

Øvelse 1: Beregninger med int og double

Du skal i denne øvelse arbejde primært med de to datatyper int og double, udføre forskellige typer beregninger og blive mere fortrolig med variabler samt indlæsning fra og udskrivning til konsolvinduet.

Udfør følgende:

- Opret en ny konsolapplikation til denne øvelse, hvor du skal implementere de angivne beregninger i følgende øvelser, hvor du (til at starte med) udelukkende anvender int-typen til at gemme data med. **Al kode skrives inden i Main()-funktionen, også erklæring af evt. lokale variable.** Vælg selv passende variabelnavne. Brug Console.ReadLine() og Console.WriteLine()/Write() til at kommunikere med brugeren.

Øvelse 1.1: Rektanglets areal

Beregn arealet af et rektangel ud fra formelen $A = \text{højde} * \text{bredde}$.

Implementér følgende:

- Indlæs højde på rektanglet
- Indlæs bredde på rektanglet
- Beregn og udskriv arealet

Afprøv din kode, og beregn forskellige arealer.

Øvelse 1.2: Hældning af linjestykke

Beregn hældning af linjestykke med startpunkt i (x_1, y_1) og slutpunkt i (x_2, y_2) ud fra formelen $h = (y_2 - y_1) / (x_2 - x_1)$.

Implementér følgende:

- Indlæs startpunktets x-koordinat x_1
- Indlæs startpunktets y-koordinat y_1
- Indlæs slutpunktets x-koordinat x_2
- Indlæs slutpunktets y-koordinat y_2
- Beregn og udskriv hældningen for linjestykket

Beregn hældningen for hver af linjestykkerne:

- $(3,3) \rightarrow (5,3)$
- $(3,3) \rightarrow (5,5)$
- $(3,3) \rightarrow (5,4)$

Beregn dernæst hældningen i hånden eller med regnemaskine.

- Er der noget, der undrer dig ved en af resultaterne?
- Overvej, hvorfor det ikke helt er, som du forventer

Øvelse 1.3: Ny beregning af linjestykkets hældning

I stedet for at bruge heltal (int) til repræsentation af koordinaterne og beregning af hældningen, skal du ændre din implementering, så du kun anvender double-typen til dine data.

Ret koden, og udfør samme beregninger som i forrige øvelse 1.2. Får du samme resultat?

Benyt **Ordet rundt**, og overvej forskellen. Hvad er den mest korrekte implementering af de to?

Øvelse 2: Manipulation af tekststreng

Som du har læst og set i dagens forberedelse, så har string-typen forskellige funktioner til at inspicere og manipulere tekststreng. I de følgende øvelser skal du arbejde med nogle disse funktioner.

Afgør selv, hvilke datatyper er passende til øvelserne.

Øvelse 2.1: Length()

Afgør længden af en tekststreng.

Implementér følgende:

- Indlæs en vilkårlig tekststreng
- Udskriv tekststrengens længde

Prøv forskellige tekststreng, og se, om længden passer.

Øvelse 2.2: Substring()

Udtræk en delstreng fra en givet tekststreng fra en bestemt position.

Implementér følgende:

- Indlæs den fulde streng
- Indlæs positionen
- Udtræk og udskriv delstrengen

Afprøv forskellige tekststreng og positioner.

Ud over delstrengens position kan du også angive, hvor lang delstrengen skal være.

Opdater din kode til:

- Indlæs den fulde streng
- Indlæs positionen
- Indlæs længde af delstreng
- Udtræk og udskriv delstrengen

Afprøv.

Øvelse 2.3: IndexOf()

Find positionen for en bestemt karakter i en angivet streng.

Implementér følgende:

- Indlæs den fulde streng
- Indlæs karakteren (dvs. bogstavet), der søges efter
Vigtigt: Gem karakteren i en variabel af typen char
- Find og udskriv positionen for den angivne karakter, hvis den blev fundet, ellers udskriv beskeden, at karakteren ikke blev fundet

Øvelse 3: Brug af Git

Det at kunne dele og versionere din kode er helt afgørende for et godt samarbejde i et team. I skal frem over arbejde med Git, der er et meget udbredt værktøj til netop versionsstyring. Men inden du går i gang, så gennemfør næste terminologi-øvelse.

Øvelse 3.1: Terminologi

Test din forforståelse af dagens emne(r) med udgangspunkt i dagens forberedelse.

Del teamet op i to mindre grupper, og brug **Møde på midten** (beskrivelse følger i punktform) i hver gruppe til at diskutere begreberne "git init", "git pull", "git add", "git commit", "git push", "git status" og "git clone". Husk, at alle i gruppen skal have taletid:

1. Hvert enkelt medlem i gruppen skriver sine overvejelser om begreberne ned (2 minutter).
2. Hver gruppe starter **Ordet rundt**, idet det første gruppemedlem forklarer sine overvejelser (2 minutter). Hele gruppen diskuterer bidraget (2 minutter). Hvis gruppen enes om, at det er en god forklaring, skriver det første gruppemedlem sit bidrag ind i gruppens dokument.
3. Det næste gruppemedlem forklarer sine overvejelser osv. Fortsæt med **Ordet rundt** indtil alle fire gruppemedlemmer har bidraget, eller tiden er gået.

Tidsramme: 20 minutter

Øvelse 3.2: Brugen af Git

Du forventes på nuværende tidspunkt at have installeret Git og have oprettet en bruger inde på GitHub (med en anden e-mail end jeres studie e-mail, så det kan bruges til fremtidig brug).

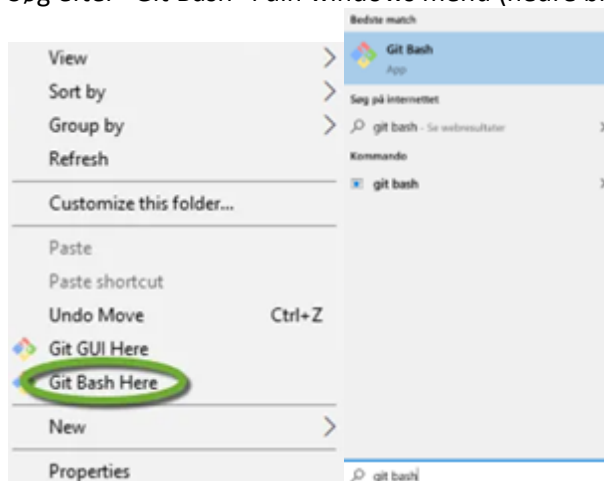
Note: Det er tilladt at benytte en anden Git repository manager end GitLab, for eksempel GitHub eller BitBucket. Det vigtigste er, at I benytter jer af Git. Men vi beskriver ikke de andre Git repository managers, kun GitLab.

På et senere tidspunkt skal du benytte Git inde fra Visual Studio IDE, men for nu, skal du afprøve Git Bash og få styr på, hvordan man gør alt dette uden brug af et grafisk brugerinterface (en: Graphical user interface).

Øvelser 3.2.1: Konfigurering af Git

Dette er en individuel øvelse, hvor du skal konfigurere Git på din maskine.

- Benyt **en** af de følgende metoder til at åbne Git Bash:
 - Søg efter "Git Bash" i din windows menu (nedre billede til venstre)



- Højreklik i en mappe eller på dit skrivebord og vælg "Git Bash Here" (nedre billede til højre)

- Tilføj dit brugernavn og din e-mail til den globale git config, benyt følgende kommandoer i Git Bash (husk at erstatte "Fornavn Efternavn" med dit navn, samt "DinE-mail@her.dk" med din studie e-mail):
 - `git config --global user.name "Fornavn Efternavn"`
 - `git config --global user.email DinE-mail@her.dk`
- Benyt følgende kommando i Git Bash: `git config --list`
 - Burde du kunne se følgende i nærheden af bunden af konsollen:

`user.name=Fornavn Efternavn`

`user.email=DinE-mail@her.dk`

Nu er du klar til næste del af øvelsen.

Øvelse 3.2.2: Oprettelse af Git repository til hjemmeprojekter

Dette er en individuel øvelse, hvor du skal oprette dit helt eget repository inde på Gitlab til dine hjemmeprojekter.

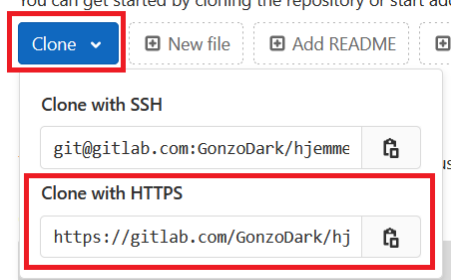
- Benyt følgende link og opret et nyt projekt inde på GitLab: <https://gitlab.com/projects/new>
 - Vælg "Create blank project"
 - Giv dit projekt et godt navn for eksempel "Hjemmeprojekter"
 - Lad dit projekt være "Private"
 - Undgå at initialisere projektet med en "README"-fil

Hvis alt er gået som det skulle, bør du nu have en nyt tomt repository, med en beskrivelse af "Command line instructions", vi vender tilbage til disse instruktioner længere nede i opgaven.

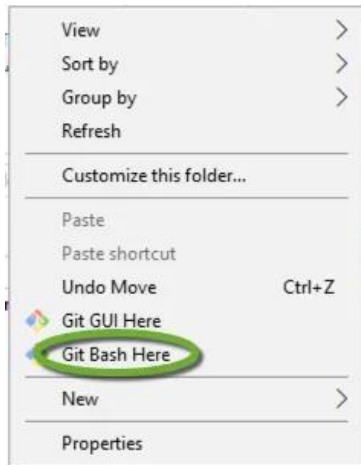
Øvelse 3.2.3: Klon dit repository via Git Bash

Du skal nu klonе dit netop oprettede Git repository.

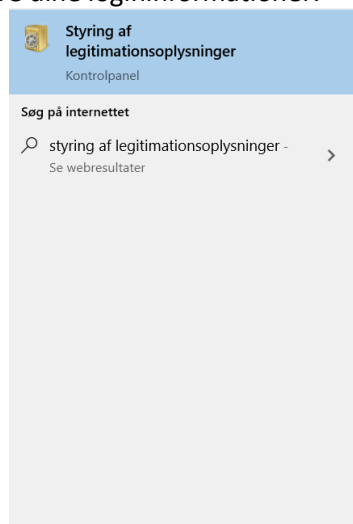
- Hvis du ikke allerede er inde og se på dit nyoprettede projekt inde på GitLab, så gå ind på <https://gitlab.com/dashboard/projects> og find dit nye projekt på listen og vælg det




- Tryk på den blå "Clone"-drop down knap og kopier adressen der står i boksen med "Clone with HTTPS", som på billedet nedenunder:
- Find et sted på din computer, hvor du har lyst til at placere dine hjemmeprojekter



- Højreklik i roden af mappen, vælg "Git Bash Here", som på billedet nedenunder:
- Benyt følgende kommando: git clone (og den adresse du kopierede tidligere fra GitLab (Clone with HTTPS))
 - For eksempel: git clone <https://gitlab.com/DitBrugerNavn/hjemmeprojekter.git>
- Nu vil Git kræve at du logger ind, og Windows beder dig om at gemme dine informationer – benyt dit GitLab brugernavn (**ikke** din e-mail) og din GitLab kode.
 - **Hvis** du ved en fejltagelse har indtastet et forkert brugernavn/adgangskode, skal du opdatere dine logininformationer:



-  styring af legitimationsoplysninger Åbn Windows-menuen og søg efter "styring af legitimationsoplysninger" (eller Credential Manager, hvis din udgave af Windows er på engelsk), som på billedet nedenunder:

[Administrer dine legitimationsoplysninger](#)

Se og slet dine gemte logonoplysninger for websteder, forbundne programmer og netværk.



Legitimationsoplysninger til websted



Windows-legitimationsoplysninger

- Vælg herefter "Windows-legitimationsoplysninger", som på billedet nedenunder:

git:https://gitlab.com

Ændret: 25-06-2020



Internet- eller netværksadresse: git:https://gitlab.com

Brugernavn: DitBrugernavn

Adgangskode:

Fastholdelse: Lokal computer

Rediger Fjern

- Se under "Generiske legitimationsbeskrivelser", lokalisér "git:https://gitlab.com" og fold punktet ud og, tryk på "Fjern", som på billedet nedenunder:
- Nu er du klar til at indtaste dine legitimationsoplysninger i Git igen

Sådan, nu har du en lokal kopi af dit repository. Denne lokale kopi arbejder vi videre med i næste øvelse.

Øvelse 3.2.4: Lav en ændring og få den op på GitLab

- Åbn Git Bash inden i din projektmappe
 - Du kan tjekke, om du er inden i mappen ved at benytte følgende kommando: git status
- ```
$ git status
fatal: not a git repository (or any of the parent directories): .git
```

Hvis du ikke er inde i en mappe med Git, vil projektet brokke sig, som således:
- Lav en ændring i mappen, benyt følgende kommando: touch README.md
- Nu skal du tjekke, hvordan status er igen, benyt kommandoen: git status
  - Git burde gerne fortælle dig, at du har nogle filer, der ikke bliver fulgt af Git (Untracked files)
- Du skal bede Git om at følge din netop oprettede fil med følgende kommando: git add README.md
- Nu skal du tjekke, hvordan status er igen, benyt kommandoen: git status
  - Nu burde Git sige, at du har ændringer der er klar til at blive committed (Changes to be committed)
    - Dette område er også kendt som "staging area" og viser dig, hvilke filer der tilføjes, når du opretter dit commit
- Commit filen med: git commit -m "message"
- Nu skal du tjekke, hvordan status er igen, benyt kommandoen: git status
- Nu skal du skubbe filen afsted op til dit remote repository (GitLab), benyt følgende kommando: git push
- Benyt git status igen, samt følgende kommando: git log
  - Bemærk, at du har en enkelt commit med et commit-id, samt en forfatter (author), der indeholder de informationer, som du indtastede i øvelse 5.2.1
- Gå ind på GitLab og bemærk, at du også har dine ændringer på dit remote repository (GitLab)
  - (genopfrisk vinduet)

Tidsramme: 30-45 minutter