

Sensurveiledning

Vurdering og individuell begrunnelse

Besvarelsen vurderes til Bestått eller Ikke bestått. Del 1&2 (samlet), del 3 og del 4 må bestås hver for seg for at totalresultatet skal være Bestått. Individuell, automatisk generert begrunnelse sendes til alle kandidater per epost innen 3 uker fra sensur foreligger.

Seksjon 1 og 2 (automatisk rettet)

Seksjonen tester hovedsakelig læringsmålet: «.. er du i stand til å sette deg inn i andres programmer, finne eventuelle feil i dem og modifisere dem». Enkelte deloppgaver er blitt rettet feil i automatretting. Der kandidaten ligger an til stryk på del 1&2 er disse rettet opp manuelt for å sikre korrekt resultat. Krav til bestått: Mer enn 40% av oppnåelige poeng.

Seksjon 3

Seksjonen tester hovedsakelig læringsmålet: «.. kan du programmere i programmeringsspråket Python og kan bruke dette til å løse mindre problemer ved hjelp av valg, løkker, funksjoner, lister, klasser og objekter». For detaljering av poenggivning, se løsningsforslaget for del 3. Krav til bestått: Mer enn 40% av oppnåelige poeng.

Seksjon 4

Seksjonen tester hovedsakelig læringsmålet: «.. forstår du prinsippene for objektorientert programmering og kan benytte disse til å skrive enklere objektorienterte programmer». Krav til full uttelling oppgis nedenfor og vurderes skjønnsmessig på tvers av del 4. Avvik fra oppgaven som begrunnes og løses korrekt gir ikke nødvendigvis trekk. Krav til bestått: Mer enn 35% av oppnåelige poeng.

Konstruktør

Definerer og initialiserer instansvariabler (inkludert 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).

Metode-definisjoner inkludert `__str__`

Definerer metoder og bruker parametere, instansvariabler og returverdier korrekt. Angir eventuelle interne hjelpemetoder som non-public med `_`. `__str__`-metoder returnerer en streng.

Oppbygging og bruk av en objektorientert datastruktur

Bygger opp og benytter datastrukturen slik at den virker i henhold til oppgaven. Oppretter objekter på rett sted med argumenter av rett type, setter objekter riktig inn i datastrukturen, kaller eksisterende metoder i andre objekter når dette er naturlig, inkludert egendefinerte `__str__`-metoder for konvertering til string. Bevarer innkapsling. Onske- og Gave-klassen brukes (kun) i henholdsvis Onskeliste og Julegavekalender/ Juleferiekalender.

Problemløsning i Python i et objektorientert program

Finner fremgangsmåter for å løse større og mindre algoritmiske utfordringer, og implementerer disse korrekt i Python. For eksempel oppbygging, aksess og traversering av lister og ordbøker, manipulering av indekser. Lesing fra fil, streng-manipulering og konvertering mellom typer.