

BIS 420 PROGRAMMING FOR DATA SCIENCE

**PRAJAKTA POHARE
CHAPTER 7 EXERCISE 7.4
ILLINOIS STATE UNIVERSITY**

The built-in function `eval` takes a string and evaluates it using the Python interpreter. For example:

```
>>> eval('1 + 2 * 3')
```

```
7
```

```
>>> import math
```

```
>>> eval('math.sqrt(5)')
```

```
2.2360679774997898
```

```
>>> eval('type(math.pi)')
```

```
<type 'float'>
```

Write a function called `eval_loop` that iteratively prompts the user, takes the resulting input and evaluates it using `eval`, and prints the result.

It should continue until the user enters 'done', and then return the value of the last expression it evaluated.

Output:

```
import math
```

```
def eval_loop():
```

```
    last_result = None
```

```
    while True:
```

```
        user_input = input("Enter an expression (or 'done' to quit): ")
```

```
        if user_input.lower() == 'done':
```

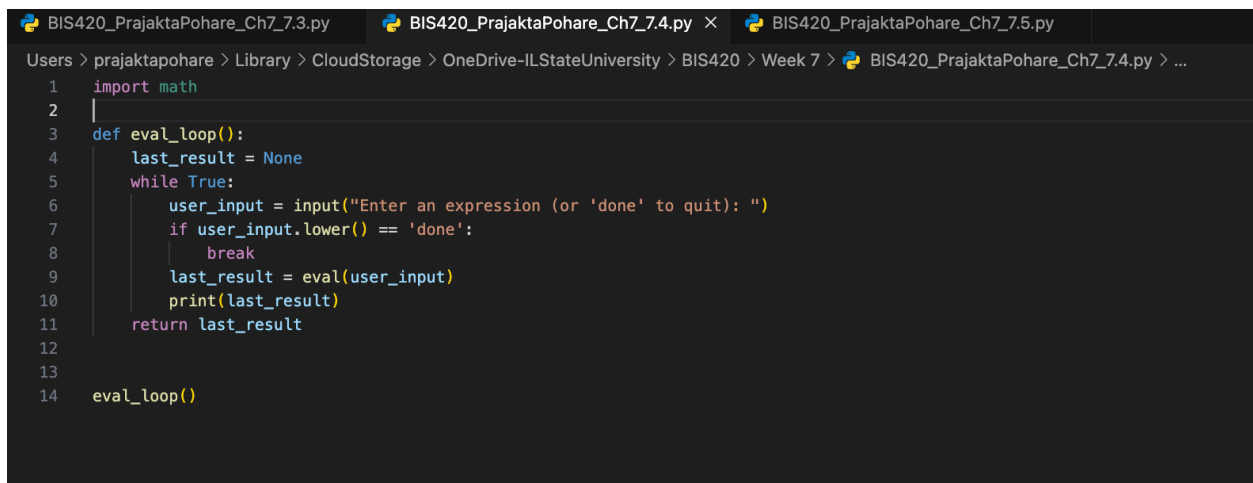
```
            break
```

```
        last_result = eval(user_input)
```

```
        print(last_result)
```

```
    return last_result
```

```
eval_loop()
```



The screenshot shows a code editor with three tabs: BIS420_PrajaktaPohare_Ch7_7.3.py, BIS420_PrajaktaPohare_Ch7_7.4.py (active), and BIS420_PrajaktaPohare_Ch7_7.5.py. The active tab displays the following Python code:

```
1  import math
2  |
3  def eval_loop():
4      last_result = None
5      while True:
6          user_input = input("Enter an expression (or 'done' to quit): ")
7          if user_input.lower() == 'done':
8              break
9          last_result = eval(user_input)
10         print(last_result)
11     return last_result
12
13
14 eval_loop()
```