HO GENT	**Hogeschool Gent – Faculteit Bedrijf en organisatie fbo** **Toegepaste Informatica 1e jaar – Academiejaar 2019 – 2020** **OEFENINGEN OO-Programmeren I – Hoofdstuk 2**

Elke opgave hieronder ziet er als volgt uit in UML:

Je hebt 1 klasse met daarin 1 methode main.

1. Schrijf een programma dat de volgende zinnen op het scherm zet aan de hand van 3 statements:
   Open de mappen
   C:\Mijn Documenten
   D:\temp\OOP

   Mijn naam is “Jan Janssens”,
   Ik woon te Gent!

   85 + 95 = 180 // 180 wordt bepaald door het programma

```
run:
Open de mappen
C:\Mijn Documenten
D:\temp\OOP

Mijn naam is "Jan Janssens",
Ik woon te Gent!

85 + 95 = 180
BUILD SUCCESSFUL (total time: 0 seconds)
```

2. Schrijf een applicatie die aan de gebruiker de zijde van een vierkant (strikt positief geheel getal) vraagt.
   Geef van het vierkant onder elkaar
   - De zijde
   - De omtrek
   - De oppervlakte

   Gebruik printf!!

```
run:
Geef een strikt positieve zijde van een vierkant in:   8
De zijde = 8
De omtrek = 32
De oppervlakte = 64
BUILD SUCCESSFUL (total time: 3 seconds)
```

3. Schrijf een applicatie die aan de gebruiker één strikt positief geheel getal vraagt.
   Geef de octale en hexadecimale notatie (met kleine- én hoofdletters) weer van het ingevoerde getal.
   Gebruik printf!!

```
run:
Geef een strikt positief geheel getal in:  26
octale notatie = 32
hexidecimale notatie (klein) = 1a
hexidecimale notatie (groot) notatie = 1A
BUILD SUCCESSFUL (total time: 12 seconds)
```

4. Schrijf een applicatie die aan de gebruiker één strikt positief geheel getal bestaande uit 5 cijfers vraagt, en dit getal terug op het scherm afdrukt waarbij alle cijfers met **drie** spaties van elkaar worden gescheiden.

```
run:
Geef een strikt positief geheel getal bestaande uit 5 cijfers, in:   85412
   8   5   4   1   2
BUILD SUCCESSFUL (total time: 4 seconds)
```

5. Voer een geheel getal in. Vermenigvuldig dit getal met 1, 10, 100, 1000 en 10000.
   Toon het resultaat van deze berekeningen in tabelvorm met bijhorende titel en aligneer telkens rechts in een veldbreedte van 15 posities:

```
run:
Geef een geheel getal in:  -9
          GETAL       GETAL*10      GETAL*100     GETAL*1000
             -9            -90           -900          -9000
BUILD SUCCESSFUL (total time: 3 seconds)
```

6. Schrijf een applicatie die aan de gebruiker twee gehele getallen (teller en noemer) vraagt en daarvan het resultaat van de gehele deling, de rest en als de rest nul is de vereenvoudigde breuk weergeeft.

```
run:
Geef de teller in van de breuk:  -12
Geef de noemer in van de breuk:  3
-12/3 = -4
rest =  0

vereenvoudigde breuk = -4 / 1
BUILD SUCCESSFUL (total time: 5 seconds)
```

```
run:
Geef de teller in van de breuk:  12
Geef de noemer in van de breuk:  5
12/5 = 2
rest =  2

BUILD SUCCESSFUL (total time: 3 seconds)
```

7. Schrijf een applicatie die aan de gebruiker een geheel getal vraagt en controleert of het getal even of oneven is. Geef een duidelijke boodschap weer.

```
run:
Geef een geheel getal in:  -22
Het ingevoerde getal -22 is een even getal
BUILD SUCCESSFUL (total time: 2 seconds)
```

```
run:
Geef een geheel getal in:  13
Het ingevoerde getal 13 is een oneven getal
BUILD SUCCESSFUL (total time: 2 seconds)
```

8. Schrijf een applicatie die aan de gebruiker drie gehele getallen vraagt en daarvan de som, het gemiddelde (als geheel getal), het product en het kleinste getal weergeeft. Bepaal het kleinste zo performant mogelijk!

   De in- en uitvoer ziet er als volgt uit:

```
run:
Geef eerste getal:  12
Geef tweede getal:  5
Geef derde getal:  42
Van de ingevoerde getallen 12, 5 en 42
is de som 59
het gemiddelde 19
het product 2520
en het kleinste getal 5
BUILD SUCCESSFUL (total time: 4 seconds)
```