**When and why we will use Threads in our programs?**

* Fordi tråde kan bruges til at køre kode parallelt, i stedet for normalt kun at kunne gøre efter hindanden. Dette gør at man fx i normal sekvens programmering, hvis man har noget som tager lang tid at beregne vil det sætte resten af programmet ud af spillet indtil beregningen er færdig. Med tråde kan man køre beregningen ved siden af hoved programmet og derved vil det ikke stoppe programmet.

**Explain about the Race Condition Problem and ways to solve it in Java**

* “Race Condition Problem” er når to tråde vil arbejde på samme variable. Dette er et problem fordi vi ingen kontrol har over hvornår de to tråde arbejder. Hvis den ene tråd tager og lægger 1 til en integer og derefter gemmer det til en fil, og den anden tråd gør det samme. Hvis det så er samme variabel de arbejder på kan den første tråd godt nå at lægge 1 til variablen hvorefter den anden tråd lægger 1 til, derefter gemmer den første tråd variablen og tror det er fint, men i virkeligheden er variablen 1 mere end den skulle.
* I Java kan vi løse dette ved hjælp af låse og *synchronized* ordet. Hvis vi fx vælger at låse et objekt kan kun 1 tråd bruge det af gangen, hvilket løser problemet. Det er lidt det samme som *synchronized* gør. Den går ind og sørger for at det kun er en tråd der kan bruge enten metoden eller objektet på en gang.

**Explain how we can write reusable non-blocking Java Controls using Threads**

* Det kan vi gøre ved at lave de tunge beregninger i en anden tråd end brugeren bruger til at styre sit interface og så implementere et *Observer Pattern* så tråden kan gå tilbage til interfacet og vise resultatet af beregningerne.

**Explain about deadlocks, how to detect them and ways to solve the Deadlock Problem**

* *Deadlock* er når en tråd venter på et objekt som er låst af en anden tråd, og den anden tråd venter på et objekt der er låst af den første. I dette tilfælde vil begge tråde bare stå og vente på den anden og derved ikke komme videre.
* Man kan løse *Deadlock* problemet ved i værste tilfælde at dræbe en af trådene. Det vil dog være bedre at gå ind og se hvor *Deadlock*’en finder sted og så se om der er ikke en ved uden om det med enten manuelt brug af låse eller med *synchronized* ordet.