Oblig4\_Sindre Flatin

Teorioppgave 1. -Exception:

En exception gir muligheten til å bestemme hva koden, eller programmet, skal gjøre ved en gitt feil. For eksempel det å taste inn bokstaver i et felt som kun skal ha tall. Da kan man bestemme at for den feilen, så skal programmet gjøre dette.   
*F.eks print(“The input type was wrong”)*

Teorioppgave 2:

En klasse er et objekt som kan inneholde flere beskrivende variabler, eller objekter, som så kan brukes rett inn i funksjoner og metoder. Si hvis vi har en klasse Person. En person har gjerne flere egenskaper. For eksempel et navn, en alder, adresse, og en jobb. Her har jeg skrevet litt kjapp kode med definisjonen av en klasse, og en liten funksjon som bruker den. Merk at det strukturen ble litt rar da jeg limte den inn i word-dokumentet.

class Person:  
 def \_\_init\_\_(self, fornavn, etternavn, alder, jobb):  
 self.fornavn = fornavn  
 self.etternavn = etternavn  
 self.alder = alder  
 self.jobb = jobb  
  
 def hent\_info(self):  
 return f"Personens navn er {self.fornavn} {self.etternavn}," \  
 f" er {self.alder} år gammel, og jobber som {self.jobb}"

Som ved å skrive:   
Person1 = Person("Sindre", "Flatin", 28, "Teamleder")  
print(Person1.hent\_info())

Vil returnere:  
*Personens navn er Sindre Flatin, er 28 år gammel, og jobber som Teamleder.*

Teorioppgave 3:

Et objekt er rett og slett noe vi kan definere, eller bruke i funksjoner. Som for eksempel Personen jeg hadde med i forrige oppgave. I det tilfellet er objektet det jeg oppretter ved hjelp av klassen med dens forhåndsdefinerte egenskaper. Objektet er da meg, eller Sindre Flatin. Jeg er litt usikker på det neste her, men jeg mener også at variabler kan annsees som objekter i den eventuelle koden.

Dokumentasjon

Jeg prøvde å gjennomføre oppgaven uten å se på de mer detaljerte oppgavene og laste ned blackjack-modulen. Men etter 3-4 timer hadde jeg ikke klart å planlegge oppsettet på kortstokk-listene og programmet, så valgte etter hvert å jobbe meg gjennom de tidligere nevnte oppgavene. Jeg lastet også ned den tilhørende .py-filen.

På oppgave 1.2 C, og D endte jeg opp med å slite mer enn jeg hadde forventet. Jeg satt vel i 5 timer og nistirret på koden før jeg endelig fant ut hva feilen var etter å ha spurt en klassekamerat. Jeg hadde slurva med variablene, tegnsetting når det kommer til riktig bruk av ‘<’ og ‘>’, og lagt til et par if-setninger som var fullstendig unødvendig i funksjonen jeg hadde definert til programmet. Dette brukte jeg mye tid på å finne ut av. En av de if-setningene jeg endte opp med å flytte på var denne:

if player\_score > 21:  
 print(f"You now have {player\_hand} with a value of {player\_score}")  
 print("Busted! You got more than 21\nDealer wins!.")  
 exit()

Denne flytta jeg til slutt inn i programkoden, i stedet for å ha den inne i print\_results-funksjonene jeg hadde definert. Det å ha tunga rett i munnen når jeg brukte > og < lærte jeg fort var rimelig viktig også.

Det neste jeg slet med å implementere, var å få programmet til å velge ess som 1, eller 11. Det har jeg prøvd meg litt fram på, men det ser ikke til å funke i det hele tatt. Etter å ha satt det på is en stund, så ga jeg det et nytt forsøk, men det må være noe jeg misser, rett og slett. Jeg klarte ikke å få det til.

Så var det det å implementere chips. Det knota jeg en del med å se for meg hvordan skulle se ut. Etter å ha sittet med det i veldig mange timer, og for så vidt dager, innså jeg at jeg tror jeg har lagt opp programmet feil, i den forstand at det ikke alltid går tilbake til ønsket utgangspunkt når spilleren har valgt “stand”. Til slutt fungerer det nokså greit, men jeg det er noen få feil noen ganger. Og hvorfor det ikke er enten, eller, det vet jeg ikke.

Prøvde til slutt å implementere muligheten til å lagre antall chips til fil, men det fikk jeg rett og slett ikke til:

try:   
 with open("score\_save.txt") as chip\_stack:   
 if open("score\_save.txt".format(int)) > 0:   
 chip\_stack = int(open("score\_save.txt"))  
 else:  
 chip\_stack = 5  
except FileNotFoundError:  
 print("File was not found")

Dette var koden jeg endte opp med å fjerne fra toppen av koden. Jeg fikk det å skrive til fil til, men klarte ikke hente filen inn i spillet som en gyldig verdi for chips.

Nå var det helt klart ting jeg ikke fikk til. Men om det er noe jeg sitter igjen med etter denne oppgaven, så er det helt klart dette; Jeg har aldri vært borti noe som har vært så ekstremt frustrerende, men likevel så utrolig artig og lærerikt å jobbe med. Så fort jeg begynte, så kunne det gå åtte til 13 timer før jeg innså at jeg hadde kodet mase uten å få i meg et måltid.