Generelle Spørsmål

* Hva er git og hvorfor er git nyttig?
  + For versjonskontroll, publisering, sammarbeid
* Kan du forklare de vanligste Git-kommandoene du har brukt?
  + add, commit, push, pull
    - add: forteller git at man vil legge til endringene i denne filen til de filene som skal bli commitet.
    - Commit: Lagrer «et bilde av» endringene, og git vil aldri endre disse endringene uten at man forteller spesifikt at git skal endre de.
    - Push: Oppdaterer et «fjernt» bibliotek, med dine lokale endringer.
    - Pull: Henter endringer fra det «fjerne» biblioteket, til det lokale biblioteket. Git fetch, git merge FETCH\_HEAD.
* Hva er formålet med en .gitignore-fil?
  + Forteller git hvilke filer git skal ignorere. Ved bruk av regexp kan man generalisere slik at git kan ignorere alle .exe filer.
* Hva er en git commit melding, og hvorfor er de viktige?
  + Meldingen der der for å gjøre endringen i filen lettere å forstå, eller at man raskt ser hva som ble endret.
* I alle tre prosjektene har vi presisert viktigheten av god kodestil. Kan du forklare oss hva kodestil er for noe? Hvordan har du jobbet for å oppnå god kodestil?
  + Pep 8
  + En kod kodestil er en konsistent måte å skrive kode på. Slike at koden blir leselig. Eksempel kan være pep8 for python
* Hvilket av de tre prosjektene syns du var vanskeligst? Hvorfor det?
  + Nytt språk nye problemer.
  + Nye bugs, problemer med compiler, tregere å søke etter bugs, mangel på index error.
* Har du opplevd noen rare bugs i kurset? Kanskje mens du jobbet med C++?
  + Hvordan gikk du frem for finne ut av løse problemet?
  + Hva var buggen?
    - Likned list: insert.
* Nevn noen teknikker man kan bruke for å gjøre koden mer effektiv?
  + Vektorisering. Trekk alle tilfeldige først.
  + I linkked list, går fra den raskeste siden. Circ: tar mod av indeksen slik at man ikke trenger å gå 20 ganger rundt.
* Hva bør man gjøre før man starter å optimalisere koden?
  + Sjekke at den fungerer, sjekke hva i koden som tar tid. Profiler.
* Hva er forksjellen på multiprocessing og multithreading?
  + Processing: bruker flere CPUer
  + Bruker: istedenfor å vente kan man gjøre noe imens.
* Når vi lager en klasse går det an å implemetere slike “magiske metoder” som starter med \_\_. Ett eksempel er \_\_add\_\_. Kan du gi noen andre eksempler? Hvorfor er slike metoder nyttige?
  + Disse er nyttige for eksempel \_\_add\_\_ som definerer hva som skjer om vis skriver instance1 + instance2. hvis classen til instance er en liste kan man enten velge å legge listene sammen eller legge de sammen i en liste.
* Gi noen gode argumenter for hvorfor vi bør skrive tester?
  + For å forsikre oss om at koden fungerer, selv om vi gjør en forskjell.
  + Man kan gått skrive testene først, slik at man vet at koden fungerer hvis testen sier så.
* Hva er forskjellen på instansmetoder, klassemetoder of statiske metoder?
  + Instansmetoder: den vanlige metoden. En metode som kan hente informasjon fra self.
  + statiske metoder: kan ikke hente verdier fra self og klassen trenger ikke være inint før den kalles.
  + Class methods: kan kalle på statiske metoder, men kan ikke hente self, men kan hente cls. Kan modifisere klasse ting.
* Nevn de fire pilarene innenfor objekt orientert programmering og fortell kort hva de innebærer
  + Abstraction: vise bare det som er nødvendig til brukeren.
  + Encapsulation: gjømme data som bruker ikke trenger
  + Inheritance: bruke allerede lagde classer for å lage nye ved å arve metoder.
  + Polymorphism: en underklasse kan ha sin unike oppførsel. Men fortsatt ha mye av de samme egenskapene.
* Hva er et design pattern?
  + En allerede testet og god måte å låse et vanlig problem på.
* Hvordan gjør man arv i python? Hvordan kaller man på metoder i foreldre klassen?
  + Super(classname, self).method()
  + Parentclas.method()
* Hva er et namespace, og hvorfor er namespace nyttige?
  + Namespace i python er for eksempel hvis vi vil bruke numpy og math i python. Så har begge funksjoner for sqrt. Så uten namespace vil python bruke den som ble importert sist. Men med namespace kan vi kalle numpy sin sqrt for np.sqrt.
* Gi eksempler på sorteringsalgoritmer? Hvordan kan vi vurdere hvilken sorteringsalgoritme vi bør velge?
  + Bouble sort, tree sort.
* Hva er et binært søketree? Hva er kjøretiden på de ulike operasjonene vi kan gjøre på et binært søketre?
  + O(logn \* n)
* Hvorfor er vi opptatt av høyden til et binært tre?
  + Det er høyden som bestemmer kjøretiden.
* Hva er rekursjon? Gi ett eksempel hvor vi bruker rekursjon?
  + Fibonacii: fibo(n) = fibo(n-1) + fibo(n-2)
* Hva er en random number generator? Hva er en pseudorandom number generator? Hvordan kan vi generere “tilfeldige” tall på en datamaskin?
  + Middle-square method.
    - Henter tallet som er midt i kvadratet av frøet.
* Hva er hensikten med å bruke et seed (frø)?
* Hva sier “pigeonholde principle”?
* Hva er en random walk?
* Hva er en markov chain?

Prosjekt 1 - Dobbelpendel

* I prosjekt 1 måtte dere løse systemer av ordinære partielle differensialligninger (ODE), hvordan gikk dere frem for å gjøre det?
  + Jeg implementerte RHS i en callen til pendulum. Og så en sovle metode som bruker solve\_ivp fra scipy.integrate
* Hvordan ble objekt-orientert programmering brukt i prosjektet?
  + Brukte arv til å implementere DampenedPendulum slik at vi bare trengte å lage ny call function.
  + Kunne nokk også arvet i alle pendulum classene. Siden vi har mange properties som er like og metoder som er lik. F.eks solve er veldig lik. Men trenger bare en forksjellig utpakking.
  + Hva var fordelene?
* I prosjektet lagde dere en DoublePendulum klasse man kunne bruke for a løse bevegelsesligningene til en dobbelpendel og plotte resultatet. Kan du forklare i grove trekk hvordan man går frem for a bruke denne klassen?
  + Lager et objekt med init verdier, ingen nødvendige. Så løser med solve. Da har man verdiene regnet ut og tilgjengelige.
  + Kan også lage animasjon etter å ha kalt solve, med create animation, dermed kan man lagre og vise animaskjonen. Med save animation og show animation.
* Flere steder ble dere bedt om a skrive unit tests/enhetstester. Kan du forklare hva en enhetstest er?
* Klarer du å skrive opp en av enhetstestene du lagde i prosjekt 1 på tavla? Det er ikke krise om den ikke blir helt riktig, målet er bare å vise ideen.
* I prosjektet ble dere bedt om a bruke @property dekoratoren. Kan du forklare hva denne er, og hvordan den brukes?
* I koden får man tilgang til løsningen som attributter på objektet, f.eks pendulum.time, pendulum.x, pendulum.y.
  + Hva skjer om man prøver a bruke disse attributtene før man har kallet på solve?
    - Attribute error.
  + Kan du forklare hvordan dette ble implementert?
* I prosjektet implementerer dere en \_\_call\_\_ metode. Hva gjør denne metoden?
* I oppgave 2g) skal dere lage en dempet pendel, altså en pendel med friksjon. Dette gjøres gjennom “arv”
  + Kan du forklare hvordan arv brukes i denne oppgaven?
  + Hva er fordelen med arv i dette tilfellet?

Prosjekt 2 - Lister

* I dette prosjektet lager dere to typer lister. Hvilke to typer er dette, og hva er hovedforskjellene mellom dem i grove trekk?
* Hva er forskjellen på private og public?
  + Kan du bruke ArrayList klassen som et eksempel? Hvilke attributter bør være private, og hvilke bør være public?
* Hva er forskjellen på klasser og structs i C++. Når bruker vi hvilken?
* Til både ArrayList og LinkedList skulle dere lage en destructor-metode. Hva gjør denne metoden, og hvorfor er den viktig i teorien? Hvorfor er den ikke alltid viktig i praksis?
* Hva er en array liste? Hva er baktanken?
* Hva er en lenket liste? Hva er baktanken?
* I ArrayList definerer dere en funksjon som heter resize. Hva gjør resize operasjonen?
* Hvordan er det å inserte i midten av en ArrayListe?
  + Forklar hva kosten blir i Big-O
  + Hva med a fjerne et element med remove?
  + Hva om du prøver å inserte i midten av en liste som allerede er full?
* Hva er forskjellen på en enkellenket og en dobbellenket liste?
* Klarer du å skrive append-metoden til ArrayList på tavla?
* Klarer du å skrive append-metoden til LenketListe på tavia?
* I Oppgave 3 sammenligner dere kostnadene til forskjellige operasjoner på de to type listene. Dette gjøres med stor O:
  + Kan du forklare i grove trekk hva vi mener med kostnad? Og hva stor-O er?
  + Velg en metode helt selv, f.eks. append i lenket liste, kan du forklare hvordan du går frem for å analysere kostnaden?
* I Oppgave 3 sammenligner dere de to type listene
  + Hva er fordelene med en ArrayListe?
  + Hva er fordelene med en LenketListe?
* Hvordan gikk du frem for å teste listene dine?
* Du blir bedt om a lage en metode med Signaturen int& operator[](int)
  + Hva er denne metoden?
  + Hva betyr & i dette tilfellet?
* Du blir bedt om a overloade konstruktøren med metoden: LinkedList(vector<int>)
  + Hva betyr det å overloade?
  + Hva gjør denne metoden? Hvorfor er den forskjellig fra den andre konstruktøren?
* Kan du gi et eksempel på et sted koden din kaster ett exception?
* Hva er en sirkulær lenket liste?
* Klarer du a forklare hva en Josephus-sekvens er?
  + Hvorfor er sirkulært lenkede lister gode for å finne slike sekvenser?
  + Kan du skrive opp sekvensen for n=6, k=3 for hånd?

Prosjekt 3 - Kaosspill

* Kan du forklare i grove trekk hvordan kaosspillet fungerer?
* Hva slags figur får vi ut fra trekanteksempelet?
* Hvordan lager vi selve plottet?
* Kan du forklare hvordan vi velger et tilfeldig startpunkt innenfor polygonet?
* I 1e) blir du bedt om å plotte trekanten i rødt, grønt og blått.
  + Hvordan går du frem for å gi en farge til hvert punkt?
  + Hvordan ser figuren ut i disse fargene?
* I 1f) blir du bedt om a lage figuren i farger på en annen måte
  + Hva er forskjellene i fremgangsmaten i 1e og 1f?
  + Hvordan blir figuren seende ut denne gangen?
* Når du generaliserer problemstillingen introduserer vi to parametere, n og r
  + Hva er disse to parametrene?
* Hvordan gikk du frem for å finne hjørnene i en regulær n-kant?
* Hvordan lagret du bildene du lagde?
* I 2g) blir du bedt om skrive minst 4 enhetstester. Kan du forklare hva en enhetstest er, og gi et eksempel på en test du skrev?
* I siste del av prosjektet ble dere bedt om a lage en AffineTransform klasse
  + Hva gjorde denne klassen?
  + Hva slags metode implementerte du?
  + Hvorfor tror du vi lager denne klassen?
* I 3c) må vi plukke en tilfeldig funksjon ut av 4, men ikke med uniform sannsynlighet. Forklar hvordan du kan plukke en av fire funksjoner med en vektet sannsynlighet - Man kan gjøre dette med np.random.choice, men vi så eksplisitt at dere ikke skulle gjøre det. Kan du huske hvorfor?
* Hvordan blir plottet i oppgave 3? Hva er det vi ser i figuren?
* I oppgave 4 blir du bedt om at implemtentere en klasse Variation. Hva er hensikten med denne oppgaven?