AI LAB
ASSIGNMENT 1

by

AMMAR JAMIL (01-134231-010)

Submitted to: DR ARSHAD FARHAD
Date: 2/0/2025

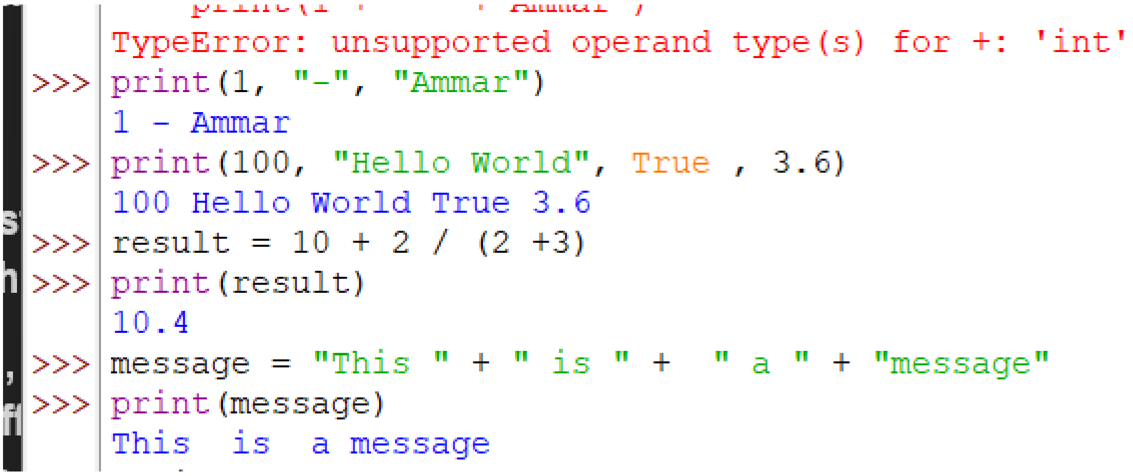
DEPARTMENT OF COMPUTER SCIENCE

BAHRIA UNIVERSITY ISLAMABAD E-8

Task # 1:

Open IDLE and run the following. Write your observations for each point.

1. **Create simple expressions that combine values of different types and math operators.**

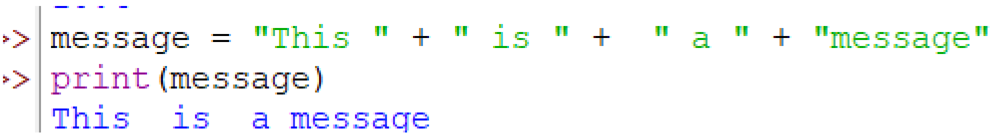


```
    print(1 + " - " + "Ammar")
    TypeError: unsupported operand type(s) for +: 'int'
>>> print(1, "-", "Ammar")
    1 - Ammar
>>> print(100, "Hello World", True , 3.6)
    100 Hello World True 3.6
>>> result = 10 + 2 / (2 +3)
>>> print(result)
    10.4
>>> message = "This " + " is " +  " a " + "message"
>>> print(message)
    This  is  a message
```

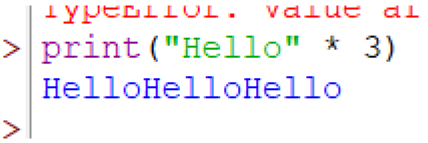
2. **Which operators can be used with string values? Give examples of expressions involving them. What happens when you use other operators?**

SOL:

The operator “+” and “*” operators can be used with string values.

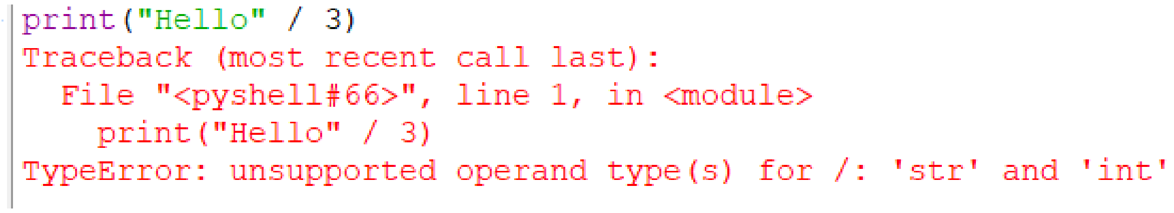


```
>> message = "This " + " is " +  " a " + "message"
>> print(message)
   This  is  a message
```



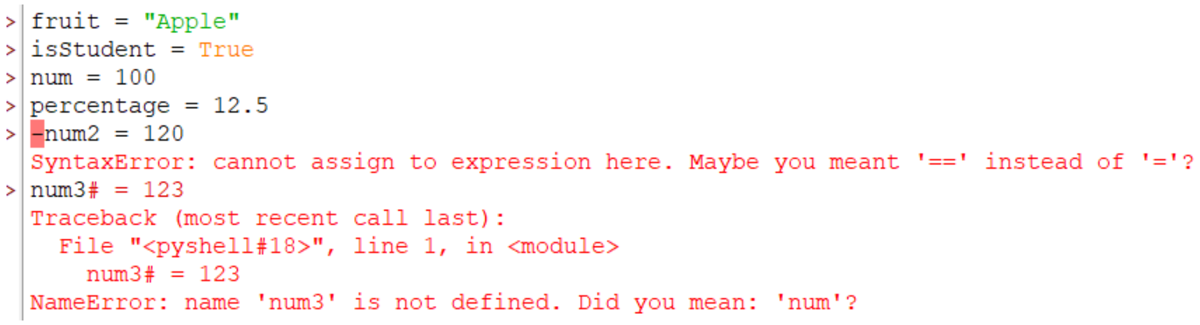
```
  TypeError: value at
> print("Hello" * 3)
  HelloHelloHello
>
```

For the rest of operators it will give TypeError with the description unsupported operand type error



```
print("Hello" / 3)
Traceback (most recent call last):
  File "<pyshell#66>", line 1, in <module>
    print("Hello" / 3)
TypeError: unsupported operand type(s) for /: 'str' and 'int'
```

3. Write a few assignment statements, using as assigned values either literals or expressions. Experiment with different variable names that start with different characters to learn what is allowed and what not.

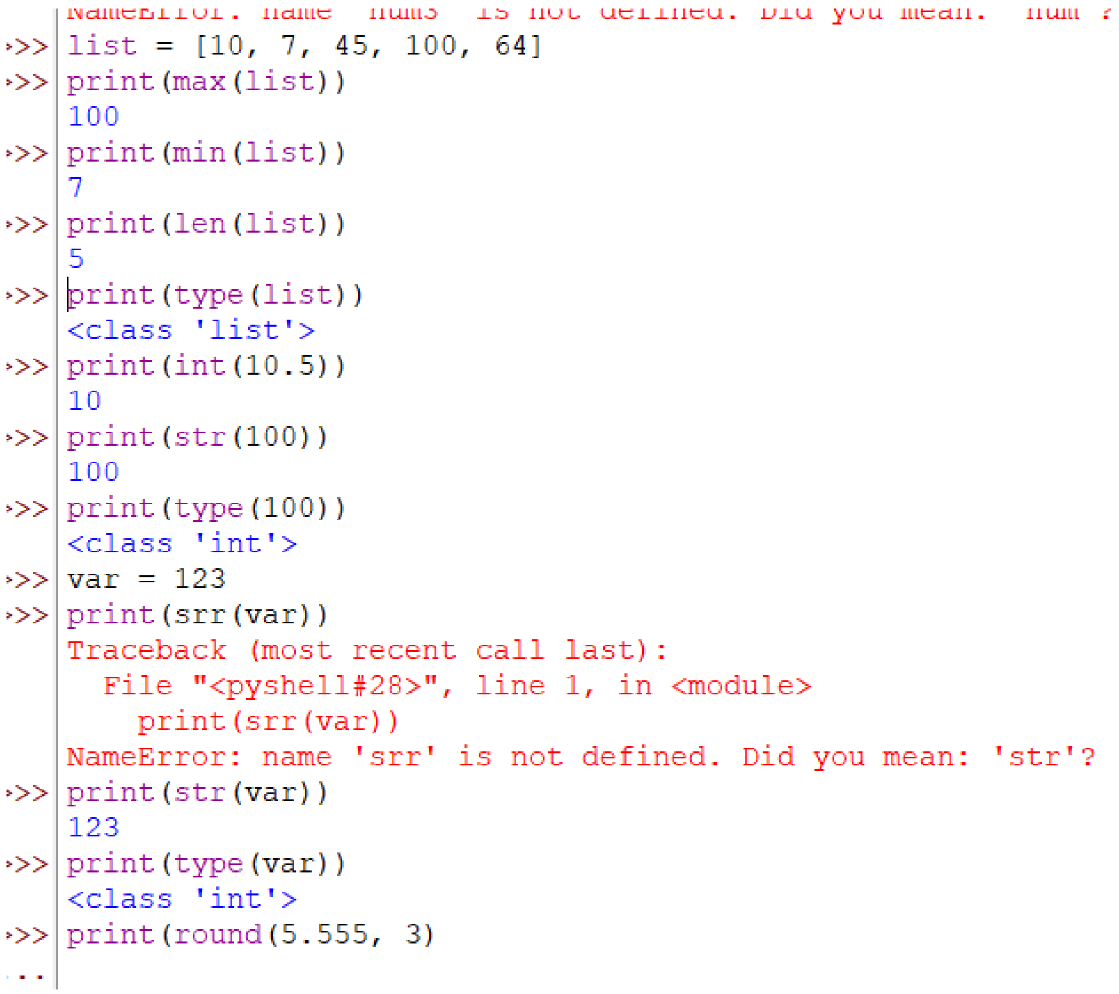


```
> fruit = "Apple"
> isStudent = True
> num = 100
> percentage = 12.5
> -num2 = 120
  SyntaxError: cannot assign to expression here. Maybe you meant '==' instead of '='?
> num3# = 123
  Traceback (most recent call last):
    File "<pyshell#18>", line 1, in <module>
      num3# = 123
  NameError: name 'num3' is not defined. Did you mean: 'num'?
```

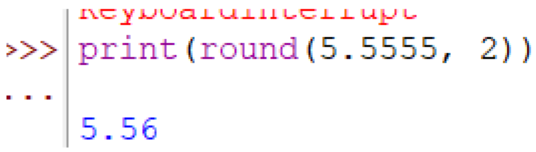
Observation:

Same as C++ there are some rules for naming variables like variable name cannot start with a number or underscore or any other special character except for underscore and character.

4. Perform different function calls of the built-in functions: max, min, len, type, int, str, float, round, print.

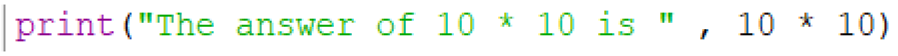


```
    NameError: name 'nums' is not defined. Did you mean: 'num'?
>>> list = [10, 7, 45, 100, 64]
>>> print(max(list))
    100
>>> print(min(list))
    7
>>> print(len(list))
    5
>>> print(type(list))
    <class 'list'>
>>> print(int(10.5))
    10
>>> print(str(100))
    100
>>> print(type(100))
    <class 'int'>
>>> var = 123
>>> print(srr(var))
    Traceback (most recent call last):
      File "<pyshell#28>", line 1, in <module>
        print(srr(var))
    NameError: name 'srr' is not defined. Did you mean: 'str'?
>>> print(str(var))
    123
>>> print(type(var))
    <class 'int'>
>>> print(round(5.555, 3)
...
```



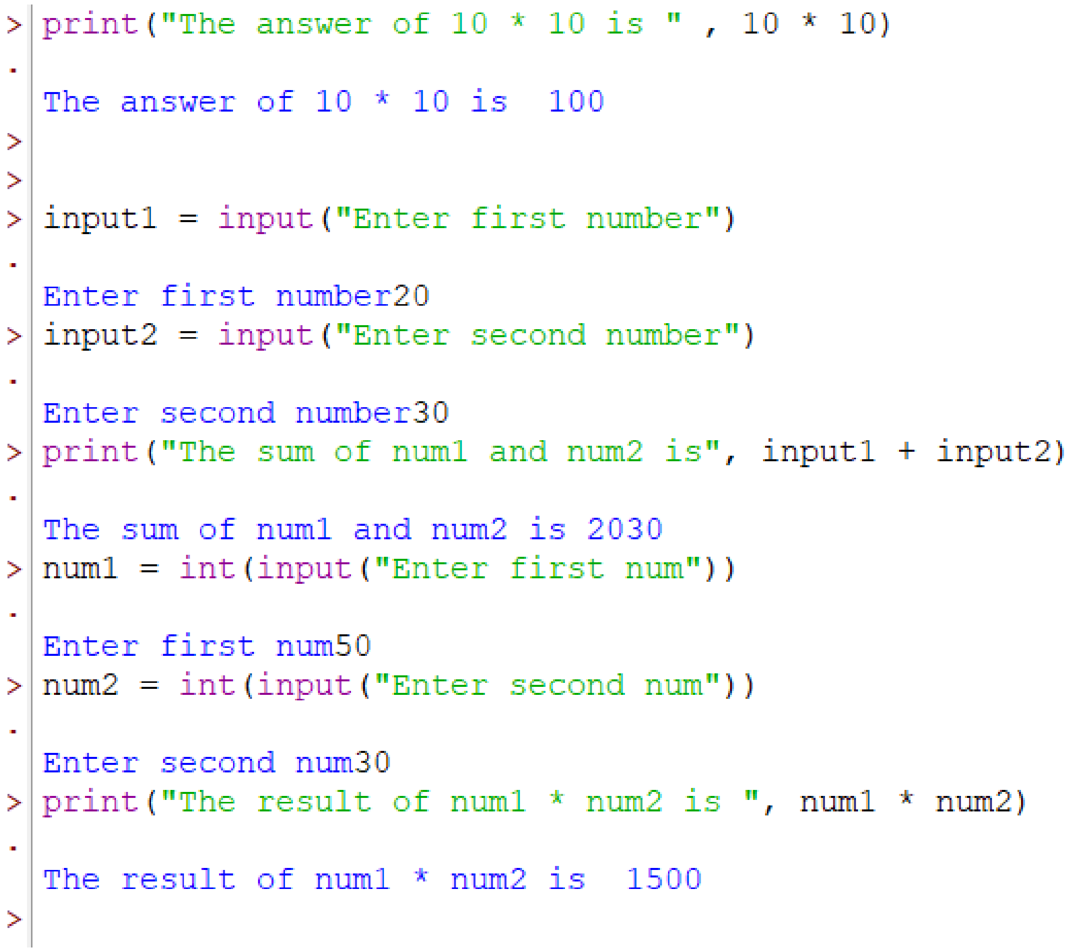
```
    KeyboardInterrupt
>>> print(round(5.5555, 2))
...
    5.56
```

5. Use the function print to display the result of expressions involving string and numerical values.



```
print("The answer of 10 * 10 is " , 10 * 10)
```

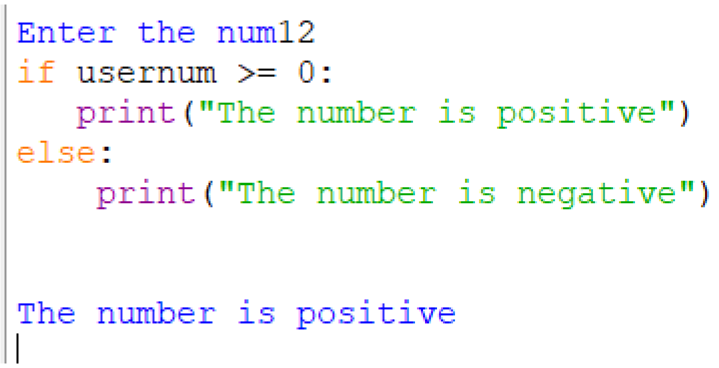
6. Write simple examples that use input to collect values from a user and use them in simple expressions. Remember to convert numerical values.



```
> print("The answer of 10 * 10 is " , 10 * 10)
.
  The answer of 10 * 10 is  100
>
>
> input1 = input("Enter first number")
.
  Enter first number20
> input2 = input("Enter second number")
.
  Enter second number30
> print("The sum of num1 and num2 is", input1 + input2)
.
  The sum of num1 and num2 is 2030
> num1 = int(input("Enter first num"))
.
  Enter first num50
> num2 = int(input("Enter second num"))
.
  Enter second num30
> print("The result of num1 * num2 is ", num1 * num2)
.
  The result of num1 * num2 is  1500
>
```

Task 2:

1. **Check if a number is positive or negative**



```
Enter the num12
if usernum >= 0:
   print("The number is positive")
else:
    print("The number is negative")


The number is positive
```

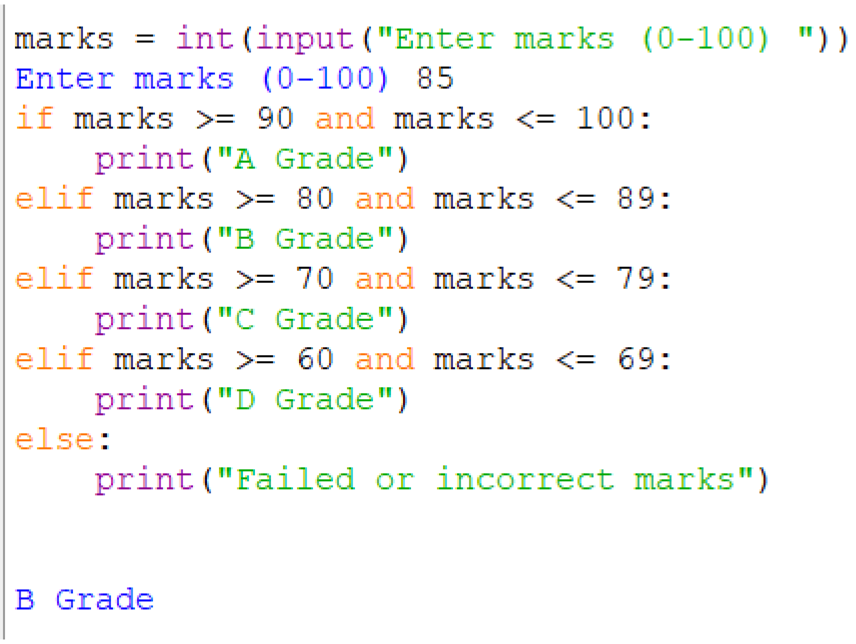
1. **Takes a student's marks as input (typically a number between 0 and 100). Classifies the marks into a grade based on standard grading criteria, such as:**

A: 90–100

B: 80–89

C: 70–79

D: 60–69

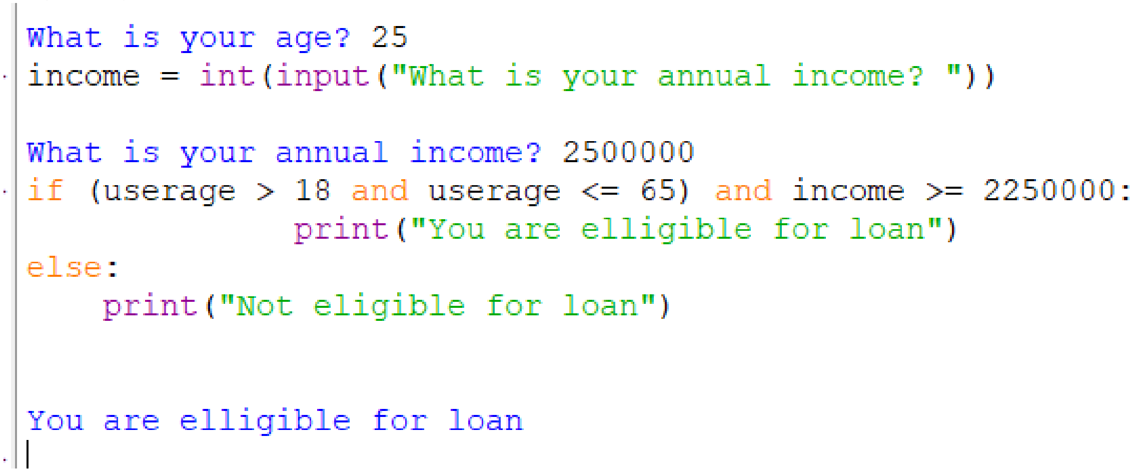


```
marks = int(input("Enter marks (0-100)  "))
Enter marks (0-100) 85
if marks >= 90 and marks <= 100:
    print("A Grade")
elif marks >= 80 and marks <= 89:
    print("B Grade")
elif marks >= 70 and marks <= 79:
    print("C Grade")
elif marks >= 60 and marks <= 69:
    print("D Grade")
else:
    print("Failed or incorrect marks")


B Grade
```

1- Check eligibility for a loan. (consider age, and income)

The core idea of the loan eligibility checker is to prompt the user for relevant information—age and income—and apply predefined rules to determine eligibility. For example, a common set of criteria might require the applicant to be at least 18 years old (the legal age for financial contracts in many regions) and no older than 65 (a typical upper limit due to repayment concerns). Additionally, the income requirement might be set at a minimum threshold, such as 2,250,000 PKR per year, to ensure the applicant has the financial capacity to repay the loan. These criteria can vary depending on the lending institution, so the program can be designed to allow flexibility, such as customizable thresholds or additional factors like credit score or employment status. The students should use Python's input() function for interactive user input, conditional statements (if-elif-else) to evaluate eligibility.



```
What is your age? 25
income = int(input("What is your annual income? "))

What is your annual income? 2500000
if (userage > 18 and userage <= 65) and income >= 2250000:
              print("You are elligible for loan")
else:
    print("Not eligible for loan")


You are elligible for loan
```