AMAN GUPTA 17 FYIT OOP ASSIGNMENT 2

ASSIGNMENT 2

1) Python program to show use of + operator for different purposes.
   CODE:

dddd.py - C:/Users/Aman Gupta/dddd.py (3.9.2)

File Edit Format Run Options Window Help

```
#EXAMPLE 1
# Python program to show use of
# + operator for different purposes

print(2 + 3)

# concatenate two strings
print("FY"+"IT")

# Product two numbers
print(2 * 3)

# Repeat the string
print("HELLO "*5)
```

OUTPUT:

IDLE Shell 3.9.2

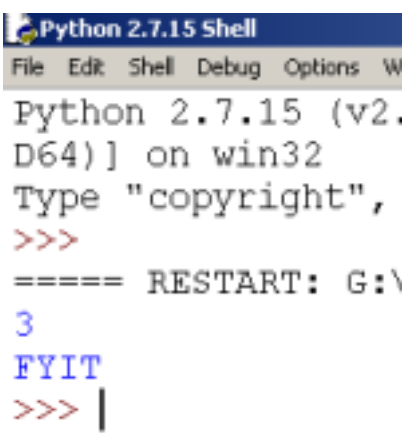
File Edit Shell Debug Options Window Help

```
Python 3.9.2 (tags/v3.9.2:1a79785, Feb 19 2021, 13:44:55) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==================== RESTART: C:/Users/Aman Gupta/dddd.py ====================
5
FYIT
6
HELLO HELLO HELLO HELLO HELLO
>>>
```

2) Python Program illustrate how to overload an binary + operator
CODE:

```
#EXAMPLE 2
#Code 1:

# Python Program illustrate how
# to overload an binary + operator
class S:
    def _init_(self, a):
        self.a = a

    # adding two objects
    def _add_(self, o):
        return self.a + o.a
ob1 = S(1)
ob2 = S(2)
ob3 = S("FY")
ob4 = S("IT")

print(ob1 + ob2)
print(ob3 + ob4)
```

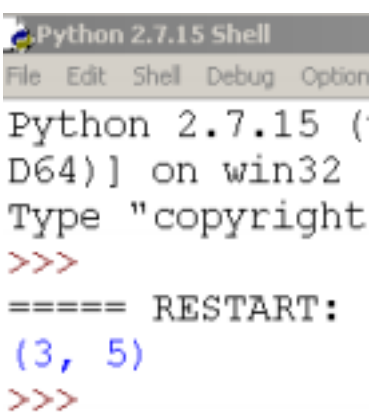
OUTPUT:



3) Python Program to perform addition of two complex numbers using binary + operator overloading.

```
#EXAMPLE 3
#Code 2:

# Python Program to perform addition
# of two complex numbers using binary
# + operator overloading.

class complex:
    def _init_(self, a, b):
        self.a = a
        self.b = b

     # adding two objects
    def _add_(self, other):
        return self.a + other.a, self.b + other.b

    def _str_(self):
        return self.a, self.b

Ob1 = complex(1,2)
Ob2 = complex(2,3)
Ob3 = Ob1 + Ob2
print(Ob3)
```

OUTPUT:



4) Python program to overload a comparison operators

```
# Python program to overload
# a comparison operators

class P:
    def _init_(self, a):
        self.a = a
    def _gt_(self, other):
        if(self.a>other.a):
            return True
        else:
            return False
ob1 = S(2)
ob2 = S(3)
if(ob1>ob2):
    print("ob1 is greater than ob2")
else:
    print("ob2 is greater than ob1")
```

OUTPUT:

```
Python 2.7.15 Shell
File  Edit  Shell  Debug  Options  Window  Help
Python 2.7.15 (v2.7.15:ca079
D64)] on win32
Type "copyright", "credits"
>>>
===== RESTART: G:\CLG\OOP\Pr
ob2 is greater than ob1
>>>
```

5) Python program to overload equality and less than operators

```
# Python program to overload equality
# and less than operators

class P:
    def _init_(self, a):
        self.a = a
    def _lt_(self, other):
        if(self.a<other.a):
            return "obl is lessthan ob2"
        else:
            return "ob2 is less than obl"
    def _eq_(self, other):
        if(self.a == other.a):
            return "Both are equal"
        else:
            return "Not equal"

obl = S(2)
ob2 = S(3)
print(obl < ob2)

ob3 = S(4)
ob4 = S(4)
print(obl == ob2)
```

OUTPUT:

```
Python 2.7.15 Shell
File  Edit  Shell  Debug  Options  Window  Hel
Python 2.7.15 (v2.7.15:
D64)] on win32
Type "copyright", "cred
>>>
===== RESTART: G:\CLG\C
ob1 is lessthan ob2
Not equal
>>> |
```