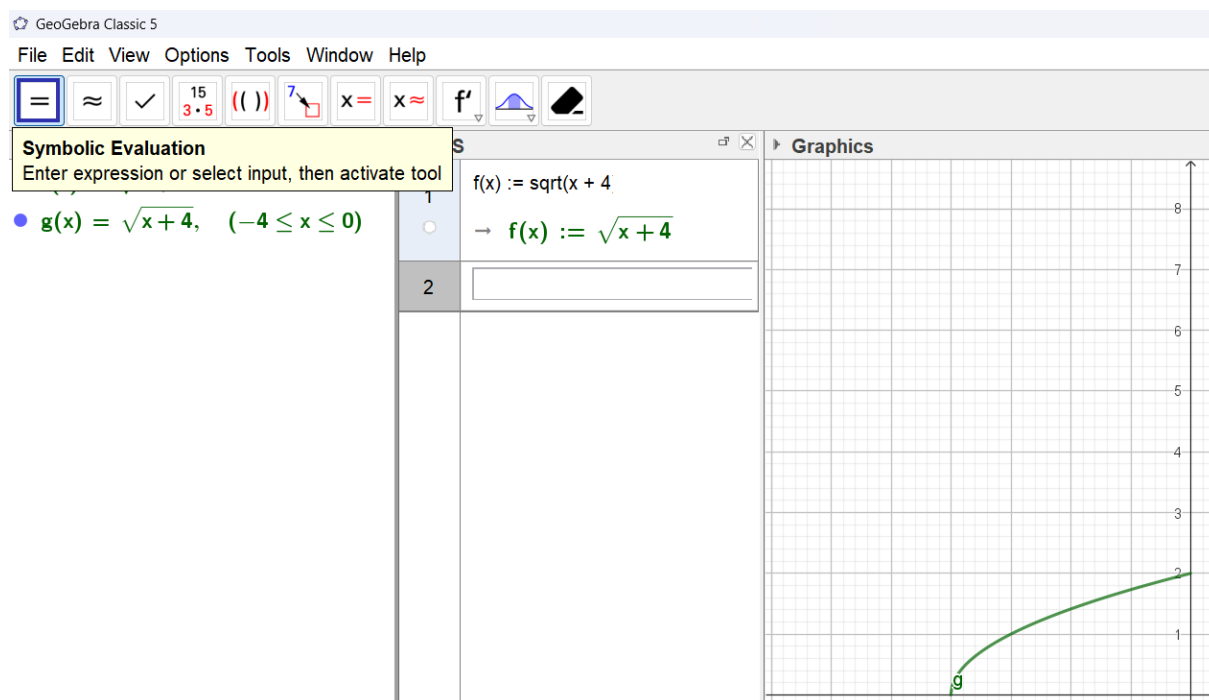


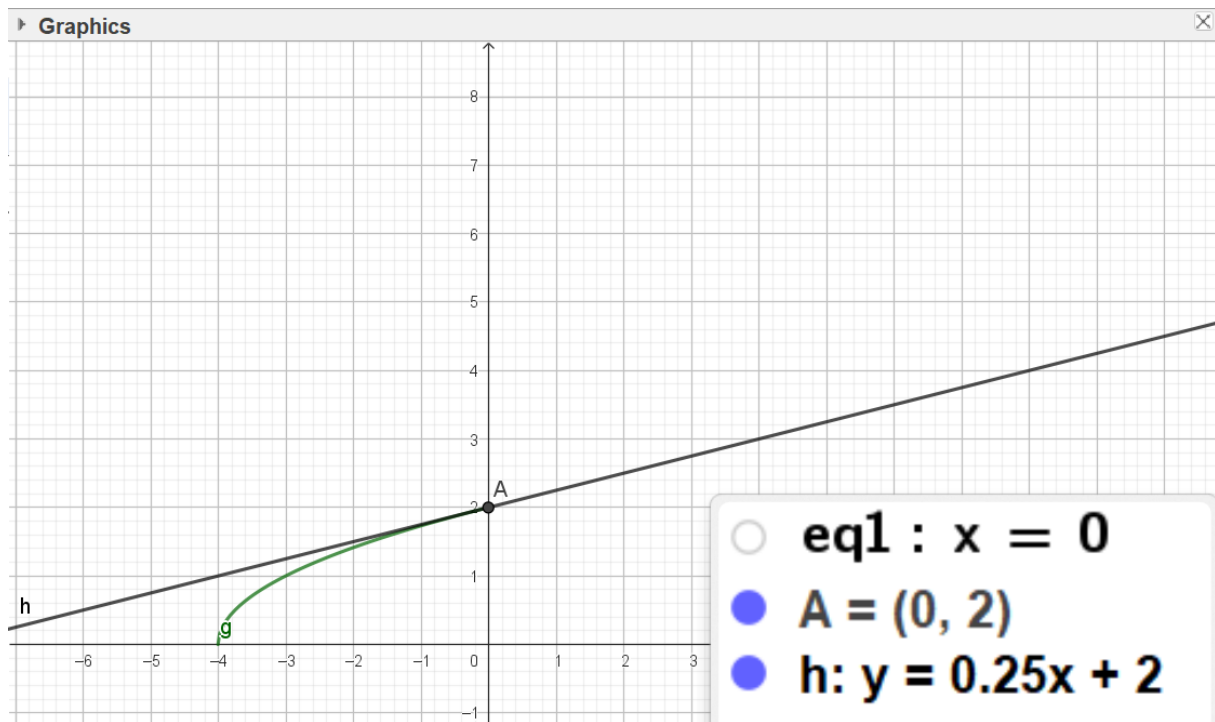
# MAT104 Obligatorisk innlevering 3

## Oppg 1

a)



b)



c)

- $h: y = 0.25x + 2$
- $w(x) = 0.25x + 2$

3	$w(0)$
<input type="radio"/>	$\approx 2$
4	$\sqrt{3.96}$
<input type="radio"/>	$\approx 1.99$

Svar = 2

d)

5	$(x, 0.25x + 2) = (x, f(x))$
<input type="radio"/>	Solve: $\{x = 0\}$

$x=0$ ?

Usikker – FEIL

Sett  $f'(x)$  = «stiningstall»

0.25 er ikke rett så er tangenten feil?

## Oppg 2

2  
b)

$$\lim_{x \rightarrow 0} \frac{3^x - e^x}{\pi^x - \cos(2x)}$$

$$\frac{3^0 - e^0}{\pi^0 - \cos(2 \cdot 0)} = \frac{1 - 1}{1 - 1} = \frac{0}{0}$$

$$\left( \frac{3^x - e^x}{\pi^x - \cos(2x)} \right)' \stackrel{[8]}{=} \frac{\ln(3) \cdot 3^x - e^x}{\ln(\pi) \cdot \pi^x + 2 \cdot \sin(2x)}$$

$$\frac{\ln(3) \cdot 3^0 - e^0}{\ln(\pi) \cdot \pi^0 + 2 \cdot \sin(2 \cdot 0)} = \frac{(\ln(3) \cdot 1) - 1}{(\ln(\pi) \cdot 1) + 2 \cdot 0} =$$

$$\frac{\ln(3) - 1}{\ln(\pi)} = \frac{1.1 - 1}{1.145} = \frac{0.1}{1.145} = 0.087$$

Løsning= 0.087?

5)

$$\lim_{x \rightarrow 1} \frac{\ln x}{x^2 - 1} = \frac{\ln(1)}{1^2 - 1} = \frac{0}{0}$$

$$\left( \frac{\ln x}{x^2 - 1} \right)' = \frac{\frac{1}{x}}{2x - 0}$$

$$\lim_{x \rightarrow 1} \frac{1}{2x} = \frac{1}{2 \cdot 1} = \frac{1}{2}$$

$$6) \quad \lim_{x \rightarrow \infty} \frac{x^2}{e^x + x^2} = \frac{2x}{e^x + 2x} = \frac{2}{e^x + 2}$$

$$\frac{2}{\infty + 2} = \frac{2}{\infty} = \underline{\underline{0}}$$



### Oppg 3

3. (b)  $\int_0^4 (3x^2 - 1) dx = \int_0^4 3x^2 dx - \int_0^4 1 dx =$   
 $\left[ x^3 - x \right]_0^4 = 4^3 - 4 - (0^3 - 0) = 64 - 4 = \underline{\underline{60}}$

c)  $\int \sin(2x) dx = \frac{1}{2} \cdot -\cos(2x) = -\frac{\cos(2x)}{2} + C$

### Oppg 4

a)

$$\Delta x = (b-a)/n$$

$$((\pi/4) - 0)/4 = \pi/16$$

$$T_n = \Delta x / (\pi/16) [f(x_0), (\pi/16) * f(x_1), (\pi/16) * f(x_2), (\pi/16) * f(x_3), \dots, f(x_n)]$$

b)

Sjekk boken.

$$\Delta x = (b-a)/n$$

$$\Delta x = \frac{(\frac{\pi}{4}) - 0}{n}$$

$$\int_0^{\frac{\pi}{4}} f(x) dx = \left( \frac{2}{\Delta x} \right) \left[ f(x_0) + f(x_n) + 2 \sum_{i=0}^{n-1} f(x_i) \right]$$

$$= \left( \frac{2}{\Delta x} \right) \left[ f(a) + f(b) + 2 * \sum_{i=0}^{n-1} f(x + i * \Delta x) \right]$$

c)

```
1
2 public class oppgave4 {
3     static double trapesHoyder;
4     public static void main(String[] args) {
5         double startVerdi = 0;
6         double sluttVerdi = (Math.PI/4);
7
8         double antallTrapeser = 25;
9         double deltax = finnDelta_X(antallTrapeser);
10        trapesHoyder = f(startVerdi)+f(sluttVerdi);
11        for (int i = 0; i < antallTrapeser; i++) {
12            trapesHoyder = trapesHoyder + 2*f(startVerdi+i*deltax);
13        }
14        double trapessum = (deltax/2)*trapesHoyder;
15        System.out.println(trapessum);
16    }
17
18    public static double f(double x) {
19        double f_av_x = (2*Math.pow(x,2))+x+5; //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
20        return f_av_x;
21    }
22    public static double finnDelta_X(double n) {
23        double delta_x = (((Math.PI/4)-0)/n); //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
24        return delta_x;
25    }
26
27 }
```

```
1
2 public class oppgave4 {
3     static double trapesHoyder;
4     public static void main(String[] args) {
5         double startVerdi = 0;
6         double sluttVerdi = (Math.PI/4);
7
8         double antallTrapeser = 25;
9         double deltax = finnDelta_X(antallTrapeser);
10        trapesHoyder = f(startVerdi)+f(sluttVerdi);
11        for (int i = 0; i < antallTrapeser; i++) {
12            trapesHoyder = trapesHoyder + 2*f(startVerdi+i*deltax);
13        }
14        double trapessum = (deltax/2)*trapesHoyder;
15        System.out.println(trapessum);
16    }
17
18    public static double f(double x) {
19        double f_av_x = (1/x); //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
20        return f_av_x;
21    }
22    public static double finnDelta_X(double n) {
23        double delta_x = (((Math.PI/4)-0)/n); //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
24        return delta_x;
25    }
26
27 }
```

```
1
2 public class oppgave4 {
3     static double trapesHoyder;
4     public static void main(String[] args) {
5         double startVerdi = 0;
6         double sluttVerdi = (Math.PI/4);
7
8         double antallTrapeser = 25;
9         double deltax = finnDelta_X(antallTrapeser);
10        trapesHoyder = f(startVerdi)+f(sluttVerdi);
11        for (int i = 0; i < antallTrapeser; i++) {
12            trapesHoyder = trapesHoyder + 2*f(startVerdi+i*deltax);
13        }
14        double trapessum = (deltax/2)*trapesHoyder;
15        System.out.println(trapessum);
16    }
17
18    public static double f(double x) {
19        double f_av_x = Math.pow(x, 2)+x+7; //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
20        return f_av_x;
21    }
22    public static double finnDelta_X(double n) {
23        double delta_x = (((Math.PI/4)-0)/n); //Må kunne endres til hva som helst annen funksjon slik koden fungerer enda.
24        return delta_x;
25    }
26
27 }
```

## Oppg 5

5

a)

$$F(x) = 6x^2 + 18x - 24$$

b)

$$6x^2 + 18x - 24 = 0$$

$$a = 6$$

$$b = 18$$

$$c = -24$$

$$\frac{-18 \pm \sqrt{18^2 - 4 \cdot 6 \cdot (-24)}}{2 \cdot 6}$$

$$\frac{-18 \pm \sqrt{324 + 576}}{12}$$

$$\frac{-18 \pm \sqrt{900}}{12}$$

$$x = \frac{-18 \pm 30}{12}$$

$$x_1 = 1 \quad \vee \quad x_2 = -4$$

$$2 \cdot 1^3 + 9 \cdot 1^2 - 24 \cdot 1 - 1 = -12$$

$$2 \cdot 4^3 + 9 \cdot 4^2 - 24 \cdot 4 - 1 = 113$$

$$\underline{\text{topp} = (1, -12) \quad \vee \quad \text{Bunn} = (-4, 113)}$$



$$c) f''(x) = 12x + 18$$

$$f''(x) < 0 \text{ for alle } x < -1.5$$

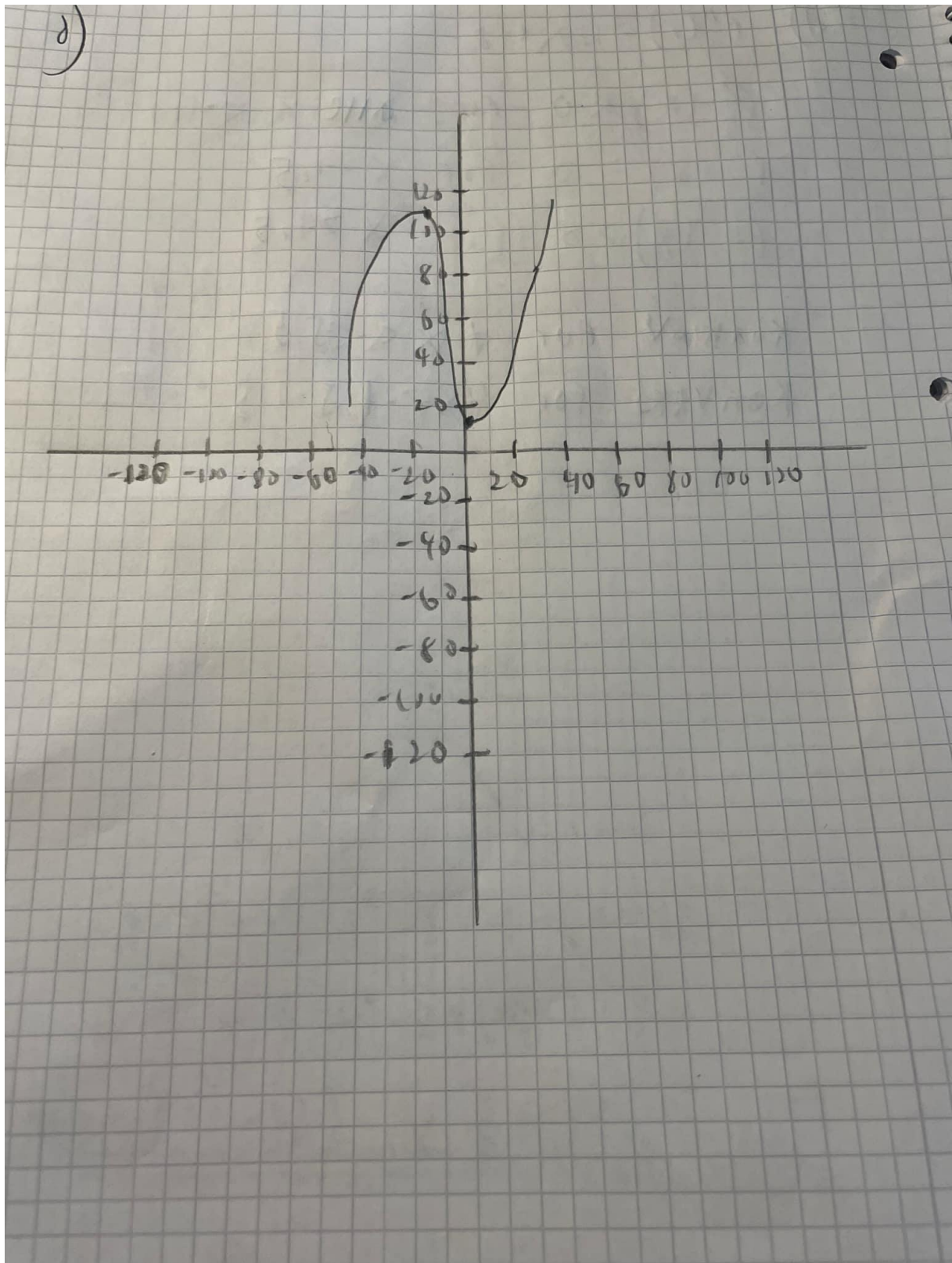
$$f''(x) = 0 \text{ for } x = -1.5$$

$$f''(x) > 0 \text{ for } x > -1.5$$

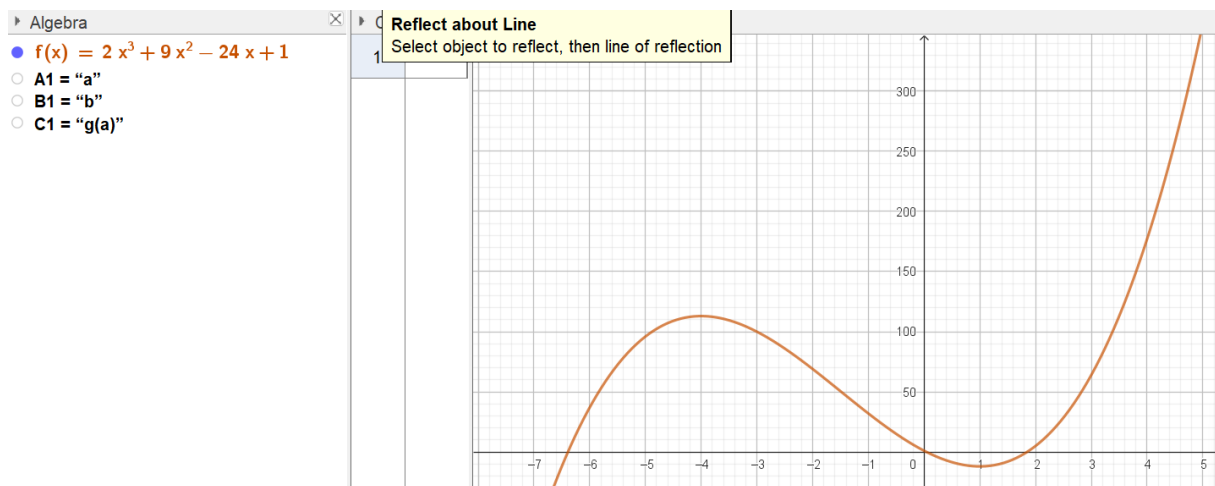
Konkav for  $\forall x < -1.5$

Konveks for  $\forall x > -1.5$

$f''(x)$  er en linærfunksjon som vil alltid øke pga positiv stigningstall. Er verken konveks eller konkav. Stigningspunkt blir 0?



e)



	A	B	C	D
1	a	b	g(a)	g(b)
2	-7	3	-76	64
3	-6.5	3	-12	64
4	-6	3	37	64
5	-5.5	3	72.5	64
6	-5	3	96	64
7	-4.5	3	109	64
8	-4	3	113	64
9	-3.5	3	109.5	64
10	-3	3	100	64
11	-2.5	3	86	64
12	-2	3	69	64
13	-1.5	3	50.5	64
14	-1	3	32	64
15	-0.5	3	15	64
16	0	3	1	64
17	0.5	3	-8.5	64
18	1	3	-12	64
19	1.5	3	-8	64
20	2	3	5	64
21	2.5	3	28.5	64

Summen for y for verdier av  $f(a)$  « $g(a)$  i tabell» gjør at når a øker vil y verdien bytter fortegn 3 ganger. Først negativt, så positivt, så negativt så positivt igjen. Det må også være 3

punkt på y mellom:  $g(a) = -12$  v  $g(a) = 37$ ,  $g(a) = 1$  v  $g(a) = -8.5$  og  $g(a) = -8$  v  $g(a) = 5$  som blir 0 for y.

Så den må gå igjennom 0 på y akse 3 ganger og dermed har 3 nullpunkt på y akse.

f)

```
1 package testproject;
2
3 import java.lang.Math;
4
5 public class oppg5f {
6     public static void main(String[] args) {
7         newtonMethod(-2, 10);
8     }
9     public static double f(double x) {
10         double y = 2*Math.pow(x, 3)+9*Math.pow(x, 2)-(24*x)+1;
11         return y;
12     }
13
14     public static double fder(double x) {
15         double y = 6*Math.pow(x, 2)+(18*x)-24;
16         return y;
17     }
18
19     public static double newtonMethod(double start, int repeats) {
20         int n = 0;
21         while (n < repeats) {
22             double x_nplus1 = start-(f(start)/fder(start));
23             System.out.println((n+1) + ": " + x_nplus1);
24             start = x_nplus1;
25             n++;
26         }
27         System.out.println(start);
28         return start;
29     }
30 }
31 }
```

```
<terminated> oppg5f [Java Applicatio
1: -0.083333333333333326
2: 0.03691580287324969
3: 0.042333752007324595
4: 0.04234541966289611
5: 0.042345419717134195
6: 0.04234541971713419
7: 0.042345419717134195
8: 0.04234541971713419
9: 0.042345419717134195
10: 0.04234541971713419
0.04234541971713419
```

g)

```

1 package testproject;
2
3 import java.lang.Math;
4
5 public class oppg5f {
6     public static void main(String[] args) {
7         newtonMethod(0, 10);
8     }
9     public static double f(double x) {
10        double y = 2*Math.pow(x, 3)+9*Math.pow(x, 2)-(24*x)+1;
11        return y;
12    }
13
14    public static double fder(double x) {
15        double y = 6*Math.pow(x, 2)+(18*x)-24;
16        return y;
17    }
18
19    public static double fdoubleder(double x) {
20        double y = 12*x+18;
21        return y;
22    }
23
24    public static double newtonMethod(double start, int repeats) {
25        int n = 0;
26        while (n < repeats) {
27            double x_nplust1 = start-(f(start)/fder(start));
28            System.out.println((n+1) + ": " + x_nplust1);
29            start = x_nplust1;
30            n++;
31        }
32        System.out.println(start);
33        return start;
34    }
35 }
36

```

(0,04, 0)



```

1 package testproject;
2
3 import java.lang.Math;
4
5 public class oppg5f {
6     public static void main(String[] args) {
7         newtonMethod(2,10);
8     }
9     public static double f(double x) {
10         double y = 2*Math.pow(x, 3)+9*Math.pow(x, 2)-(24*x)+1;
11         return y;
12     }
13
14     public static double fder(double x) {
15         double y = 6*Math.pow(x, 2)+(18*x)-24;
16         return y;
17     }
18
19     public static double fdoubleder(double x) {
20         double y = 12*x+18;
21         return y;
22     }
23
24     public static double newtonMethod(double start, int repeats) {
25         int n = 0;
26         while (n < repeats) {
27             double x_nplus1 = start-(f(start)/fder(start));
28             System.out.println((n+1) + ": " + x_nplus1);
29             start = x_nplus1;
30             n++;
31         }
32         System.out.println(start);
33         return start;
34     }
35 }
36

```

Find

> All

<terminated> oppg5f [Java Applicatio

1: 1.8611111111111112  
2: 1.847910898209018  
3: 1.8477929408977811  
4: 1.8477929315019168  
5: 1.8477929315019168  
6: 1.8477929315019168  
7: 1.8477929315019168  
8: 1.8477929315019168  
9: 1.8477929315019168  
10: 1.8477929315019168  
1.8477929315019168

(1.8478,0)

eclipse-workspace-dat100 - testproject\src\testproject\oppg5f.java - Eclipse IDE

```

1 package testproject;
2
3 import java.lang.Math;
4
5 public class oppg5f {
6     public static void main(String[] args) {
7         newtonMethod(-5,10);
8     }
9     public static double f(double x) {
10         double y = 2*Math.pow(x, 3)+9*Math.pow(x, 2)-(24*x)+1;
11         return y;
12     }
13
14     public static double fder(double x) {
15         double y = 6*Math.pow(x, 2)+(18*x)-24;
16         return y;
17     }
18
19     public static double fdoubleder(double x) {
20         double y = 12*x+18;
21         return y;
22     }
23
24     public static double newtonMethod(double start, int repeats) {
25         int n = 0;
26         while (n < repeats) {
27             double x_nplus1 = start-(f(start)/fder(start));
28             System.out.println((n+1) + ": " + x_nplus1);
29             start = x_nplus1;
30             n++;
31         }
32         System.out.println(start);
33         return start;
34     }
35 }
36

```

Find

> All

<terminated> oppg5f [Java Applicatio

1: -7.6666666666666666  
2: -6.684537684537685  
3: -6.41150799546632  
4: -6.390263660673798  
5: -6.39013835556606  
6: -6.390138351219051  
7: -6.390138351219051  
8: -6.390138351219051  
9: -6.390138351219051  
10: -6.390138351219051  
-6.390138351219051

testproject

- main(String[]): void
- f(double): double
- fder(double): double
- fdoubleder(double): double
- newtonMethod(double, int)

(-6.39, 0)

$L = \{(0,04, 0), (1.8478,0), (-6.39, 0)\}$

**TODO: (Fjern før dokument innleveres)**

- 1d – Usikker/Uferdig
- Oppg 4
  - 4a – må sjekkes
  - 4b – Må sjekkes
  - 4c – må sjekkes
- 5e - usikker