April 5 Python Demonstration Notes

Here are some syntax notes on the following code demonstration at https://replit.com/@SE4003/demo (https://replit.com/@SE4003/demo).

1. A comment is prefaced by a pound sign (#) and does not execute.
2. A *string* is a sequence of characters delimited by single or double quotes.
3. A *list* is a sequence of data values enclosed in brackets and separated by commas. The sequence can be iterated on.
4. A *for loop* will repeat a block of code that is indented beneath the `for` statement:

   ```
   for name in se4003_students:
   ```

   - The block starts on the next line after the end colon in the `for` statement
   - The loop will repeat for each value in the provided iterator
5. The `input` statement will send a prompt to the user and extract the response typed in.
6. The `if else` compound statement will execute the conditional indented blocks of code based on the `if` statement check.

In [12]:

```
# print a string enclosed in quotes
print('Hello SE4003')  # another comment

# print a string variable
today = "April 5, 2022"
print('Hello SE4003 on', today)

# a list of strings that can be iterated on
se4003_students = [
"Auld, Sean",
"Camp, Daniel",
"Daley, Steven",
"Dogum, Gregory",
"Gizas, Ashley",
"Gonzalez, Victor",
"Hill, Benjamin",
"Hogan, Martin",
"Iya, Gabriel",
"Jones Maia, Kristin",
"Kylander, Paul",
"Meisner, Megan",
"Meszaros, Michele",
"Novoa, Jonathan",
"Patterson, Dawn",
"Stokes, Joshua",
"Vermeulen, Suzanne",
"Vey, Nathan",
"Villarreal, Rene",
"Willis, Jerald",
"Wilson, Nicole",
]

# loop through list and assign each value to the variable 'name' in each it
for name in se4003_students:
  print ('Hello', name)

# get user inputs and make variable assignments
name = input('What is your first name? ')
last_name = input('What is your last name? ')
print ('Hello', name, last_name)

# square a number provided by the user and print result
# input statement gets user input
# float function converts string input into floating point number for calcu
number = float(input('What is the number to square? '))
number_squared = number * number
print(number, 'squared = ', number_squared)

# if else conditional statement
number = float(input('What is a number between 0 - 10? '))
if number < 5:
  print("Your number was less than 5")
else:
  print("Your number was 5 or greater")
```

```
Hello SE4003
Hello SE4003 on April 5, 2022
Hello Auld, Sean
Hello Camp, Daniel
Hello Daley, Steven
Hello Dogum, Gregory
Hello Gizas, Ashley
Hello Gonzalez, Victor
Hello Hill, Benjamin
Hello Hogan, Martin
Hello Iya, Gabriel
Hello Jones Maia, Kristin
Hello Kylander, Paul
Hello Meisner, Megan
Hello Meszaros, Michele
Hello Novoa, Jonathan
Hello Patterson, Dawn
Hello Stokes, Joshua
Hello Vermeulen, Suzanne
Hello Vey, Nathan
Hello Villarreal, Rene
Hello Willis, Jerald
Hello Wilson, Nicole
What is your first name? r
What is your last name? m
Hello r m
What is the number to square? 6
6.0 squared =  36.0
What is a number between 0 - 10? 5
Your number was 5 or greater
```

Homework 1 Use Case Diagrams

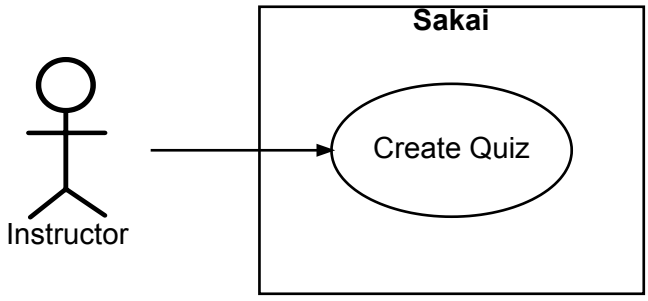
These examples require installation of the PyML library. See http://pyml.fun (http://pyml.fun).

In [2]:
```
import pyml

# modified from http://pyml.fun/examples.py
# system model
system_name = "Sakai"
actors = ['Instructor']
use_cases = ['Create Quiz']
interactions = [('Instructor', 'Create Quiz')]
use_case_relationships = []

# create diagram
pyml.use_case_diagram(system_name, actors, use_cases, interactions, use_cas
```

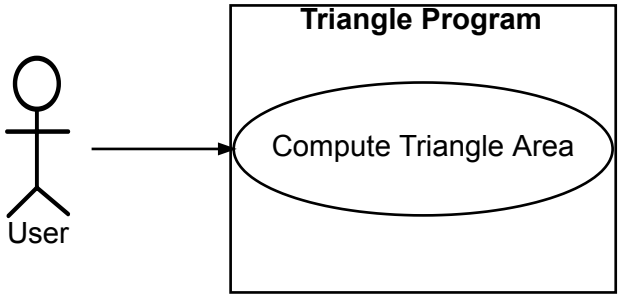
Out[2]:



In [3]:
```
# triangle program
system_name = "Triangle Program"
actors = ['User']
use_cases = ['Compute Triangle Area']
interactions = [('User', 'Compute Triangle Area')]
use_case_relationships = []

# create diagram
pyml.use_case_diagram(system_name, actors, use_cases, interactions, use_cas
```

Out[3]:



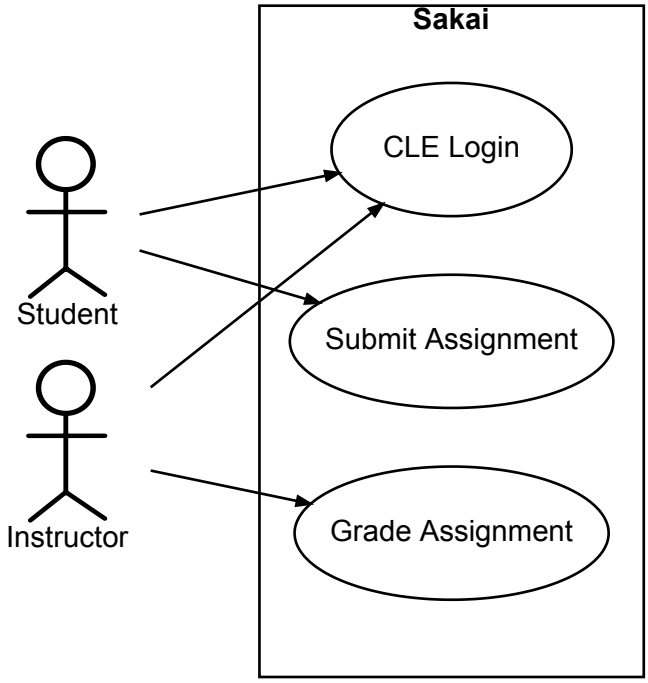
In [11]:
```
number = float(input('What is the number to square? '))
number_squared = number * number
print(number, 'squared = ', number_squared)
```

```
What is the number to square? 5
5.0 squared =  25.0
```

In [4]:
```
system_name = "Sakai"
actors = ['Student', 'Instructor']
use_cases = ['CLE Login', 'Submit Assignment', 'Grade Assignment']
interactions = [('Student', 'CLE Login'), ('Student', 'Submit Assignment'),
use_case_relationships = []

# create diagram
pyml.use_case_diagram(system_name, actors, use_cases, interactions, use_cas
```

Out[4]:



April 19 Tutorial and Demonstrations

Control statements

Control statements are used to decide whether some other “controlled” statements are executed

- `if` statements decide whether or not to execute a group of statements
- `if-else` statements decide which of two groups of statements to execute
- `while` statements execute a group of statements as long as some condition is true
- `for` statements execute a group of statements with a variable taking on a sequence of values

For every kind of control statement:

- The control statement ends in a colon :
- The controlled statements are indented four spaces
- In Python editors, hitting Enter after the colon will automatically indent the following lines.

While loops

A “while loop” has this syntax:

```
while condition:
    # one or more statements
```

It does the following:

1. Checks the condition
2. If it is `True`
    A. execute the statements
    B. start the loop again and check the condition
3. If it is `False` exit the loop

Example:

```
countdown = 10
while countdown >= 0:
    print(countdown)
    countdown = countdown - 1
print("Blast off!")
```

In [3]:
```
# demonstrate while loop to calculate position over time with Euler's integ

time = 0.
dt = .25 # Timestep
position = 50 # Initial position (miles)
velocity = 55 # Velocity (miles/hour)

# print header for time output and initial state
print ("Time Velocity Position  ")
print (f"{time:6.2f} {velocity:6.2f} {position:6.2f} ")

# Drive for 4 hours
while time < 4:
    time = time + dt
    #integrate velocity to get new position: x(t) = x(t-1) + dx/dt(t-1) * d
    position = position + velocity*dt
    print (f"{time:6.2f} {velocity:6.2f} {position:6.2f} ")

print (f"Final position: {position:6.2f}")
```

```
Time Velocity Position
  0.00  55.00  50.00
  0.25  55.00  63.75
  0.50  55.00  77.50
  0.75  55.00  91.25
  1.00  55.00 105.00
  1.25  55.00 118.75
  1.50  55.00 132.50
  1.75  55.00 146.25
  2.00  55.00 160.00
  2.25  55.00 173.75
  2.50  55.00 187.50
  2.75  55.00 201.25
  3.00  55.00 215.00
  3.25  55.00 228.75
  3.50  55.00 242.50
  3.75  55.00 256.25
  4.00  55.00 270.00
Final position: 270.00
```

In [1]:
```
countdown = 10
while countdown >= 0:
    print(countdown)
    countdown = countdown - 1
print("Blast off!")
```

```
10
9
8
7
6
5
4
3
2
1
0
Blast off!
```

In [ ]:
```

```

In [ ]:
```

```