Arithmetic Operation

● Which of the following expressions results in SyntaxErrors? Identify the syntax SyntaxErrors and type the reason of error along with the correct code by mentioning itssequence number.

In [ ]:
```
n. 6 * -----------8
o. 8 = people
p. ((((4 ** 3))))
q. (-(-(-(-5))))
r. 4 += 7 / 2
```

In [7]:
```
6 * -----------8
```

Out[7]:
```
-48
```

In [4]:
```
8 = people
The statement print(8 = people) is not valid in Python.
the '=' operator in python is used for assignment, not for equality comparison.
if the value of a variable is equal to 8, we should use the equality comparison operator
```

```
  Cell In[4], line 1
    print(8 = people)
          ^
SyntaxError: expression cannot contain assignment, perhaps you meant "=="?
```

In [12]:
```
8==people
```

Out[12]:
```
True
```

In [8]:
```
((((4 ** 3))))
```

Out[8]:
```
64
```

In [9]:
```
(-(-(-(-5))))
```

Out[9]:
```
5
```

In [10]:
```
4 += 7 / 2
In Python the statement print(4 += 7 / 2) is not valid. The '+=' operator is used for in
but the left operand of the operator must be a variable.
```

```
  Cell In[10], line 1
    4 += 7 / 2
    ^
SyntaxError: 'literal' is an illegal expression for augmented assignment
```

In [11]:
```
x = 4
x += 7 / 2
```

- Using only the techniques you have learned so far, write a program that calculates the square and cube of the numbers from 0 to 10 and uses tabs to print the following table of values:

In [33]:
```
print("Number\tsqaure\tcube")
for Number in range(11):
    square = Number ** 2
    cube = Number ** 3
    print(f"{Number}\t{square}\t{cube}")
```

```
Number  sqaure  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
6       36      216
7       49      343
8       64      512
9       81      729
10      100     1000
```

- Write a program that assigns values to two variables and then swap their values.

In [38]:
```
def swap_values(x, y):
    print("Before swapping:")
    print("x =", x)
    print("y =", y)

    # Swapping the values
    temp = x
    x = y
    y = temp

    print("After swapping:")
    print("x =", x)
    print("y =", y)

def main():
    # Assigning values to variables
    x = 15
    y = 25

    # Calling the swap_values function
    swap_values(x, y)

if __name__ == "__main__":
    main()
```

```
Before swapping:
x = 15
y = 25
After swapping:
x = 25
y = 15
```

Variables and Python Input

●Write a program that inputs one five-digit number, generates its reverse and displays thereverse on screen.(Hint: Make use of the arithmetic operators modulus (%) and floor (//)

In [45]:
```
num = int(input("Enter a five-digit number: "))
if num < 10000 or num > 99999:
    print("Invalid input. Please enter a valid five-digit number.")
else:
    digit_1 = num % 10
    digit_2 = (num // 10) % 10
    digit_3 = (num // 100) % 10
    digit_4 = (num // 1000) % 10
    digit_5 = num // 10000
    reversed_num = digit_1 * 10000 + digit_2 * 1000 + digit_3 * 100 + digit_4 * 10 + dig
    print("The reverse of the number is:", reversed_num)
```

```
Enter a five-digit number: 87654
The reverse of the number is: 45678
```

●Using whatever we have learned so far, write a program that prints a box, an oval, an arrow and a diamond as follows

In [46]:
```
def print_box():
    print("Box:")
    print("------")
    box = "*****\n*   *\n*   *\n*   *\n*****"
    print(box)

def print_oval():
    print("Oval:")
    print("------")
    oval = "  ***  \n *   * \n*     *\n *   * \n  ***  "
    print(oval)

def print_arrow():
    print("Arrow:")
    print("------")
    arrow = "    *\n   ***\n  *****\n    *\n    *"
    print(arrow)

def print_diamond():
    print("Diamond:")
    print("-------")
```

```
    diamond = "   *   \n  ***  \n ***** \n*******\n ***** \n  ***  \n   *   "
    print(diamond)

def main():
    print_box()
    print()
    print_oval()
    print()
    print_arrow()
    print()
    print_diamond()

if __name__ == "__main__":
    main()
```

```
Box:
------
*****
*   *
*   *
*   *
*****

Oval:
------
  ***
 *   *
*     *
 *   *
  ***

Arrow:
------
    *
   ***
  *****
    *
    *

Diamond:
------
   *
  ***
 *****
*******
 *****
  ***
   *
```

Decision Making using Condition Statements

- Write a program that reads three nonzero integers and determines and prints if they could be the sides of a right triangle. (Hint : use Pythagoras theorem you learned in mathematics during FSC or matric , try different combinations of the same integers)

Enter three integers: 3 4 5 The three integers are the sides of a right triangle

In [2]:
```python
def is_right_triangle(a, b, c):
    # Check if any side is zero
    if a == 0 or b == 0 or c == 0:
        return False
    if a ** 2 + b ** 2 == c ** 2 or a ** 2 + c ** 2 == b ** 2 or b ** 2 + c ** 2 == a **
        return True
    else:
        return False

def main():
    a, b, c = map(int, input("Enter three non-zero integers (separated by spaces): ").sp
    if is_right_triangle(a, b, c):
        print("The three integers are the sides of a right triangle.")
    else:
        print("The three integers are not the sides of a right triangle.")

if __name__ == "__main__":
    main()
```

```
Enter three non-zero integers (separated by spaces):  3 4 5
The three integers are the sides of a right triangle.
```

Enter three integers: 9 4 1 The three integers are not the sides of a right triangle

In [4]:
```python
def is_right_triangle(a, b, c):
    # Check if any side is zero
    if a == 0 or b == 0 or c == 0:
        return False
    if a ** 2 + b ** 2 == c ** 2 or a ** 2 + c ** 2 == b ** 2 or b ** 2 + c ** 2 == a **
        return True
    else:
        return False

def main():
    a, b, c = map(int, input("Enter three non-zero integers (separated by spaces): ").sp

    # Check if they could be the sides of a right triangle
    if is_right_triangle(a, b, c):
        print("The three integers are the sides of a right triangle.")
    else:
        print("The three integers are not the sides of a right triangle.")

if __name__ == "__main__":
    main()
```

```
Enter three non-zero integers (separated by spaces): 9 4 1
The three integers are not the sides of a right triangle.
```

● Write a program to prompt for a score between 0.0 and 1.0. If the score is out of range, print an

error message. If the score is between 0.0 and 1.0, print a grade using the following table:

In [6]:
```
def get_grade(score):
    if score < 0.0 or score > 1.0:
        return "Error: Score is out of range."
    elif score >= 0.9:
        return "A"
    elif score >= 0.8:
        return "B"
    elif score >= 0.7:
        return "C"
    elif score >= 0.6:
        return "D"
    else:
        return "F"

def main():
    score = float(input("Enter a score between 0.0 and 1.0: "))

    grade = get_grade(score)
    print("Grade:", grade)

if __name__ == "__main__":
    main()
```

```
Enter a score between 0.0 and 1.0: 0.9
Grade: A
```

Bonus:

Write a program that asks the user for an hour between 1 and 12, asks them to enter am or pm, and asks them how many hours into the future they want to go. Print out what the hourwill be that many hours into the future, printing am or pm as appropriate. An example is shown below.

In [7]:
```
def get_future_hour(current_hour, period, hours_into_future):
    if period.lower() == "pm":
        current_hour += 12
    future_hour = (current_hour + hours_into_future) % 24
    if future_hour == 0:
        future_hour = 12
        future_period = "AM"
    elif future_hour == 12:
        future_period = "PM"
    elif future_hour > 12:
        future_hour -= 12
        future_period = "PM"
    else:
        future_period = "AM"

    return future_hour, future_period
```

```
def main():
    current_hour = int(input("Enter the current hour (1-12): "))
    period = input("Enter AM or PM: ")
    hours_into_future = int(input("Enter the number of hours into the future: "))

    future_hour, future_period = get_future_hour(current_hour, period, hours_into_future

    print("The hour", hours_into_future, "hours into the future will be:", future_hour,

if __name__ == "__main__":
    main()
```

```
Enter the current hour (1-12): 5
Enter AM or PM: PM
Enter the number of hours into the future: 24
The hour 24 hours into the future will be: 5 PM
```

Bonus:

A palindrome is a number or a text phrase that reads the same backwards as forwards. For example, each of the following five-digit integers are palindromes: 12321, 55555, 45554 and 11611.Write a program that reads in a five-digit integer and determines whether or not it is a palindrome.

In [12]:
```
def is_palindrome(number):
    number_str = str(number)
    if number_str == number_str[::-1]:
        return True
    else:
        return False

def main():
    number = int(input("Enter a five-digit number: "))

    if is_palindrome(number):
        print(number, "is a palindrome.")
    else:
        print(number, "is not a palindrome.")

if __name__ == "__main__":
    main()
```

```
Enter a five-digit number: 34567
34567 is not a palindrome.
```

A palindrome is a number or a text phrase that reads the same backwards as forwards. For example, each of the following five-digit integers are palindromes: 12321, 55555, 45554 and 11611. Write a program that reads in a five-digit integer and determines whether or not it is a palindrome. Enter a five-digit number: 18181 is a palindrome

In [11]:
```python
def is_palindrome(number):
    number_str = str(number)
    if number_str == number_str[::-1]:
        return True
    else:
        return False

def main():
    number = int(input("Enter a five-digit number: "))

    if len(str(number)) != 5:
        print("Invalid input. Please enter a five-digit number.")
    elif is_palindrome(number):
        print(number, "is a palindrome.")
    else:
        print(number, "is not a palindrome.")

if __name__ == "__main__":
    main()
```

```
Enter a five-digit number: 24242
24242 is a palindrome.
```

In [ ]:
```python

```