Tid – Obligatorisk Opgave

**Deadline d. 24/6-20**

# One-2-Watch

## Opgave Beskrivelse

Denne opgave går ud på at lave et multifunktionelt ur, som kan hjælpe med flere forskellige ting i din hverdag. Tænk opgaven som, at hvis DU skulle lave et ur til dig selv, der kan en masse forskellige ting; Hvordan skal det så se ud for at give et godt overblik.? Giver interfacet mening i forhold til hvad det skal kunne.? Er det let at interagere med dit ur/program og er det nemt at bruge.? Hvordan skal det virke, for at DU selv vil bruge det.?

### Opg. 3 – Igen og igen og..

Det skal være muligt at kunne lave f.eks. flere ”Timers” og flere nedtælling på én gang i jeres program.

Udvid jeres ur i forhold til opgaverne 1a, 1b og 2, så det f.eks. er muligt at lave flere nedtællinger på én gang eller flere alarmer. Dertil skal der laves en brugerflade, så I kan holde styr på alle de ekstra alarmer mm., som der måtte være/komme.

* 1b, tage tid med uret – Udvid så det også er muligt at tage omgangstid/Lap-time.
* 1b, nedtællings ur – Udvid, så det er muligt at have flere nedtællinger i gang, på én gang.
* 2, alarmer – Udvid, så det er muligt at sætte flere alarmer!
  + De forskellige ekstra funktioner skal kunne køre separat og kunne køre flere adgangen samtidigt.

**OBS!** – Husk at have i minde, at det skal være overskueligt og nemt at navigere i. Så der er overblik over f.eks. de forskellige funktioner jeres ur har/kan.

## Krav for hele projektet

* Skal lave projektet i C#.
* Skal arbejde i WFP.
* Fokus på OOP
* Skal bruge Github, med daglige Commits/Pushes.
* Skal lave minimum ét interface og gøre brug af det.
* Skal lave minimum én abstrakt klasse og gøre brug af det.

## Krav - OBS!

Vær opmærksom på brug af.:

* **Navngivning** (Lignende navngivning er ikke godt, f.eks.; string var4, void button2\_click() eller form2.cs. Disse er fremover et no-go, I er bedre end dette!)
  + Giv jeres variabler, funktioner og klasser mm., navne som giver mening for jeres projekt.
  + Brug camelCasing til jeres variabler – somDetteHer
  + Brug PascalCasing til Projektnavne/Namespaces, funktioner og klasser – SomDetteHer
  + Brug stort ’I’ foran jeres Interfaces – IsomDetteHer
* **OOP**
  + <https://www.freecodecamp.org/news/object-oriented-programming-concepts-21bb035f7260/>