## **Oppgave 1**

* KM:bruke programmering til å innhente, analysere og presentere informasjon fra reelle dataset

**Praktisk info:**

* Denne oppgave inkluderer delvis og bygger videre på oppgave 5 fra Blandede oppgaver du/dere har jobbet med i timen. Du/dere trenger ikke å bruke OOP i denne oppgaven. Du kan lage egne funksjoner for hver oppgave, og kan gjenbruke/ kode dere har skrevet i timen, men noen tilpasning/utvidelse vil være nødvendig.
* For denne oppgaven skal du bruke den vedlagt json dokument «todos\_oppg1.json»
* Besvarelse for denne oppgave skal du levere i en undermappe som inkludere: todos\_oppgave1.json, python kode, bilder for oppgavene 6 og 7 og en word dokument med en snapshoot av konsolle for deloppgavene, 3, 4, 5 og 6.
* NB! Jobber du i par, må begge forstå koden som leveres.

**Beskrivelse av oppgaven:**Du skal lese opp et datasett med med oppgaver knyttet til et fiktivt utviklingsprosjekt. Du skal skrive ut både tekstlig informasjon på konsoll og lage passende grafer til informasjon basert på innhold i filen.

1. Åpne filen og undersøke data i den.
2. Lag et program som leser inn fila
3. Skriv ut en oppsummeringslinje som sier hvor mange brukere jobber i prosjektet, hvor mange oppgaver finnes i prosjektet og hvor mange er allerede fullført.
4. Skriv ut en liste over alle oppgavene som ikke er utført (de er ikke *completed*), og en liste over alle oppgavene som er utført
5. Finn ut hvilken bruker som har utført flest oppgaver. Hvor mange oppgaver har hen utført? Skriv ut en passende beskjed på konsollen
6. **Lag et passende diagram som viser oversikt over antall utført og ikke utført oppgaver**
7. **Lag et passende diagram som viser hvilke prosent av oppgaver av de totale har hvert av personer utført.**

## **Oppgave 2**

* KM: vurdere og bruke strategier for feilsøking og testing av programkode
* Denne oppgave bygger videre på oppgave 1, men har fokus på feilsøking og testing.
* For denne oppgaven skal du fortsatt bruke den vedlagt json dokument «todos\_oppg1.json», i tillegg til at du må lage en egen json fil.
* Besvarelse for denne oppgave skal du på en egen undermappe og skal inkludere egen opprettet json fil, kopi av python kode fra oppgave videre utviklet med tester og en word dokument med et snapshot av konsollen
* NB! Jobber du i par, må begge forstå koden som leveres

Beskrivelse av oppgaven:  
I denne oppgave må du forholde deg til mulig feildata i datase og håndtere det når du leser inn innhold fra json filen.

1. Du får vite at dataset kan inkludere duplikater og at title kan inkludere ekstra blanks melom ordene som er uønsket. duplikater skal fjernes.
   * ved manglende owner skal det settes estimat settes til 3 dager
   * uønsket blanks må fjernes fra title.
2. Samtidig observerer du at noen estimat mangler og at userID i datasett er skrevet i JSON-filen som en desimal tall (1.0)i stedet for et heltall (1), og dette gjør at det ikke regnes riktig antall oppgaver per bruker. Dette må du ta hensyn til når du laster opp datasett:
   * userId som er satt til desimaltall, f.eks. 1.0 burde bli med som oppgaver tilordnet userID = 1
   * Manglende estimat settes til 1 dag. Estimatene konverteres til int.

NB! Du skal lage deg en egen json fil med noen få task som dekker disse tilfeller, og som du kan bruke for å teste funksjonalitet. Skrive passende

**Oppgave 3**

* utforske og vurdere alternative løsninger for design og implementering av et program

Praktisk info:

* Besvarelse for denne oppgave skal du på en egen undermappe og skal inkludere en klassediagram og en word dokument med deres vurderinger angående modell som presenteres. Eventuelt python kode hvis du jobber med det.

Du får vite at noen har laget en enkle ToDoListApp og ønsker å kunne forholde seg til informasjon fra json file som objekter. Velkommende har laget et enkelt grafisk grensesnitt i python for å vise lister med oppgaver og filtrere oppgaver med utgangspunkt i owner (userId) og om oppgave er utført eller ikke. Han har tatt utgangspunkt i en samtale som dere har hatt og sender noen kode:

Et bilde som inneholder tekst, skjermbilde, programvare, display

Automatisk generert beskrivelse

Konstruktor til TodoListApp: (fokus på linje11-14Et bilde som inneholder tekst, skjermbilde, programvare, Multimedieprogramvare

Automatisk generert beskrivelse

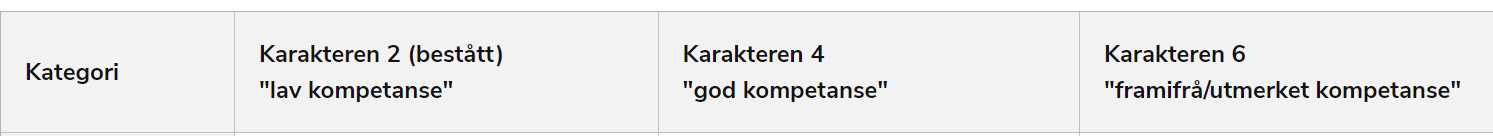
Kode fra create\_gui()- funksjon der grafisk grensesnitt bygges. Fokus på linjene 45-47

Et bilde som inneholder tekst, skjermbilde, programvare, Font

Automatisk generert beskrivelse

* 1. Lag en klassediagram som passer for data i json filen og slik at modellen som resulterer kan brukes av denne koden.
  2. Skriv hvilken vurdering dere har gjort angående klasser, medlemvariabler og metoder. Hvordan kunne modelle støtte at oppgaver blir filtrert per brukers og status completed?

Til info: legger til Vurderingskriterier fra Examen, som UDIR har utarbeidet for første gang for faget i år.

 Et bilde som inneholder tekst, skjermbilde, Font

Automatisk generert beskrivelse Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse Et bilde som inneholder tekst, skjermbilde, Font, nummer

Automatisk generert beskrivelse