ASSIGNMENT-3 PYTHON SESSION

Q1:What is difference b/w == & === ?

```
The double equals "==" is a loose or abstract comparison operator,
which compares the values of two operands for equality, after converting both operands to a common type.
This means that if the two operands are of different data types, they are coerced into the same type before comparison.
For example, the expression "5 == '5'" would return true because the two operands have the same value,
despite being of different data types.


The triple equals "===" is a strict comparison operator, which compares the values of two operands for equality,
but without type coercion. This means that the comparison will only return true if both operands have the same value and
the same data type.For example, the expression "5 === '5'" would return false because the two operands are of different data
types, even though they have the same value."===" is a strict comparison operator that compares the values of two operands without
converting their types.Using "===" is generally recommended because it avoids unexpected type conversions that could result in
bugs or errors in your code.
```

Q2:Write a program by using if-else and pass statement and continue statement.

In [4]:

```python
for i in range(11):
    if i % 2 == 0:
        print(i)
    else:
        pass # This statement does nothing and the program will move on to the next iteration
```

```
0
2
4
6
8
10
```

In [3]:

```python
for i in range(11):
    if i % 2 != 0:
        continue # This statement skips to the next iteration and does not execute the print statement below
    print(i)
```

```
0
2
4
6
8
10
```

Q3:Write a program by using range() with three parameter start, stop and step.

In [7]:

```python
for k in range (2,15,1):
    print(k)
```

```
2
3
4
5
6
7
8
9
10
11
12
13
14
```

Q4: Write a program using break() with parameter start,stop and step.

In [6]:

```python
for i in range(2, 21, 2):
    print(i)
    if i == 10:  # This statement demonstrates the use of the break keyword to terminate the loop early
        break
```

```
2
4
6
8
10
```

In [ ]:

```python

```