

Sensorveiledning

IN1000 høst 2023 (konteeksamen januar 2024)

Seksjon 1-2

Totalt 17 poeng (autorettes).

Felles for seksjon 3, 4, 5

(nytt) Dersom studentene har misforstått oppgaveteksten slik at de ulike punktene for poeng (i løsningsforslaget) ikke passer, så kan man på skjønn gi en del uttelling dersom studenten likevel viser tilsvarende kompetanse. Det bør likevel være litt/noe trekk for manglende forståelse av selve oppgaveteksten i slike tilfeller.

Små skrive/slurvefeil overses (eks: mangler _ i et instansvariabelnavn eller en self-parameter), men gjennomgående feil som viser manglende forståelse skal gi trekk (ingen instansvariabler markert som non-public, alle/ de fleste metoder mangler self-parameter). Manglende lesbarhet gir trekk der det er uklart hva studenten faktisk kan.

Læringsmål/ aspekter i seksjon 3 og 5

I oppgave 3 og 5 vurderes spesielt de siste 3 punktene fra [Læringsmålene for IN1000](#) (første punkt testes hovedsakelig i oppgave 4). Poeng i deloppgavene på 3 og 5 fordeles på følgende aspekter i henhold til tabellen nedenfor:

- **PRO**: Løse PROblemer i python (fornuftig logikk/ fremgangsmåte) og
- **PRE**: PREsis implementasjon av fremgangsmåten
- **MOD**: sette seg inn i programmer, finne feil, MODifisere dem

Oversikt seksjon 3: Poeng fordelt på deloppgave og kategori

LUB/ deloppgave	Løse PROblemer i python	PREsis implementasjon av fremgangsmåten	sette seg inn i og MODifisere programmer	Sum per oppgave
3a	3	2		5
3b	3	2		5
3c	4	2		6
3d	5	2		7
5	7	3		10
Sum per kriterium	22	11	0	33

Læringsmål/ aspekter i seksjon 4

Det er definert 3 aspekter på tvers av deloppgavene i oppgave 4. Løsningsforslaget angir hvordan poeng for hver deloppgave fordeles på disse. Noen eksempler på feil og mangler som gir trekk er gitt - bruk skjønn basert på disse.

Klassedefinisjon med konstruktør og metode («Definisjon»)

Definerer og initialiserer instansvariabler (inkl lister og ordbøker) som non-public med `_`, har med parameter `self`. Håndterer initialisering av instansvariabler både direkte fra parametere og konstanter, og ved hjelp av operasjoner som for eksempel å bygge opp en ordbok (*Hvordan* ordboken bygges opp vurderes under problemløsning og/ eller OO).

Definerer metoder og bruker parametere og returverdier riktig og på rett sted, inkl `self`. Angir eventuelle interne hjelpemetoder som non-public med `_`. `__str__`-metoder returnerer en streng.

Forstår prinsippene for objektorientert programmering og kan benytte disse til å skrive enklere objektorienterte programmer («OO»)

Bygger opp og benytter datastrukturen slik at den virker iht. oppgaven. Oppretter objekter på rett sted med argumenter av rett type, setter objekter riktig inn i datastrukturen. Skiller på strenger og referanser til Kategori-objekter. Kaller korrekt på eksisterende metoder i andre objekter når dette er hensiktsmessig (også dersom disse ikke er implementert). Bevarer innkapsling.

Problemløsning («Problem»)

Finner fremgangsmåter for å løse større og mindre algoritmiske utfordringer i en objektorientert system, og implementerer disse korrekt i Python. For eksempel oppbygging, aksess og traversering av lister og ordbøker, manipulering av indekser. Streng-manipulering og konvertering mellom typer. Riktige returverdier i ulike situasjoner. Trekk for feil og mangler som ikke faller inn under Def eller OO, men gjør at oppgaven ikke er fullstendig/ korrekt løst, kan foretas her.

Oversikt oppgave 4: Poeng fordelt på deloppgave og aspekt

Kort oppsummert: Den poengsummen en kandidat får på en deloppgave uttrykker hvor godt oppgaven er løst. Poengsum per aspekt på tvers av deloppgavene brukes i begrunnelser for å indikere sterke og svake sider hos kandidaten.

Oppg/ LUB	DEFinere klasser	OO - Objektorientert programmering	PROblemløsning i et større (oo) system	Sum per oppgave
4a	4		2	6
4b	3	4	3	10
4c	4	2	2	8
4d	4	2	4	10
4e		3	3	6
4f		10		10
Sum aspekt	14	22	14	50