Tutorial=16

Q.1	Perform 7 pattern using class and object (default constructor).

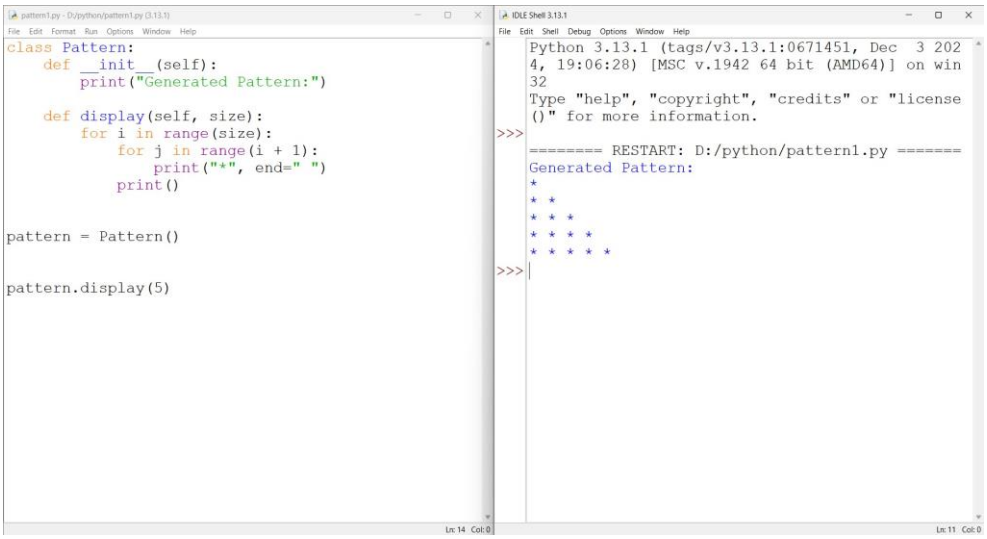
Code	1. class Pattern:
	def __init__(self):
	print("Generated Pattern:")
	
	def display(self, size):
	for i in range(size):
	for j in range(i + 1):
	print("*", end=" ")
	print()
	
	pattern = Pattern()
	
	pattern.display(5)
	
	2. class Pattern:
	def __init__(self):
	print("Generated Pattern:")
	def display(self, size):
	for i in range(size, 0, -1):
	for j in range(i):
	print("*", end=" ")
	print()

```
pattern = Pattern()


pattern.display(5)

3. class NumberPattern:
    def __init__(self):
        print("Generated Number Pattern:")
    def display(self, size):
        for i in range(1, size + 1):
            for j in range(1, i + 1):
                print(j, end=" ")
            print()
pattern = NumberPattern()
pattern.display(5)

4. class NumberPattern:
    def __init__(self):
        print("Generated Number Pattern:")
    def display(self, size):
        for i in range(1, size + 1):
            for j in range(size,0,-1):
                print(j, end=" ")
            print()


pattern = NumberPattern()
```

```
pattern.display(5)

5. class NumberPattern:
    def __init__(self):
        print("Generated Number Pattern:")
    def display(self, size):
        for i in range(1, size + 1):
            for j in range(size, size - i, -1):
                print(j, end=" ")
            print()
pattern = NumberPattern()
pattern.display(5)

6. class NumberPattern:
    def __init__(self):
        print("Generated Reverse Number Pattern:")
    def display(self, size):
        for i in range(size, 0, -1):
            for j in range(size, size - i, -1):
                print(j, end=" ")
            print()


pattern = NumberPattern()
```

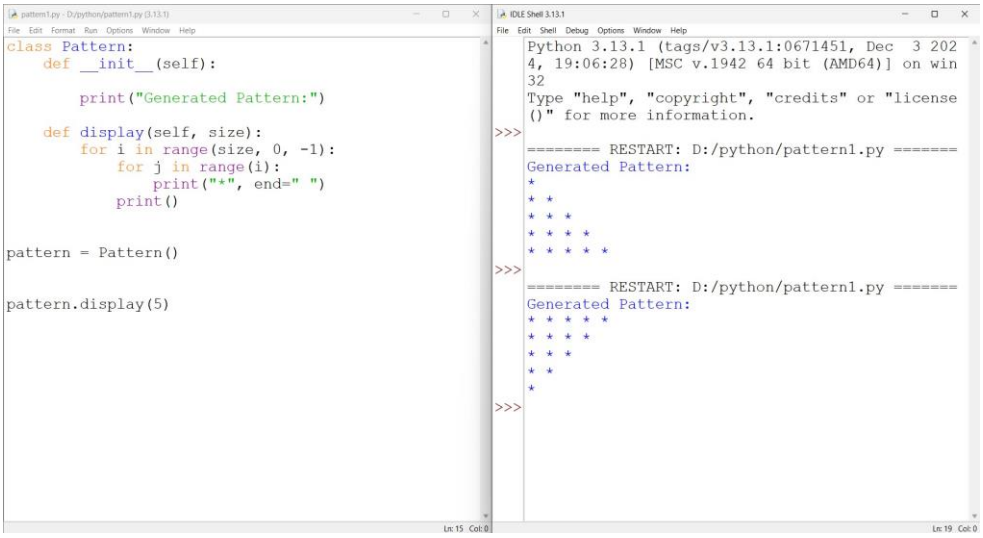
Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

	pattern.display(5)
Output	



```
class Pattern:
    def __init__(self):
        print("Generated Pattern:")

    def display(self, size):
        for i in range(size):
            for j in range(i + 1):
                print("*", end=" ")
            print()


pattern = Pattern()


pattern.display(5)
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 202
4, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license
()" for more information.
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
*
* *
* * *
* * * *
* * * * *
>>>
```



```
class Pattern:
    def __init__(self):

        print("Generated Pattern:")

    def display(self, size):
        for i in range(size, 0, -1):
            for j in range(i):
                print("*", end=" ")
            print()


pattern = Pattern()


pattern.display(5)
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 202
4, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license
()" for more information.
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
*
* *
* * *
* * * *
* * * * *
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
* * * * *
* * * *
* * *
* *
*
>>>
```

Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

pattern1.py - D:/python/pattern1.py (3.13.1)

```python
class NumberPattern:
    def __init__(self):

        print("Generated Number Pattern:")

    def display(self, size):
        for i in range(1, size + 1):
            for j in range(1, i + 1):
                print(j, end=" ")
            print()


pattern = NumberPattern()


pattern.display(5)
```

IDLE Shell 3.13.1

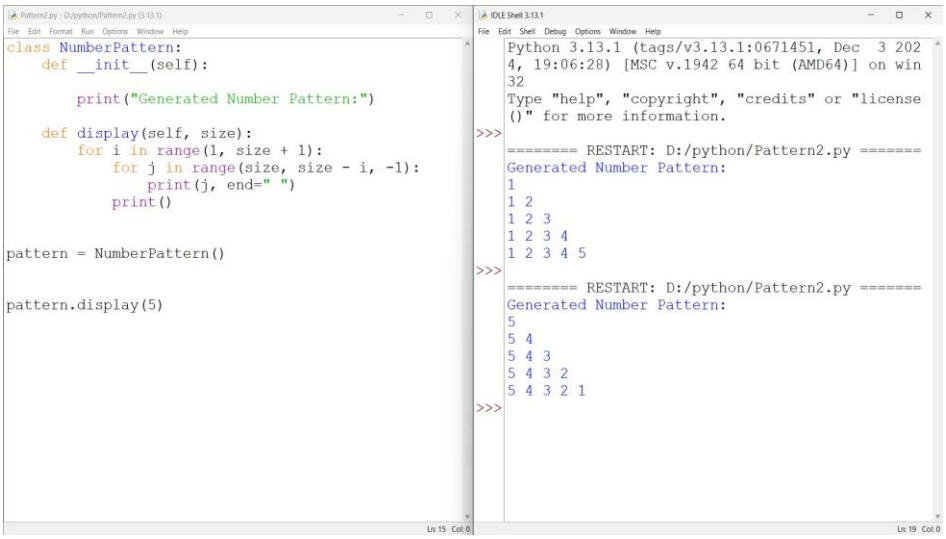
```
32
Type "help", "copyright", "credits" or "license
()" for more information.
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
*
* *
* * *
* * * *
* * * * *
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
* * * * *
* * * *
* * *
* *
*
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Number Pattern:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
>>>
```

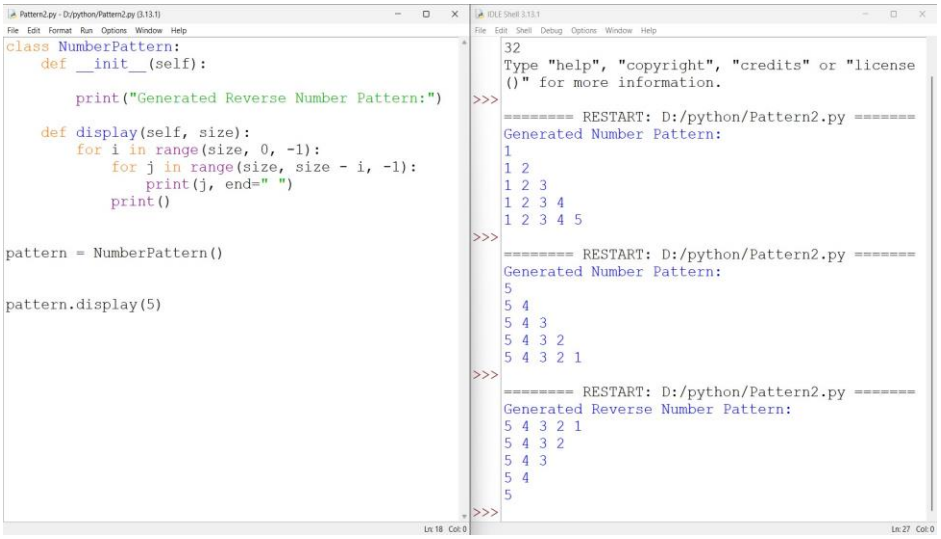
pattern1.py - D:/python/pattern1.py (3.13.1)

```python
class NumberPattern:
    def __init__(self):

        print("Generated Number Pattern:")

    def display(self, size):
        for i in range(1, size + 1):
            for j in range(size,0,-1):
                print(j, end=" ")
            print()


pattern = NumberPattern()


pattern.display(5)
```

IDLE Shell 3.13.1

```
* * *
* * * *
* * * * *
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Pattern:
* * * * *
* * * *
* * *
* *
*
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Number Pattern:
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
>>>
======== RESTART: D:/python/pattern1.py =======
Generated Number Pattern:
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
5 4 3 2 1
>>>
```

<table>
<tr><td></td><td>


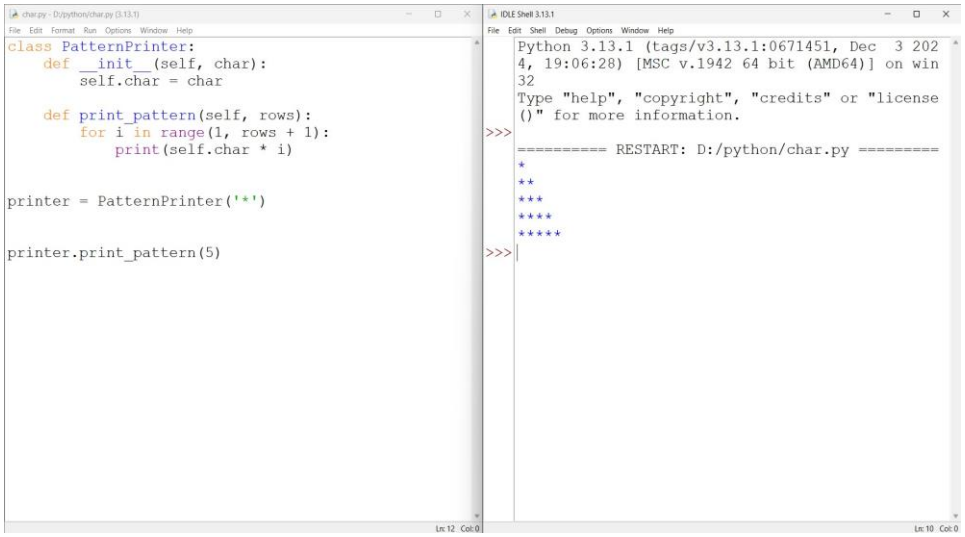
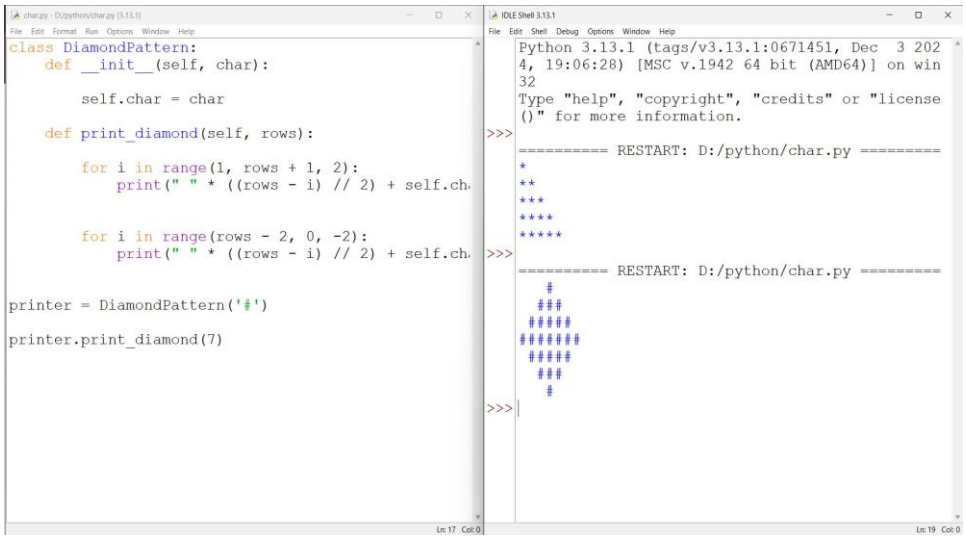


</td></tr>
<tr><td>Q.2</td><td>Perform 3 pattern using string using class and object (Member function passing argument).</td></tr>
<tr><td>Code</td><td>1. class PatternPrinter:
def __init__(self, char):
self.char = char

def print_pattern(self, rows):
for i in range(1, rows + 1):</td></tr>
</table>

```
        print(self.char * i)


printer = PatternPrinter('*')


printer.print_pattern(5)

2. class DiamondPattern:
    def __init__(self, char):

        self.char = char
    def print_diamond(self, rows):
        for i in range(1, rows + 1, 2):
            print(" " * ((rows - i) // 2) + self.char * i)


        for i in range(rows - 2, 0, -2):
            print(" " * ((rows - i) // 2) + self.char * i)


printer = DiamondPattern('#')

printer.print_diamond(7)

3. class SquarePattern:
    def __init__(self, char):
        self.char = char
    def print_square(self, size):
        for _ in range(size):
            print(self.char * size)
```

Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

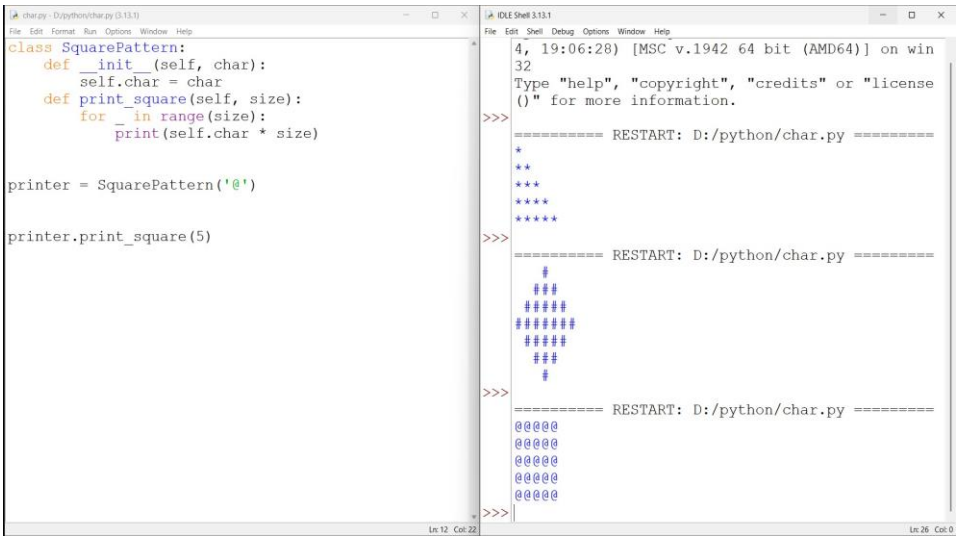
	printer = SquarePattern('@') printer.print_square(5)
Output	 

```
class PatternPrinter:
    def __init__(self, char):
        self.char = char

    def print_pattern(self, rows):
        for i in range(1, rows + 1):
            print(self.char * i)


printer = PatternPrinter('*')


printer.print_pattern(5)
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/python/char.py =========
*
**
***
****
*****
>>>
```

```
class DiamondPattern:
    def __init__(self, char):

        self.char = char

    def print_diamond(self, rows):

        for i in range(1, rows + 1, 2):
            print(" " * ((rows - i) // 2) + self.ch

        for i in range(rows - 2, 0, -2):
            print(" " * ((rows - i) // 2) + self.ch


printer = DiamondPattern('#')

printer.print_diamond(7)
```

```
Python 3.13.1 (tags/v3.13.1:0671451, Dec  3 2024, 19:06:28) [MSC v.1942 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: D:/python/char.py =========
*
**
***
****
*****
>>>
========== RESTART: D:/python/char.py =========
   #
  ###
 #####
#######
 #####
  ###
   #
>>>
```



Q.3

Perform program in Class and Object

- Fibonacci Series
- Prime Number
- Factorial
- Even and Odd
- Table of 5
- Userinput (Medical details)
- Student Details
- Bank Details

Code

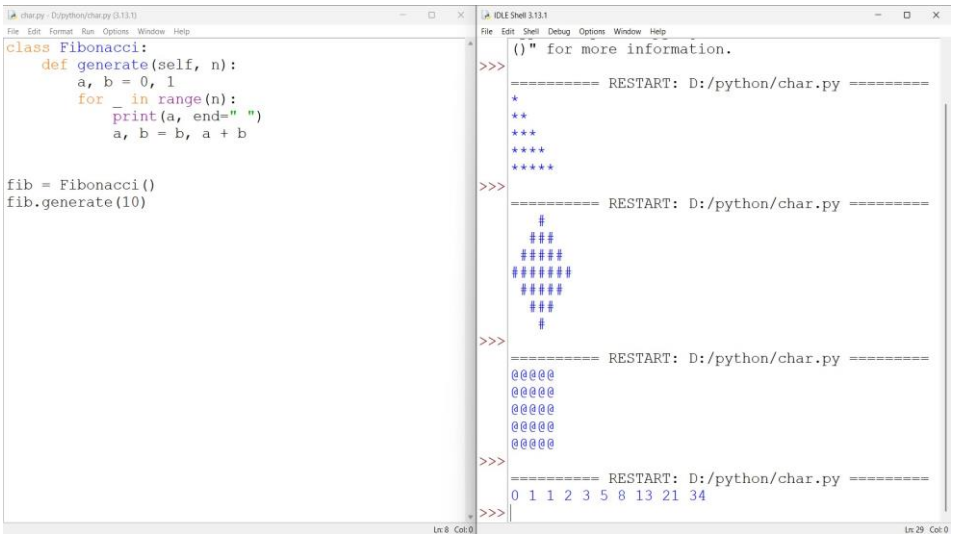
```
class Fibonacci:
    def generate(self, n):
        a, b = 0, 1
        for _ in range(n):
            print(a, end=" ")
            a, b = b, a + b


fib = Fibonacci()
fib.generate(10)
```

```
2. class PrimeCheck:
    def is_prime(self, num):
        if num < 2:
            return False
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                return False
        return True


prime = PrimeCheck()
print(prime.is_prime(5))

3. class Factorial:
    def calculate(self, num):
        fact = 1
        for i in range(1, num + 1):
            fact *= i
        return fact


fact_obj = Factorial()
print(fact_obj.calculate(5))

4. class NumberCheck:
    def check(self, num):
        return "Even" if num % 2 == 0 else "Odd"


num_check = NumberCheck()
print(num_check.check(10))

5. class MedicalDetails:
    def __init__(self):
        self.name = input("Enter the Name: ")
```

```
        self.age = input("Enter the Age: ") self.blood_group
        = input("Enter the Blood Group: ")

    def display(self):
        print(f"Name: {self.name}, Age: {self.age}, Blood
Group: {self.blood_group}")


person = MedicalDetails()
person.display()

6. class Student:
    def __init__(self):
        self.name = input("Enter the Name: ")
        self.roll_no = input("Enter the Roll No: ")
        self.marks = input("Enter the Marks: ")

    def display(self):
        print(f"Name: {self.name}, Roll No: {self.roll_no},
Marks: {self.marks}")


student = Student()
student.display()

7. class BankAccount:
    def __init__(self):
        self.name = input("Enter Account Holder Name: ")
        self.acc_no = input("Enter Account Number: ")
        self.balance = input("Enter Balance: ")

    def display(self):
        print(f"Account Holder: {self.name}, Account No:
{self.acc_no}, Balance: {self.balance}")
```

```
account = BankAccount()
account.display()

8.class Table:
    def print_table(self, num):
        for i in range(1, 11):
            print(f"{num} x {i} = {num * i}")


table = Table()
table.print_table(5)
```

Output



Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

```python
class PrimeCheck:
    def is_prime(self, num):
        if num < 2:
            return False
        for i in range(2, int(num ** 0.5) + 1):
            if num % i == 0:
                return False
        return True


prime = PrimeCheck()
print(prime.is_prime(5))
```

```
****
*****
>>>
========== RESTART: D:/python/char.py =========
   #
  ###
 #####
#######
 #####
  ###
   #
>>>
========== RESTART: D:/python/char.py =========
@@@@@
@@@@@
@@@@@
@@@@@
@@@@@
>>>
========== RESTART: D:/python/char.py =========
0 1 1 2 3 5 8 13 21 34
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
True
>>>
```

```python
class Factorial:
    def calculate(self, num):
        fact = 1
        for i in range(1, num + 1):
            fact *= i
        return fact


fact_obj = Factorial()
print(fact_obj.calculate(5))
```

```
========== RESTART: D:/python/char.py =========
   #
  ###
 #####
#######
 #####
  ###
   #
>>>
========== RESTART: D:/python/char.py =========
@@@@@
@@@@@
@@@@@
@@@@@
@@@@@
>>>
========== RESTART: D:/python/char.py =========
0 1 1 2 3 5 8 13 21 34
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
120
>>>
```

Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

```python
class NumberCheck:
    def check(self, num):
        return "Even" if num % 2 == 0 else "Odd"


num_check = NumberCheck()
print(num_check.check(10))
```

```
 #####
#######
 #####
  ###
   #
>>>
========== RESTART: D:/python/char.py =========
@@@@@
@@@@@
@@@@@
@@@@@
@@@@@
>>>
========== RESTART: D:/python/char.py =========
0 1 1 2 3 5 8 13 21 34
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
120
>>>
========== RESTART: D:/python/char.py =========
Even
>>>
```

```python
class Table:
    def print_table(self, num):
        for i in range(1, 11):
            print(f"{num} x {i} = {num * i}")

table = Table()
table.print_table(5)
```

```
>>>
========== RESTART: D:/python/char.py =========
0 1 1 2 3 5 8 13 21 34
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
120
>>>
========== RESTART: D:/python/char.py =========
Even
>>>
========== RESTART: D:/python/char.py =========
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
>>>
```

Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

```python
class MedicalDetails:
    def __init__(self):
        self.name = input("Enter the Name: ")
        self.age = input("Enter  the Age: ")
        self.blood_group = input("Enter the Blood G

    def display(self):
        print(f"Name: {self.name}, Age: {self.age},


person = MedicalDetails()
person.display()
```

```
>>>
========== RESTART: D:/python/char.py =========
True
>>>
========== RESTART: D:/python/char.py =========
120
>>>
========== RESTART: D:/python/char.py =========
Even
>>>
========== RESTART: D:/python/char.py =========
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
>>>
========== RESTART: D:/python/char.py =========
Enter the Name: Meet
Enter  the Age: 20
Enter the Blood Group: B+
Name: Meet, Age: 20, Blood Group: B+
>>>
```

```python
class Student:
    def __init__(self):
        self.name = input("Enter the Name: ")
        self.roll_no = input("Enter the Roll No: ")
        self.marks = input("Enter the Marks: ")

    def display(self):
        print(f"Name: {self.name}, Roll No: {self.r


student = Student()
student.display()
```

```
>>>
========== RESTART: D:/python/char.py =========
Even
>>>
========== RESTART: D:/python/char.py =========
5 x 1 = 5
5 x 2 = 10
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
>>>
========== RESTART: D:/python/char.py =========
Enter the Name: Meet
Enter  the Age: 20
Enter the Blood Group: B+
Name: Meet, Age: 20, Blood Group: B+
>>>
========== RESTART: D:/python/char.py =========
Enter the Name: Dhruv
Enter the Roll No: 28
Enter the Marks: 200
Name: Dhruv, Roll No: 28, Marks: 200
>>>
```

Name: Mohmadhusen Khimani Enrollment No: 22FOTCA11071 Roll No. 14 6BCAB

```python
class BankAccount:
    def __init__(self):
        self.name = input("Enter Account Holder Nam
        self.acc_no = input("Enter Account Number:
        self.balance = input("Enter Balance: ")

    def display(self):
        print(f"Account Holder: {self.name}, Accoun


account = BankAccount()
account.display()
```

```
5 x 3 = 15
5 x 4 = 20
5 x 5 = 25
5 x 6 = 30
5 x 7 = 35
5 x 8 = 40
5 x 9 = 45
5 x 10 = 50
>>>
========== RESTART: D:/python/char.py =========
Enter the Name: Meet
Enter  the Age: 20
Enter the Blood Group: B+
Name: Meet, Age: 20, Blood Group: B+
>>>
========== RESTART: D:/python/char.py =========
Enter the Name: Dhruv
Enter the Roll No: 28
Enter the Marks: 200
Name: Dhruv, Roll No: 28, Marks: 200
>>>
========== RESTART: D:/python/char.py =========
Enter Account Holder Name: Jay
Enter Account Number: 12424343421
Enter Balance: 70000
Account Holder: Jay, Account No: 12424343421, B
alance: 70000
>>>
```