

Oblig 1

Nytt forsøk

Forfall 20. jan. av 23.59 **Poeng** 1 **Må leveres** en filopplasting

Alle oppgavene skal være forsøkt løst etter beste evne (du *må* ikke løse bonusoppgavene). Hvis du har en oppgave som du har forsøkt på, men ikke fått helt til, er det OK om denne kommenteres ut slik at vi kan se hvordan du har tenkt/hva du har forsøkt på. Resterende kode bør kunne kompileres og kjøres uten at det kræsjer. Oppgaver som er gjennomført og fungerer burde *ikke* kommenteres ut.

Les: [Obligatoriske oppgaver - Informasjon \(https://hiof.instructure.com/courses/7058/pages/obligatoriske-oppgaver-informasjon\)](https://hiof.instructure.com/courses/7058/pages/obligatoriske-oppgaver-informasjon)

Obligen er strukturert opp i oppgaver, dere skal levere alle filene samlet.

Teorispørsmålene: Leveres som docx/pdf/txt navngitt: Oblig1_PLANETS_<DittNavn>.<filtype>

Eks: Oblig1_PLANETS_LarsEmilKnudsen.pdf

Programeringsoppgavene: Du skal levere **.java** filene som en .zip fil navngitt:

Oblig1_PLANETS_<DittNavn>.zip

Eks: Oblig1_PLANETS_LarsEmilKnudsen.zip

I TILLEGG: I denne obligen er det fint hvis dere laster opp **alle .java filene en og en** direkte, i tillegg til en .zip fil med alle.

Det kan bli noen mindre endringer på oppgaven, dette vil i så fall informeres om.

Teori

Du skal svare så godt du kan på disse teroioppgaven, meg *egne ord*.

Oppgave 1.1

Forklar hva JRE (Java Runtime Environment) og JDK (Java Development Kit) er.

Oppgave 1.2

Forklar flyten av hvordan java-applikasjoner blir bygget og kjørt.

Knytt også forklaringen din opp mot terminal-kommandoene, *javac* og *java*.

Oppgave 1.3

Forklar forskjellen på *compile time* og *runtime* errors. Gi noen eksempler

Oppgave 1.4

Forklar forskjellen på en klasse og et objekt. Vis gjerne et lite eksempel på hvordan man definerer en klasse, og oppretter et objekt.

Programmering

Dere skal kun benytte en teksteditor samt kommandolinja til å kompilere og kjøre koden dere lager i denne obligatoriske oppgaven. Altså ikke benytte IntelliJ eller lignende IDE'er. Dette gjør at dere får et forhold til hvordan Java-programmer kompileres før de kjører. Det er tilstrekkelig å levere .java filene dere lager, det er *ikke nødvendig* å levere de kompilerte .class filene.

Oppgave 2.1

Lag programmet Hello World! ved hjelp av din favoritt teksteditor, og kjør dette ved hjelp av Java SDK i terminalen/kommandolinja.

Oppgave 2.2

Gravitasjonen på månen er ca. 17 prosent av jordens. Skriv et program som beregner din vekt på månen.

Oppgave 2.3

I denne oppgaven skal du lage to klasser, og vise at du kan lage en klasse og opprette nye objekter av denne klassen.

a) Lag en klasse som representerer en planet. Det vi ønsker å vite om en planet er *navn*, *radius* og *masse* (den skal altså ha disse instansvariablene). En planet skal kunne opprettes ved hjelp av en konstruktør på følgende måte:

```
Planet planet = new Planet("Mars", 3889, 6.39E23);
```

Husk å gjøre instansvariablene private, og lag get- og set-metoder. Da får vi en innkapsling av disse (ett av prinsippene i OOP).

b) En klasse kalt "Oppgave2_3.java" som benytter Planet-klassen du akkurat har laget. Denne må ha en main-metode slik at du får kjørt den. Du skal her instansiere noen Planet-objekter og deretter skrive ut informasjon om disse til terminalen på følgende måte:

Planeten Mars har en radius på 3389.5 km og en masse på 6.39E23 kg

Bonusoppgave 3.1

Du skal gjenbruke Planet-klassen du laget i oppgave 2.3 og lage en ny klasse som tar input fra terminalen til å opprette et objekt av Planet-klassen.

Lag en klasse med en main-metode som tar input fra terminalen. Det enkleste er nok å benytte [Scanner](https://docs.oracle.com/en/java/javase/13/docs/api/java.base/java/util/Scanner.html) (<https://docs.oracle.com/en/java/javase/13/docs/api/java.base/java/util/Scanner.html>)-klassen som finnes i Java til dette. Benytt så inputen du får til å registrere data om en planet, opprette et planet objekt, og til slutt skrive ut data om denne.

En kjøring av programmet i terminalen vil se omtrent slik ut:

```
Skriv inn navn på planeten: Saturn
Skriv inn radius til planeten: 58232
Skriv inn massen til planeten: 5.683E26
```

Saturn har en radius på 58323.0 km og en masse på 5.863E26 kg.

Bonusoppgave 3.2

Utvid bonusoppgave 3.1 til å kunne registrere et vilkårlig antall planeter etter hverandre. Sørg for å kunne avslutte applikasjonen.

Bonusoppgave 3.3

Utvid bonusoppgave 3.2 til å holde på alle planetene du registrerer. Mellom hver registrering, legg til muligheten for å liste ut de planetene som er registrert tidligere, slette eller oppdatere en av dem, eller starte en ny registrering.