten som vi vill belysa här. Första frågan rör Wikins betydelse för lärandet. Det var lite olyckligt eftersom Wikin är en teknisk plattform mer än något annat. På wikin ligger textmaterial med teori och exempel. Men det är även via wikin man når videofilmerna. Det är även på wikin övningsuppgifterna (och inlämningsuppgiften) ligger. Även all källkod nås via wikin. Därför bör man nog slå ihop resultaten rörande Wikin, videofilmerna och övningarna i denna analys. Det blir dock då fortsatt tydligt att materialet som nås via Wikin (texter, exempel, schema, övningar, videofilmer) sammantaget får ett mycket högre betyg än våra föreläsningar i sal. Så vi står för slutsatserna ovan.

På frågan om **klassrumshandledningens** betydelse för studenternas lärande uppger endast 24% att den var mycket viktig. 34% uppgav att den var viktig och 36% att den var av medelbetydelse. Det innebär att endast 58% ansåg att den var mycket viktig eller viktig. Vi drar slutsatsen att vi ska behålla handledning men att vi måste utveckla formerna för detta. Eftersom vi fortfarande tror att övningar och praktik är en förutsättning för att lära sig programmering, är det naturligt att handledda övningspass måste ta en stor plats i kursen. Övningar är bara användbara tills studenten kör fast och det är då handledning behövs. Här kan det vara på sin plats att också nämna att övningar ställer krav på studenten att inte bara lösa övningarna mekaniskt, utan också stanna upp och säkerställa att hon har förstått varför hon löste övningen på det sätt hon gjorde och huruvida det finns alternativa sätt att lösa uppgiften på. Det finns en risk annars att studenterna betar av övningarna (med hjälp av de tips och lösningsförslag som finns) utan att egentligen förstå vad det är man har gjort. Detta måste kommuniceras till studenterna, att det är deras ansvar att lära sig och att förstå och

att det är lärarnas ansvar att hjälpa dem nå målen och förståelsen men att detta kräver att studenterna reflekterar över sitt lärande och tar ansvar för att säga till när man inte förstår och då be om handledning.

När det gäller **handledning utanför schemat** exempelvis genom att knacka på hos lärarna utanför lektionstid, så uppger 28% att detta var mycket viktigt för deras lärande, 30% att det var viktigt och 25% att det var av medelbetydelse. Nästan 60% sätter alltså vikt vid detta, vilket väl är att vänta. De studenter som lägger ned mycket tid och energi på att klara av övningar och inlämningsuppgift blir naturligtvis frustrerade och ivriga att komma vidare, även om detta sker utanför schemalagd tid och då är det inte konstigt att de uppskattar att få komma och ställa frågor. Naturligtvis kan det bli mycket sådana frågor med en klass på över 100 studenter men vi har upplevt att det gått att hantera och varit mödan värt då vi tror att det är viktigt att behålla studenternas energi och vilja att lära sig.

På frågan om **inlämningsuppgiftens** betydelse för lärandet, uppger 28% att den var mycket viktig, 44% att den var viktig och 26% att betydelsen var "medel". Här kan man förstås läsa in att uppgiften var obligatorisk och betyggrundande, vilket kan öka känslan av uppgiftens betydelse men vi tror att den kanske just därför sporrat studenterna att klara av den och att förstå uppgiften. Vi har försökt lägga in moment i uppgiften som kompletterar målen för kursen utöver de mål som examineras i tentamen. Det finns naturligtvis ett visst överlapp, vilket är svårt att undvika i en programmeringskurs. Uppgiften har även frivilliga överkursmoment som inte examineras för de studenter som upplevde att uppgiften inte var tillräckligt utmanande. Vi har dock inte mätt hur många som gjort överkursdelarna, vilket vi bör göra i nästa iteration av kursen.

Vi har använt den inskickade koden för att få en bild av hur studenterna strukturerar kod stilmässigt och sett att här finns det utrymme för förbättring vad gäller indentering och placering av exempelvis "krullparenteser" med mera, vilket har föranlett ett antal extra övningar och en del nytt material i teoridelen som försöker sätta fokus på kodstil. Vi har även gått igenom vår exempelkod och lösningsförslagen utifrån kodstil och förbättrat vår egen kod för att den ska vara mer konsekvent för att underlätta för studenterna att få en intuitiv känsla för hur koden stilmässigt skall struktureras. Naturligtvis är detta viktigt men vi har funnit utrymme för förbättring och vi tror att detta varit en nödvändig åtgärd, då studenter förmodligen undermedvetet eller medvetet söker mönster när de läser kod. Även i tentamenssvaren fann vi att kodstilen är något vi i år misslyckats sätta vikt vid och inte riktigt lyckats lära ut.

Vi frågade även om betydelsen för **externa källor för vidareläsning som vi aktivt länkat** och uppmuntrat till läsning av. Endast 10% uppgav detta som mycket viktigt, 12% som viktigt och hela 58% som av medelbetydelse. Vi tycker detta är synd, då vi nogsamt valt ut källor som vi tycker är bra, korrekta och hjälpsamma. Varför detta inte gått fram kan säkert förklaras med att kursen har ett mycket högt tempo och är krävande i sig, så studenterna har säkert inte känt att de hunnit eller mäktat med ytterligare läsning. Vi sammanställde även alla länkar (128 stycken) i ett dokument med sökbara nyckelord för länkarna för att underlätta för studenterna att hitta länkarna. 25 studenter laddade ned detta dokument från GUL. Av de som laddat ned fanns det såväl studenter som fick U på tentan som studenter

som fick G eller VG, så det är svårt att läsa ut vilken betydelse länkarna haft. Länkarna tjänar två syften, som vi ser det. Dels är det något att fördjupa sig med om man är en intresserad och flitig student. Dels utgör de ett alternativ till vissa externa källor som vi sett användas men som inte håller måttet eller ligger för långt från vår förklaringsmodell för att studenterna ska ha nytta av dem. Vissa externa källor tror vi är rent av skadliga, då de verkar författade av personer som inte riktigt har på fötterna eller/och saknar pedagogisk förmåga.

Just för att vi sett vissa externa källor, som vi inte skulle kunna rekommendera till någon som vill lära sig programmering och nå målen för vår kurs, användas av studenter så ville vi mäta hur studenterna upplevt att **externt material de hittat själva** hjälpt dem förstå, så frågade vi även detta. 8% uppgav att sådana källor var mycket viktiga, 16% att de var viktiga och 40% att de var av medelbetydelse. Vår åtgärd kommer att bli att vi är mer tydliga när vi i början av kursen tar upp externa källor, såsom tutorialer och bloggar. Vi måste tydligare förmedla att vi inte sett att studenter som använt sig av sådana källor i högre utsträckning nått målen och därmed klarat examinationen. Faktum är att vi funnit vissa tecken på missuppfattningar och även kodstilar som återfinns i några av de källor vi vet att studenter använt. Det är svårt att leda i bevis att studenter skaffat sig olater just från dessa källor men blotta misstanken gör det värt att adressera detta tydligare. För att nå målen i vår kurs, så bör man fokusera på vårt kursmaterial och våra föreläsningar, eftersom vi designat dessa för att underlätta för studenterna att nå målen. Det är aldrig fel att komplettera med externt material men det är väldigt svårt att som nybörjare avgöra kvaliteten och sanningshalten i externa kurser eller källor. Att se tutorialer som en genväg eller alternativ till att gå kursen och använda kursens material kan vi med säkerhet säga inte kommer fungera. Det var av just dessa skäl som vi tog fram eget material samt kompletterade med av oss utvalda externa källor för fördjupning och vidareläsning.

Sista frågan om studenternas lärandematerial handlade om huruvida en **kursbok i pappersform** hade hjälpt dem i deras lärande. Anledningen till att vi frågade är naturligtvis att vi avskaffat pappersbok i kursen och ersatt den med egenproducerat material i digital form. Endast 2% tyckte att en sådan hade varit "mycket viktig", blott 10% "viktig", 14% "medel" medan hela 32% "oviktig" och en tydlig andel på 42% "irrelevant/helt oviktig". Här behövs ingen åtgärd då en klar majoritet inte ser behovet av en pappersbok och en försvinnande liten andel (2%) sätter detta som "mycket viktig". Vår erfarenhet av traditionella kursböcker är ganska negativ. Innan vi tog fram vårt eget material analyserade vi några av de mest använda kursböckerna för Java-undervisning. Vår bedömning var nedslående. Få böcker ringade in kursmålen väl. Antingen saknades vissa delar eller så fanns för mycket som låg utanför kursmålen. Vi såg även en tendens hos såväl oss själva som hos andra lärare att boken influerar upplägget och framförallt progressionen i för stor grad. Det finns en tendens att designa kurser utifrån kapitlen i kursboken och även låta övningarna i kursboken vara det enda studenterna får. Vår bedömning var att kursböckerna inte hade en pedagogiskt genomtänkt ordning i vilken koncept togs upp i kapitlen. Övningarna, och i många fall även lösningsförslagen i den mån sådana fanns var enligt vår uppfattning inte av god kvalitet (vissa lösningsförslag var felaktiga rent av) eller pedagogiskt uppbyggda. Då vi inte hittade en bok som inte led av några problem utifrån de läromål kursen har, eller som

färgats av den syn på pedagogik i programmeringskurser på introduktionsnivå, såg vi ingen annan lösning än att skriva eget material och hålla materialet levande och ständigt underhållet.

Vad gäller vårt material för just denna kurs, TIG015, så består det av:

- 64 föreläsningpresentationer på sammanlagt 747 sidor (PDF-format)
- Föreläsningar på video
 - 47 på svenska
 - 61 på engelska
- Praktiska programmeringsexempel på video, "livekodning"
 - 13 på svenska
 - 25 på engelska
- 402 övningsuppgifter/frågor

Samtligt material är öppet tillgängligt på webben, utan inloggning.

Ordinarie kursvärdering

Svarsfrekvens: 53 av 104 (50%).

Jag har under kursens gång känt till och förstått kursens lärandemål.

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	0%	0%	32,1%	67,9%
Antal	0	0	17	36

Medelvärde (för siffer-svar): 3,68

Jag har fått tillräcklig information gällande kursens innehåll och upplägg för att kunna göra mitt bästa för att leva upp till kursens lärandemål.

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	1,9%	1,9%	30,2%	66%
Antal	1	1	16	35

Medelvärde (för siffer-svar): 3,6

Kursens undervisningsformer (som t ex föreläsningar, seminarier och övningar) har varit relevanta för kursens lärandemål.

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	0%	7,5%	24,5%	67,9%
Antal	0	4	13	36

Medelvärde (för siffer-svar): 3,6

Respondentkommentarer:

Många av föreläsningarna var mer förvirrande än vad dom hjälpte. Även om man kollat på filmer sedan innan och läst på var det svårt att hänga med då många av de exempel som tog upp på föreläsningarna ofta var för svåra att förstå sig på, undantag eller ett sidospår till något annat. Önskar mer liveprogrammering i ett lugnare tempo så man kan skaffa sig en uppfattning om vad det är man lär sig och framför allt hur det kan användas.

Lite väl mycket och lång introduktion och inledning på vad det är vi ska förstå, istället för att visa med enkla exemplar

Lärarnas engagemang har varit otroligt. Deras juneweekday sida har varit viktigast för mitt lärande.

Övningar under kursens gång har varit avgörande för att lyckas i kursen. Väldigt bra!

Tydliga föreläsningar som la grund för veckans uppgifter

Övningarna var väldigt relevanta. Hade velat ha ännu mer liveprogrammering under föreläsningarna för att kunna ta in alla koncept ännu bättre.

Programmerat mycket, skulle dock behövt lite mer live-prog för att (för mig) förstå det vi gjort bättre!

Annars hur bra som helst

Det var väldigt bra att genomgångarna (programmering) fanns som videos på internet så att man alltid kunde kolla om man missat något eller bara ville gå tillbaka. Ingen idé att maila så skriver det här. Hatten av Rikard och Henrik för att ni orkar stå ut med alla dumma frågor och för att ni är chefer typ.

Vi behöver mer föreläsningar. Det blir inte riktigt bra när ni måste stressa igenom dem och inte hinner med allt. Jag hade mycket hellre haft ordentliga genomgångar till alla lektioner istället för att sitta hemma och slita mitt hår över att jag inte förstår det ni säger på videon eller fastna i en negativ spiral så fort man missar något.

Verkligen bra med filmer att kolla på innan föreläsning för att sedan upprepa på föreläsning. Det satte sig bättre på så vis

Vad är ditt allmänna omdöme om relevans och innehåll i föreläsningar, seminarier, övningar/laborationer, litteratur och examination?

Föreläsningar

Alternativ	Inte bra	Mindre bra	Bra	Mycket bra
Fördelning	0%	11,3%	41,5%	47,2%
Antal	0	6	22	25

Seminarier

Alternativ	Inte bra	Mindre bra	Bra	Mycket bra
Fördelning	3,8%	1,9%	45,3%	37,7%
Antal	2	1	24	20

Övningar/Laborationer

Alternativ	Inte bra	Mindre bra	Bra	Mycket bra
Fördelning	0%	9,4%	37,7%	50,9%
Antal	0	5	20	27

Litteratur (som t ex böcker, artiklar, filmer etc)

Alternativ	Inte bra	Mindre bra	Bra	Mycket bra
Fördelning	1,9%	3,8%	32,1%	60,4%
Antal	1	2	17	32

Examination

Alternativ	Inte bra	Mindre bra	Bra	Mycket bra
Fördelning	0%	3,8%	49,1%	45,3%
Antal	0	2	26	24

Vad tycker du om kursens svårighetsgrad?

Föreläsningar

Alternativ	För låg	Lagom	För hög
Fördelning	5,7%	88,7%	5,7%
Antal	3	47	3

Seminarier

Alternativ	För låg	Lagom	För hög
Fördelning	0%	92,5%	0%
Antal	0	49	0

Övningar/Laborationer

Alternativ	För låg	Lagom	För hög
Fördelning	5,7%	86,8%	5,7%
Antal	3	46	3

Litteratur (som t ex böcker, artiklar, filmer etc)

Alternativ	För låg	Lagom	För hög
Fördelning	1,9%	83%	13,2%
Antal	1	44	7

Examination

Alternativ	För låg	Lagom	För hög
Fördelning	3,8%	81,1%	11,3%
Antal	2	43	6

Det var tydligt vad som förväntades av mig i kursens examinerande moment (till exempel tentor, inlämningsuppgifter och projekt).

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	1,9%	3,8%	28,3%	66%
Antal	1	2	15	35

Medelvärde (för siffer-svar): 3,58

Jag har enligt min egen bedömning lagt tillräckligt med tid och engagemang på kursen för att klara av den.

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	0%	5,8%	42,3%	51,9%
Antal	0	3	22	27

Medelvärde (för siffer-svar): 3,46

Kursen har gett mig kunskaper och/eller färdigheter som jag tror jag kommer att ha nytta av i mitt framtida yrkesliv.

1 - Jag håller inte alls med.

4 - Jag håller med helt och hållet.

Alternativ	1	2	3	4
Fördelning	0%	0%	24,5%	75,5%
Antal	0	0	13	40

Medelvärde (för siffer-svar): 3,75

Fritextsvar:

Bra upplägg på tentan med tydliga rubriker om vad som faktiskt efterfrågades.

Kanske lite enklare uppgifter i början. Ex weekly inlämningsuppgiften kan innehålla lite mer basic uppgifter.

Föreläsningarna borde vara lite längre. Känns oftast stressat för att man har för lite tid.

Det har varit en väldigt intensivkurs. Jag har verkligen velat lära mig och lagt ner mycket tid på att försöka lära mig. Trots detta känner jag att jag inte hängt med. Jag tror inte det egentligen beror på undervisningen i sig, utan snarare på att kursen känns ganska ihoptryckt, det är väldigt mycket information på väldigt kort tid. Detta är inte rättvist för varken eleverna eller lärarna. Extra svårt är det om man är nybörjare och inte har någon kunskap eller erfarenhet sen tidigare. Undervisningen har varit bra och lärarna har varit hjälpsamma. För mig har det dock varit en svårighet att formulera en fråga, mer än att uttrycka att jag inte förstår, eftersom mycket av den informationen man tagit del av varit väldigt abstrakt och det har därför varit svårt att sälla. Examinationen var svårare än vad jag förväntat mig, som ändå gått igenom majoriteten av gamla tentor. Jag blev också överraskad över att comparator kom med i tentan, eftersom det nämns direkt i live videos att "detta inte skulle komma på tentan". Jag tycker det var aningen missvisande. Sedan är jag medveten om att det är något vi gått igenom under lektionstid. Men om man pluggar inför en tenta, pluggar man på saker som förväntas komma, inte på all information vi fått under kursens gång.

Lite stressade föreläsningar, för lite tid för att hinna klart med alla punkter men filmerna var grymma. Riktigt bra kurs.

Rolig och nyttig kurs. Vill ha fler av dessa istället för de teoretiska bussiness-kurserna som känns mer flummiga och irrelevanta för arbetsmarknaden.

Tyckte det var en väldigt rolig kurs. Bra nivå av både svårighetsgrad och humor vilket gjorde att det var kul att lära sig. Wikin och videos var kanon men inför tentan gjorde det också att det kändes som att det var VÄLDIGT mycket information att ta in och det var lite svårt att klura ut vad man skulle sortera ut. Tentaplan var väldigt bra att ni la upp tyckte jag eftersom det gjorde aoeteringen enklare.

Examinationen, den skriftliga tentamen, var missvisande i vissa frågor. Trots god ansträngning under hela kursens gång var det i vissa frågor svårt att förstå meningen och vad som skall besvaras. Efter att ha diskuterat i efterhand om dessa frågor var det inte fel/avsaknad på kunskapen utan förståelsen av frågan. Under passets gång räckte jag upp handen för att få frågan beskriven på ett annat sätt, vilket jag inte fick. Därav fick jag ingen rättvis möjlighet att besvara frågan. Övningstentor, eller gamla tentor har inte varit utformade på detta sätt så det känns väldigt tråkigt att det var sådan stor skillnad i svårighet på frågans utformande.

Gillade kursen som helhet då jag tyckte programmeringen var rolig och skulle därför vilja ha mer av den. Föreläsningarna var givande men spårade ur lite i fråga om svårighetsgrad ibland, annars var dem relevanta för övningarna.

Kursdelen var guld! Rolig och intressant.

Kurslitteratur:

För att kunna dra ner lite på skärmtid vore det trevligt med någon form av pappers kompendier eller bok.

Jättebra föreläsningar men när ni ger exempel på olika saker för att var så konkreta som möjligt, t.ex. med att bananskal och maneter båda är "halkbara". Så gav det bra förståelse för vad det är men det hade varit uppskattat med ett exempel som man kommer att se i koden också.

Ha gärna mer live kodning i era föreläsningar. Det var mer text än faktisk kod och det var ibland svårt att förstå hur man skulle implementera vissa saker eftersom man väldigt sällan fick en helhetsbild. Visa gärna mer helhetsbilder så att man lär sig vart man "brukar" placera olika delar i kod.

Rolig och intresant kurs!

Kursens svårighetsgrad - tyckte att kursen var lite för enkelt.

När det gäller tentor, tycker jag att man måste ha lite svårare frågor, eftersom det skulle vara rimligt om max 10% av studenten kunde få VG (annars VG förlorar sitt värde).

Addressbook inlämningsövning - det är egentligen ingen inlämningsövning om man kan gå till lärare/studenthandledare och be de ge rätt svar när saker inte fungerar. Det var också för enkelt för att fungera som inlämningsuppgift.

Överallt tyckte jag att litteratur var ganska bra och täckte mesta frågor. Skulle varit härligt att ha mer övningar (speciellt när det gällde anonymous class, interfaces osv).

Fråga 4; Övningar/laborationer.

I vissa av övningarna används termer, metoder, mm. som vi inte har använt oss av/gått igenom tidigare. Även om det ofta finns en förklarande text till dessa termer / metoder, så blir det i vissa fall rörigt. Om syftet med detta är att "utsätta" oss för delar som kommer senare i kursen hade det varit bra om vi, åtminstone i övningarna, hållit oss till termer/metoder/mm. som vi redan känner till.

Jag tycker det har varit en bra kurs, och har uppskattat föreläsningarna och Wikin för att lära mig. En sak som jag speciellt gillat är kombinationen av föreläsningar, wikin (se filmer och läsa) för då har man upprepade gånger fått ta del av informationen och på så sätt har mycket fastnat.

Även bra med handledning och att lärarna har tagit sig tid att verkligen få oss att förstå. Det har märkts väldigt tydligt att de vill att vi skall lära oss. Även bra och förstående stämning i klassrummet som gjort att man vågat ställa frågor!

Henrik och Rikard förtjänar påökt (i lönekuvertet alltså).

Pedagogiska, sakliga och "egna" lärare med humor. Det enda som var lite för svårt (för egen del) var inlämningsuppgiften som jag inte skulle klarat helt själv utan klasskamrater. Kanske dela in 2&2 till nästa kull?

Återigen. Stort tack, ni är fan genuina.

Mycket bra pedagogisk lärare!

Jag tycker att föreläsningarna har varit bra, jag skulle gärna se mer av dem. Det finns mycket material på wikin och det är lätt att bli förvirrad, dokumentet på gul med alla checkpoints har varit ovärderligt.Handledningen har varit toppen, förutom att det verkligen underlättat att klara kursen så har det fungerat som en brygga för att lära känna elever från högre terminer.

Jag tycker om att föreläsarna visar stor entusiasm och uppmuntrar oss att ta reda på och testa saker på egen hand, dock skulle jag vilja att de då och då håller sig något mer till saken när de t.ex. svarar på frågor.

Generellt är övningarna i wikin uppbyggda så att man stöter på alla fel först och sedan får fram det som är rätt, vilket gör att jag som elev i värsta fall känner mig dum och blir frustrerad över att jag måste vänta in rätt svar och i bästa fall förvirrar det mig bara. Tala om det som är rätt först och det som är fel sedan.

Angående tentamen var den lite knepig. Först och främst skulle jag verkligen vilja ifrågasätta varför vi behövde fylla i så mycket uppgifter på framsidan av tentamen. Jag har studerat vid GU tidigare och har aldrig varit med om att jag behövt sitta och skriva ner så mycket administrativa uppgifter om mig, mitt program och min kurs. Dessa har, förutom mitt namn, alltid varit färdigtryckta på framsidan av tentamen som förövrigt haft frågor och svar i samma häfte istället för en bunt med lösblad som lätt tappas bort. Det hela skapar ett extra stressmoment för eleverna. Det var dessutom märkligt att det stod Chalmers med stora bokstäver på framsidan.

Vidare tycker jag att det var för ont om tid vid tentamen, det var tråkigt att behöva lämna in sin tenta utan att ha hunnit svara på två av uppgifterna speciellt då det kan avgöra betyget. Det tar betydligt längre tid att skriva ner sin kod för hand än på datorn och det hade inte varit fel att få en extra timme så att man slapp känna sig stressad.

Jag gillade formen på föreläsningarna och videorna men hade nog önskat lite mer genomgång av kodning för att få helheten.

Ang. Examinationens svårighet så var den för mig för hög då jag tyvärr inte förberedde mig tillräckligt. Med tillräcklig förberedelse så var den nog absolut i samklang med det vi gått igenom.

Insåg lite för sent att ju inte hade helheten för att klara examinationen bra och hade själv behövt fråna och diskutera mer med klasskamrater och lärare under kursens gång.

Mycket bra introduktion till Programmering för någon som aldrig programmerat innan dock med väldigt engagerade lärare som inspirerar till att fortsätta med detta.

Styrkor: Engagerade lärare, väldigt kompetenta, tillgängliga. Har en väldig höggrad av vilja till att vi ska lära oss att programmera -BRA. Svagheter: Upplever det svårt att få svar från lärare på en specifik fråga som man har, det svävar väldigt ofta ut, stundtals med relevant information, men oftast med information som är helt irrelevant. Vilket leder till att jag upplever att det är otydlighet i vad som var rätt. Upplever de svårt att följa och få en överblick i kurslitteraturen - rörig oinspirerande.

Det är mkt som händer på kort tid och ibland så hade det varit nice med lite fler lärarledda timmar.

Vissa övningar (på Wikin) har varit väldigt svåra, medan andra har varit lättare. Jag uppskattar att ni har tydliggjort vad som är viktigt att göra och vad som är lite övertkurs för tillfället. Weeklys var svåra enligt mig.

Kommentarer till ordinarie kursvärdering

Vad gäller svaren med sammanställning av siffrvärden ser vi inget som sticker ut nämnvärt. Kursen verkar upplevts relevant och lagom svår.

Från fritextsvaren tar vi till oss att även här efterfrågar några (fyra kommentarer) mer livekodning/liveprogrammering. Wikin verkar uppskattas men några tycker den var för svår, andra för lätt. Spridda skurar således.

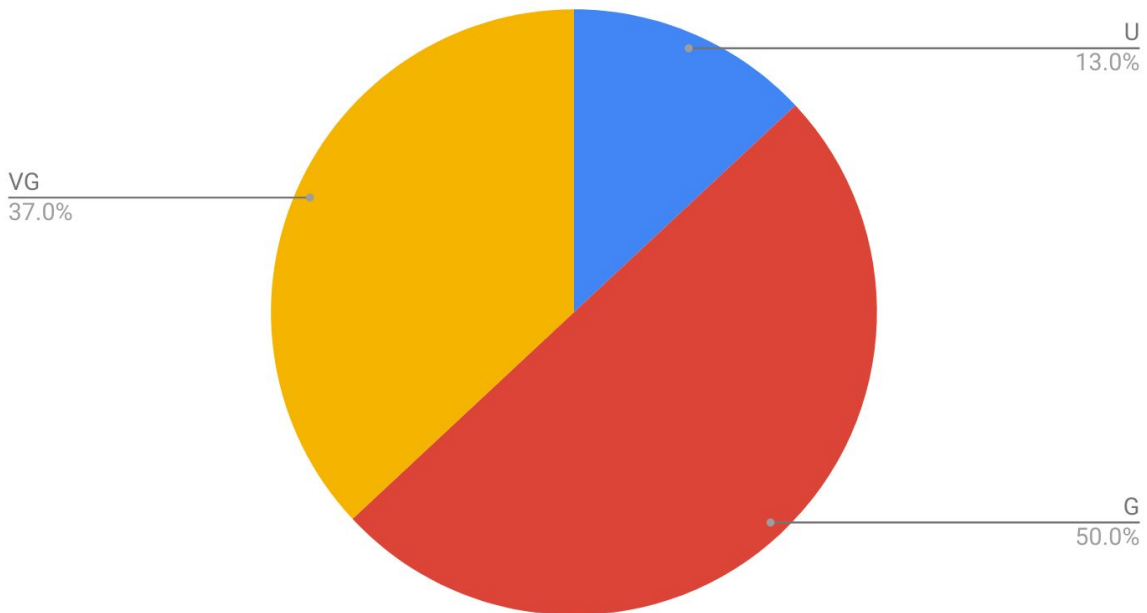
Vi ser inget alarmerande i resultatet.

Tentamensresultat

92 av 103 studenter gick upp på tentan. Av de skrivande blev betygsfördelningen som följer:

U	12	13.04%
G	46	50.00%
VG	34	36.96%

Betygsfördelning



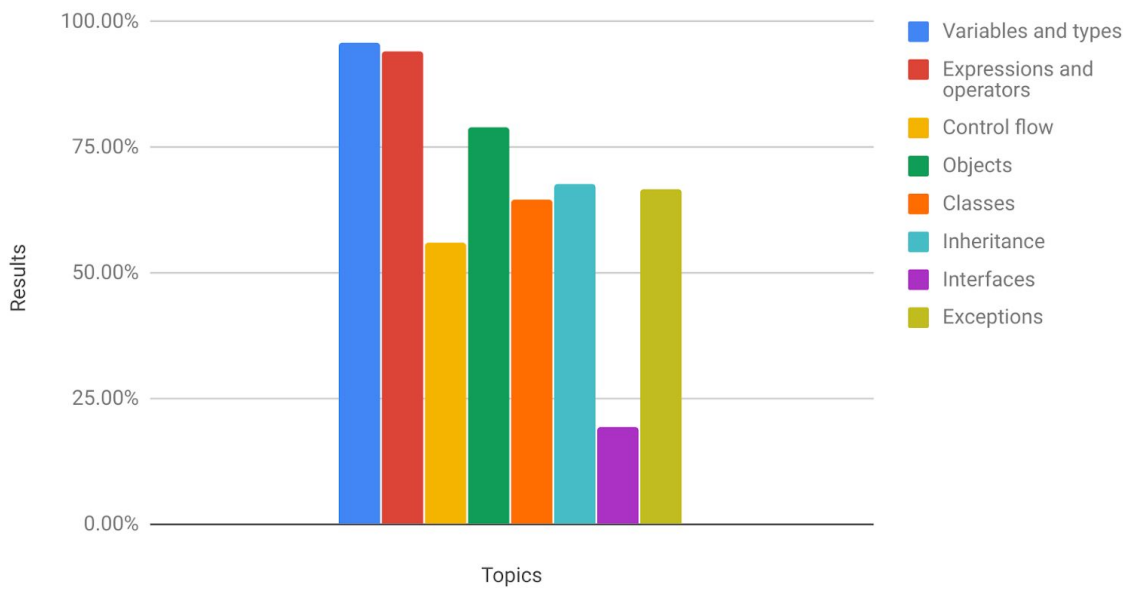
Vi tycker det är glädjande att så många gick upp och skrev och naturligtvis att endast 12 blev underkända. Andelen VG var också glädjande. Vi gjorde om tentans struktur i år, så att det skulle vara svårare att få G och VG genom att endast ha rätt på första delen. Detta försökte vi uppnå genom att varje avsnitt i tentan hade en svårare fråga, så att det inte skulle finnas avsnitt som upplevdes som "lättare" än andra avsnitt så tillvida att avsnittet enbart hade lättare frågor.

Tentan hade 8 avsnitt:

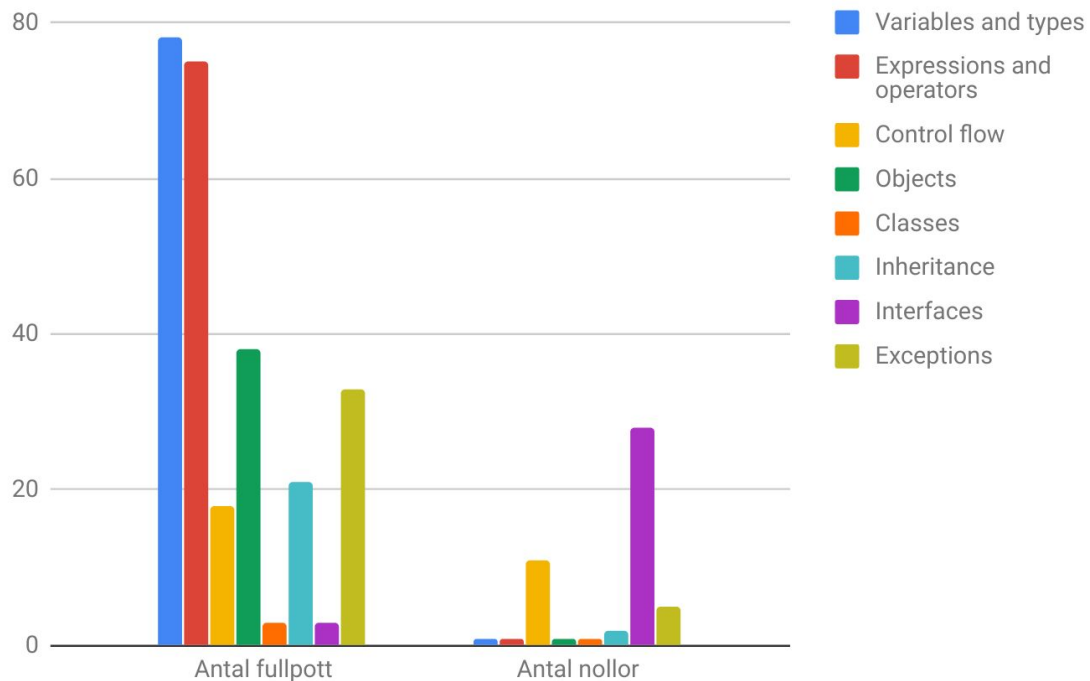
- Variables and types 4p
- Expressions and operators 4p
- Control flow 4p
- Objects 6p
- Classes 16p
- Inheritance 6p
- Interfaces 6p
- Exceptions 4p

Traditionellt sett, så brukar de tre sista avsnitten, Inheritance, Interfaces och Exceptions vara de svagaste områdena hos studenterna, vad gäller tentaresultat.

Årets resultat blev:



I år sticker som synes Interfaces ut som det i särklass svagaste avsnittet på tentan. Om vi tittar på hur många som svarade med full poäng, respektive noll poäng per avsnitt får vi följande bild:



Även här sticker Interfaces ut med hela 25% (28 studenter) som nollat hela avsnittet. 9 av de 28 som nollat avsnittet hade U på hela tentan. 17 av dem som nollat avsnittet hade alltså G eller VG. Tre studenter hade alla rätt på avsnittet.

Något annat som sticker ut i år, är att det resultatmässigt näst svagaste avsnittet var Control Flow. Det skiljer sig från våra analyser av tidigare år. Glädjande är att avsnitten om Classes, Inheritance och Exceptions gav mycket bättre resultat i år.

Det som förvånar oss mest är att så många fick noll poäng på avsnittet om Interfaces. Under de tre repetitionspassen veckan före tentan så togs ämnet upp åtminstone vid två tillfällen. Första passet närvarade 50 studenter. Pass två och tre var det betydligt färre som kom.

Utöver detta, så gav vi ut extramaterial på ämnet i form av en PDF som publicerades i GUL, och ett långt avsnitt i vår "Java FAQ" på Wikin skrevs med fokus på samma problematik som avsnittet om Interfaces berörde (att implementera interface:et Comparator). Interfacet var även med i tentans appendix. Vi får ägna mer tid och att liveprogrammera kring Comparable och Comparator i nästa iteration (trots att live-videor, och gott om exempelkod redan finns).

Det som förvånar oss näst mest är att något så grundläggande som Control Flow fick så klent resultat. Endast 18 studenter fick alla rätt (4 poäng av tentans 50) på detta avsnitt. Hela 11 studenter nollade avsnittet, vilket får anses som ovanligt. Sju av de 11 fick U på hela tentan, vilket betyder att fyra studenter med G eller mer på tentan nollade ett så grundläggande avsnitt. I snitt fick de skrivande 2,23 poäng av 4 på avsnittet.

Här får vår åtgärd bli att utöka antalet frågor och övningsuppgifter för just Control Flow. Även liveprogrammering kan sättas in här. Om vi ska experimentera med att öka andelen liveprogrammering, så verkar det rimligt att sätta in denna åtgärd där tentaresultatet var svagt. Vi får även förstärka andra avsnitt i kursmaterialet så att det krävs och övas på Control Flow även där.

Iakttagelser från rättning av tentamen

Många svar hade liknande formuleringar men var felaktiga. Detta kan tyda på att man studerat ihop (något vi aktivt uppmuntrar!) men att någon missuppfattning spridit sig genom klassen eller gruppen. Vi vet inte hur vi ska hantera detta. En del svar verkade inlärda snarare än byggde på "egna" ord eller egna beskrivningar. Detta måste vi aktivt motarbeta - programmering är inget ämne man kan lära sig utantill.

En annan iakttagelse från rättningen var att uppgifter där studenterna skriver kod hade en del egenheter som inte kommer från vårt kursmaterial eller våra föreläsningar. Ett exempel på det var att en påtagligt stor del av studenterna valde svenska variabelnamn när de skrev kod själva. Detta är något vi aldrig gör eller har med i vårt material. Faktum är att vi aktivt avråder från detta och har så gjort ett flertal tillfällen. Förhoppningsvis så gjorde några av studenterna detta för att reta oss, vilket i så fall tyder på en sund glimt i ögat. Men vi misstänker att de flesta inte gjorde detta för att provocera.

En annan iakttagelse rör en fråga där studenterna skulle skriva en klass för produkter i en webbshop, Product. En Product skulle ha bland annat pris som instansvariabel (varje produkt skulle ha sitt eget pris) och moms som en gemensam konstant (alla produkter skulle dela på - ha samma - moms). Ändå såg vi många märkliga konstruktioner där moms lagrades i märkliga former, bland annat använde många procent-tecken som om det stod för procent (i Java är det en operator för rest vid heltalsdivision). Andra använde String som typ

för moms, och ytterligare andra försökte räkna ut den gemensamma moms som en funktion av en individuell produkts pris. Påfallande många hade med moms som en parameter till konstruktorn, trots att de lyckats ange att moms var en klassvariabel och inte beroende av individuella instanser.

Slutsatsen vi kan dra av detta är att studenterna inte riktigt förstått modifieraren `static` och inte heller hur man räknar med procent i ett Java-program eller i en Java-klass.

Vi får utöka övningarna som rör `static`-variabler och -konstanter, samt hitta på några exempel och övningar som har med procent att göra.

Språkmässigt (det är inget vi bedömer men kunde inte undgå att lägga märke till) så verkar det vanligaste språkfelet vara att studenterna använder "medans" i stället för "medan". Det är talspråk och ingen stor sak, men det var väldigt vanligt.

Avsnittet som rörde Control Flow, avslöjade att många trodde att det som avgör ifall man ska använda en `for`-loop eller en `while`-loop var huruvida man ville att loop-variabeln skulle vara i scope efter loopen eller inte. Detta förbryllade oss. Huruvida loopvariabeln är i scope eller inte efter loopen, är en följd av hur `for` respektive `while` fungerar syntaktiskt, inte något man eftersträvar eller inte. Vi antar att vi gav detta för stort utrymme i föreläsningarna och övningarna om loopar, i stället för att ta upp det i föreläsningarna om block och scope.

En annan detalj som var intressant, var att många gav exempel på `for`-loopar som var korrekta men något otypiska. Looparna räknade ned mot noll, i stället som det vanligare att de börjar på noll och räknar upp mot ett randvärde.

Detta kan bero på att vi visade ett antal `while`-loopar som räknade ned och sedan översatte detta till `for`-loopar som räknade åt samma håll. Detta gjorde vi för att visa hur man går från en `while`-loop till en `for`-loop för att visa på strukturella likheter.

Vi får trycka hårdare på olika idiom (t ex `for`-loopar från 0 till ett randvärde) och försöka använda mer typisk kod i våra exempel. Det är ingen stor sak, men värt att adressera.

När det kom till operatörer, så hade förvånansvärt många studenter problem att beskriva `++`-operatören. Många kopplade ihop den med loopar (där den används men det är inte loopar den är till för). Andra beskrev den som att "värdet ökar med ett". Värdet på vad då? skrev vi i kommentarer på tentorna (även om vi ofta gav rätt om exemplen visade på korrekt användande). Det verkar som om studenterna antingen inte förstått vad operatören gör, eller helt enkelt har svårt att sätta ord på detta. Det är ju en variabls värde som ökar med ett men nästan ingen sade detta. Vi misstänker att man studerat ihop och bestämt sig i gruppen för att en korrekt beskrivning är "den gör att värdet ökar med ett".

Vissa studenter visade att de inte förstod vare sig hur literaler eller `++`-operatören fungerade och gav exempel så som: "3++". Detta vet vi inte vad det beror på.

När det gäller avsnittet om Exceptions, så var resultatet betydligt bättre i år, än det var tidigare år. Men några saker återkom i studenternas svar. När de skulle skriva kod som kastade ett `NullPointerException`, så glömde påfallande många parenteserna efter klassnamnet, även de som kom ihåg att skriva "new `NullPointerException`". Detta visar möjligen på att man studerat in kod igen och koden man lärt sig var felaktig, eller, troligare, på att man inte riktigt förstått att ett exception som kastas är en referens till ett nyskapat objekt. Det kan också visa på ovana att skapa nya objekt över lag, men vi såg det mest i samband med att detta exception skulle kastas.

En formulering som också återkom och som förvånade oss, var när studenterna skulle beskriva hur ett uttryck med en jämförelse evaluerades.

Man beskrev uttryck så som " $a < 2$ " så som att jämförelseoperatören "<", mindre än "returnerade" true eller false. Ingen stor sak, men det var väldigt många som uttryckte sig på detta olyckliga sätt. Vi satte inte fel på svaret på grund av detta men känner att vi måste utreda varifrån denna formulering kommer.

Vi har hört denna formulering från lärare och även sett den i skrift. Men vi har inte hört våra studenter. Vi får försöka bli tydligare med hur uttryck *evalueras* och vad *returnera* betyder och i vilka sammanhang något returneras.

En detalj till som vi såg väldigt många studenter skriva var användandet av kommatecken som decimalavskiljare för flyttalslitteraler. Detta är inget vi satte fel för och är ganska naturligt då vi i Sverige använder just kommatecken och inte punkt som i engelskan (och i Java).

Förklaringen är nog dessvärre också att vi haft för få exempel och övningar med flyttal. Detta bör vi åtgärda av fler skäl än att studenterna ska kunna skilja på komma och punkt.

När det kom till studenternas kod i de uppgifter där de skulle skriva sådan, så var det vissa återkommande stilproblem som även berörts tidigare. Placeringen av krullparenteser var oftare fel (eller inkonsekvent) än rätt. Samma gäller indenteringen, vilket också berörts tidigare i denna analys. Här har vi redan börjat införa åtgärder, vilket också nämnts i tidigare avsnitt ovan.

Slutligen så noterar vi att de flesta studenter på frågan om vad man kan använda en for-loop till (de anmodades ge exempel på en användning), svarade att for-loopen används till att loopa igenom en array eller en lista. Det är inte direkt fel, men snarare så skulle man nog säga tvärt om: "För att loopa igenom en lista eller array, så kan man lämpligen använda en for-loop". På frågan vad man kan använda en for-loop till i allmänhet, så finns det betydligt mer generella svar att ge där genomlöpning av listor endast är ett exempel.

Iakttagelser vid rättning av Inlämningsuppgiften

Indenteringen verkar helt enkelt inte gått hem. Detta trots att vi vid upprepade tillfällen visat dem att de kan få automatindentering av sin editor och även påpekat att vi har svårt att läsa deras kod om de inte lär sig indentera. Här kommer ett exempel ur mängden:

```
public Contact(String name, String email, String phone){
    if(name == null){
        throw new NullPointerException ("this name is not valid");
    }
    if(name.equals("")){
        throw new IllegalArgumentException("You need a name");
    }
    else{
        this.name = name;
    }

    if(email == null){
        this.email = "";
    }
    else{
        this.email = email;
    }
    if(phone == null){
        this.phone = "";
    }
    else{
        this.phone = phone;
    }
}
```

Vi har redan flera gånger tidigare i denna analys redogjort för åtgärder vi redan påbörjat vad gäller indentering och kodstil.

Med vänlig hälsning från lärarna i programmeringsdelen av kursen.

Henrik Sandklef och Rikard Fröberg, november 2018.