

# PROGRAMMERING-Python

## TENTAMINA

2018-03-13

Kontaktperson: Nicolina Månsson, tel: 0768-530640

### **Tentamensinstruktioner**

#### **Poängsättning**

Tentamina är indelad i två delar

**Del1.** Innehåller uppgifter för betyg 3, totalt 20 poäng. Du måste klara av 14 poäng för att få betyg 3.

**Del2.** Innehåller uppgift för betyg 4 och 5 totalt 10 poäng. Du måste ha fått betyg 3 på Del 1 för att kunna få poäng på Del2.

För betyg 4 krävs det 6 poäng och för betyg 5, 8 poäng.

#### **Redovisning**

Tänk på att vara noggrann och strukturerad. Det är Du som skall visa vad Du kan!

I bilagan finns information du kan behöva.

#### **Hjälpmedel**

Appendix, delas ut av tentavakten.

**Lycka till!**

## Del 1 - Python-Grunder(20p)

1.(1p)Vad blir utskriften när följande kod exekveras?

```
s = "Kontonr 234 innehåller 53 kr"
nytext=""
for tkn in s:
    if tkn.isdigit():
        nytext+=tkn
print(nytext)
```

2.(1p)Vad blir utskriften när följande kod exekveras?

```
counter=0;
sum=0;
while counter <3:
    for i in range(4):
        sum+=i
    counter+=1

print(sum)
```

3.(1p) Ett snabbväxande bakterie förökar sig enligt nedansående formel

$$\text{tot} = \text{init} * 1.5^{\text{tid}}$$

tot- är total antal bakterier  
init- initial antal bakterier  
tid- är tiden i minuter

Skriv ett program som beräknar antalet bakterier. Indata till programmet är hur många bakterier som finns initial samt tiden som har gått (i minuter).

Utdata skall vara antalet bakterier efter en tid.

4.(5p) a) Definiera en funktion **get\_operator(expression) {}** som tar som argument ett enkelt aritmetiskt uttryck som text (t.ex. "23+56", eller "4\*5") och returnerar operator som finns i uttrycket(alltså antingen +, -, \* eller ^).  
**Tips!** Använd lämplig funktion från String för att ta reda på vilken operator som finns i uttrycket

b) Definiera en funktion **calculate** (**expression**) som tar som argument en sträng innehållande ett aritmetiskt uttryck och returnerar resultatet av uttrycket. Funktionen `calculate()` skall använda `get_operator()`.

**Tips!** Resultatet från `get_operator()` skall användas som argument till funktionen `split()` som kommer att returnera en lista med operanderna. Därefter skall beräkningen göras.

c. Anropa metoden **calculate()** i ett program där användaren får mata in aritmetiska uttryck. Programmet beräknar uttrycket och skriver ut resultatet. Detta skall göras i en loop tills användaren matar in ordet "exit". Då skall programmet avslutas.

5.(4p) a) Skriv en funktion **create\_list(size)** som skapar och fyller en lista med numeriska slumpade heltalvärde i intervallet 1-50. Listan storlek skall vara `size`. Funktionen skall returnera listan.

b)Skriv en funktion **larger\_than(list, nr)** som tar som argument en lista och ett tal och returnerar en annan lista som innehåller alla talen från `list` som är större än argumenten `nr`.

6.(4p)Anta att du har filen `klasslista` som innehåller studentnamn och poäng i följande format:

```
Anna Karenina  12 hp
.....
Kalle Svensson 35 hp
.....
Albin Ratt     3 hp
```

a)Skriv en funktion **format\_ok( line )** som tar som argument en sträng motsvarande en rad från filen och kontrollerar om formatet är korrekt. Returnerar `true` om formatet är ok annars `false`.

Korrekt format är en sträng som innehåller 4 ord(förnamn, efternamn, poäng och hp). Mellan namn och poäng har vi en `tab(\t)`. Mellan de andra orden har vi `space`. Funktionen returnerar `true` om formatet är ok annars `false`. Namnet skall bestå enbart av ett förnamn och ett efternamn, alltså 2 ord. Det tredje ordet skall innehålla bara digits. Det fjärde ordet är "hp".

c) Definiera en funktion **krypt(line)** som tar som argument en sträng motsvarande en rad från filen och skriver ut den men med namnet krypterad, resten av data förblir som det är. Du väljer själv hur du krypterar.

Outputen skulle kunna bli:

Cppc Lcsgpgoc 12 hp

- - - -

7.(4p) Undersök nedanstående klass Account. Anta att den är sparad i filen bankaccount.py

1. Definiera en ny metod i klassen Account som skall heta fee() och som beräknar en "bankavgift". Bankavgiften är 3% av saldot. Metoden skall returnera avgiften. Du behöver inte skriva om hela klassen, bara metoden fee().

2. Testa klassen i ett program där du Skapar två Account-objekt där namnet skall matas in från tangentbordet och saldot skall slumpas i intervallet 100-500. Undersök vilka av de två konton som har fått mest pengar och skriv ut namnet samt saldo för respektive konto.

**class Account:**

```
# The __init__ method accepts an argument for  
# the account's balance and name.  
  
def __init__(self, ibalance, iname):  
    self.__balance = ibalance  
    self.__name=iname  
  
    # The deposit method makes a deposit into the  
# account.  
  
def deposit(self, amount):  
    self.__balance += amount  
  
    # The get_balance method returns the  
# account balance.  
  
def get_balance(self):  
    return self.__balance  
  
def get_name(self):  
    return self.__name
```

## Del 2 -Uppgift för betyg 4 och 5 (10 poäng)

Skriv ett program som simulerar en enkel Internet Bank.  
Operationerna som skall vara möjliga i Internet Banken är:

- loga\_in** . Användaren skall mata in user\_id och login. Om dessa är korrekta skall användaren kunna logga in annars ett felmeddelande visas. Inloggningen kan simuleras genom en "status variabel" som sätts till true samt ett meddelande "Du är inloggad"
- show\_info**. Visa informationen om konto och saldo för respektive användare. Detta skall kunna göras bara om kunden är inloggad.
- withdraw**. Användaren skall mata in kontonummer och summa man vill ta ut. Summan skall tas ut från respektive konto. En användare kan ha flera konton, se filen accounts.txt. Den nya saldo skall sparas i filen när kunden loggar ut.
- loga ut**. Filerna med information uppdateras samt status-variabeln sätts till false.

Programet skall byggas upp med en meny-val.  
Programmet skall delas ut i lämpliga funktioner.  
Utveckla gärna objekt orienterad med Customer-objekt som innehåller attributen user\_id, password, en lista men Account-objekt samt en satus-variabel.  
Account-objekt innehåller attributen account\_nr, user\_id och balance.

**OBS!** Om du är osäkert på hur du skall tolka uppgiften förklara kort din tolkning så vet jag det.

Filen har följande format:

user.txt

```
user_id='John Doe' password='abc123'  
user_id='Alice Bob' password='dlkfj243'
```

accounts.txt

```
account_number='1' user_id='John Doe' balance='12345'  
account_number='12' user_id='Alice Bob' balance='-23.4'  
account_number='32' user_id='Alice Bob' balance='1233'
```