Norges teknisk–naturvitenskapelige universitet
Institutt for datateknikk og informasjonsvitenskap

TDT4110 IT Grunnkurs
Høst 2012

Løsningsforslag — Øving 7

1 Newtons metode - TMA4100

Kodesnutt 1

```
# encoding: utf-8
def polynom(x):
  return x ** 5 - 4 * x ** 3 + 10 * x ** 2 - 10

def polynom_derivative(x):
  return 5 * x ** 4 - 12 * x ** 2 + 20 * x

def newton(func, deriv, start, threshold, max_iterations):
  x = start
  xold = float("inf")
  i = 0

  while abs(xold - x) > threshold:

    if i > max_iterations:
      return False

    xold = x
    x = x - func(x) / deriv(x)
    i += 1

  return x
```

2 Simpsons metode - TMA4100

Kodesnutt 2

```python
# encoding: utf-8
import math

def evilfunction(x):
  return math.exp(-(x ** 2))

def simpson(func, a, b, n):
  h = float(b - a) / n # steglengde
  result = func(a) + func(b)

  for i in range(1, n, 2):
    result += 4 * func(a + (i * h))

  for i in range(2, n - 1, 2):
    result += 2 * func(a + (i * h))

  result *= (h / 3)
  return result

print(simpson(evilfunction, 0, 1, 1000))

def simpson_error(func, a, b, error):
  i = 1
  while abs(simpson(func, a, b, 2 ** i) - simpson(func, a, b, 2 ** (i + 1))) >
                                          error:
    i += 1

  return simpson(func, a, b, 2 ** i)

print(simpson_error(evilfunction, 0, 1, 10 ** (-8)))
```

3 Omkrets - Øvrige

Kodesnutt 3

```python
import math

def pytagoras(x1, x2, y1, y2):
  return math.sqrt((x1 - x2)**2 + (y1 - y2)**2)

def perimeter(x,y):
  p = 0
  for i in range(len(x) - 1):
    p += pytagoras(x[i], x[i+1], y[i], y[i+1])
  p += pytagoras(x[i+1], x[0], y[i+1], y[0])
  return p
```

4 Løkker - Alle

a)

Kodesnutt 4

```
import math

def is_prime(number):
  for divider in range(2, number - 1):
    if number % divider == 0:
      return False
  return True
```

b)

Kodesnutt 5

```
def separate(numbers, threshold):
  list1 = []
  list2 = []
  for i in numbers:
    if i < threshold:
      list1.append(i)
    else:
      list2.append(i)
  return list1, list2
```

c)

Kodesnutt 6

```
def multiplication_table(n):
  multiplication_table = []
  for y in range(1, n + 1):
    table = []
    for x in range(1, n + 1):
      table.append(x * y)
    multiplication_table.append(table)
  return multiplication_table
```

5 Strenghåndtering - Alle

Kodesnutt 7

```
def compare(string1, string2):
  length = len(string1)

  if length == len(string2):
    for i in range(length):
      if string1[i] != string2[i]:
        return False
    return True
  return False

def reverse(string):
  reversed_string = ''

  for i in range(len(string)-1, -1, -1):
    reversed_string += string[i]

  return reversed_string

def palindrome(string):
  return string == reverse(string)

def contains(string1,string2):
  for x in range(len(string1)):
    if string1[x:(len(string2)+x)] == string2:
      return True
  return False
```