

1. Åpne, lese fra, og skrive til filer:

1. Oppgave 1:

Lag et program som oppretter en tekstfil kalt student.txt og skriver navnene til fem studenter, hver på en ny linje. Åpne deretter filen og skriv ut innholdet i konsollen.

2. Oppgave 2:

Lag et program som ber brukeren om å skrive inn fem setninger. Programmet skal lagre disse i en fil kalt innspill.txt og deretter lese innholdet og skrive det ut til konsollen.

3. Oppgave 3:

Skriv et program som leser en CSV-fil med informasjon om produkter (produktnavn, pris, antall) og skriver ut produktinformasjonen formatert i en tabell i konsollen.

4. Oppgave 4:

Skriv et program som åpner en fil, leser hele innholdet, og teller antall ord i filen. Vis dette tallet til brukeren.

5. Oppgave 5:

Lag et program som ber brukeren skrive et sitat og legger det til i en fil kalt quotes.txt. Hvis filen allerede eksisterer, skal programmet ikke overskrive eksisterende data, men legge til det nye sitatet på en ny linje.

6. Oppgave 6:

Skriv et program som leser en tekstfil og teller antall ganger et spesifikt ord forekommer. Brukeren skal skrive inn ordet som skal telles.

2. Behandle unntak ved filhåndtering:

1. Oppgave 1:

Lag et program som prøver å åpne en fil kalt data.csv. Hvis filen ikke finnes, skal programmet skrive ut en feilmelding til brukeren uten å stoppe programmet.

2. Oppgave 2:

Skriv et program som ber brukeren om et filnavn og prøver å åpne denne filen. Hvis filen finnes, skal programmet lese innholdet. Hvis filen ikke finnes, skal det opprettes en tom fil med samme navn.

3. Oppgave 3:

Skriv et program som håndterer både FileNotFoundError og PermissionError. Hvis brukeren ikke har tilgang til filen, skal en egendefinert melding vises.

4. Oppgave 4:

Lag et program som åpner en fil for lesing, men som bruker try-except blokker for å håndtere tilfeller der filen er ødelagt eller ikke kan åpnes av andre grunner.

5. Oppgave 5:

Lag et program som prøver å skrive til en fil som brukeren angir. Hvis filen allerede er åpen i et annet program, skal programmet håndtere feilen ved å gi brukeren en forståelig melding.

6. **Oppgave 6:**

Skriv et program som leser fra en fil og håndterer situasjoner hvor filen er for stor til å leses inn i minnet på en gang. Del innlesningen opp i mindre deler og skriv ut innholdet gradvis.

3. Arbeid med filsystem (Sletting av filer, Opprettelse og håndtering av mapper):

1. **Oppgave 1:**

Skriv et program som oppretter en ny mappe kalt prosjektfiler. Inne i denne mappen skal det opprettes tre filer med navnene fil1.txt, fil2.txt, og fil3.txt, og hver fil skal inneholde litt testdata.

2. **Oppgave 2:**

Lag et program som ber brukeren om et filnavn og prøver å slette denne filen. Hvis filen ikke eksisterer, skal programmet skrive ut en melding som sier "Filen finnes ikke".

3. **Oppgave 3:**

Lag et program som oppretter en mappe kalt arkiv og flytter en fil fra en annen katalog inn i denne mappen. Hvis filen allerede finnes i arkiv, skal programmet gi beskjed til brukeren.

4. **Oppgave 4:**

Skriv et program som sletter alle filer med filendelsen .tmp fra en spesifisert mappe. Programmet skal først skrive ut hvilke filer som vil bli slettet, og deretter spørre brukeren om bekreftelse før slettingen utføres.

5. **Oppgave 5:**

Lag et program som sjekker om en mappe kalt backup eksisterer. Hvis den ikke eksisterer, skal programmet opprette mappen og lage en kopi av en spesifisert fil inne i denne mappen.

6. **Oppgave 6:**

Lag et program som ber brukeren om et mappenavn. Programmet skal deretter telle og vise antall filer og mapper inne i den spesifiserte mappen.

4. Feilhåndtering og unntak (try-except blokker og håndtering av forskjellige unntak):

1. **Oppgave 1:**

Lag et program som ber brukeren om å skrive inn to tall. Programmet skal forsøke å dele det første tallet med det andre og håndtere ZeroDivisionError ved å gi en melding om at man ikke kan dele med null.

2. **Oppgave 2:**

Skriv et program som leser en fil fra brukeren. Bruk try-except blokker for å håndtere feil som oppstår hvis filen ikke finnes, eller hvis det oppstår andre feil ved filtilgang.

3. **Oppgave 3:**

Lag et program som håndterer flere forskjellige typer unntak, for eksempel feil ved inndatatype (ValueError), og skriver til brukeren hvilken type feil som oppstod.

4. **Oppgave 4:**

Lag et program som leser fra en fil, men som inkluderer en finally blokk som alltid lukker filen, uansett om det oppstod en feil eller ikke.

5. **Oppgave 5:**

Skriv et program som prøver å skrive til en fil som brukeren spesifiserer, men som håndterer `PermissionError` hvis brukeren ikke har skrivetilgang til filen.

6. **Oppgave 6:**

Lag et program som bruker flere `except` blokker for å fange forskjellige typer unntak (f.eks. `FileNotFoundError`, `IsADirectoryError`, `PermissionError`) og gir brukeren en spesifikk melding basert på hvilken feil som oppstod.