Polymorphism

Executable Code:

```
class Specialstring:
    def __len__(self):
        return 10


# Driver's code
if __name__ == "__main__":
    string = Specialstring()
    print(len(string))
    print("fly with wings")
    print("fly with fuel")


class Bird:
    def fly(self):
        print("fly with wings")


class Airplane:
    def fly(self):
        print("fly with fuel")


class Fish:
    def swim(self):
```

```
        print("Dolphins swim in sea")



for obj in Bird(), Airplane(), Fish():
    if hasattr(obj, 'fly'):
        obj.fly()
    else:
        print("Cannot fly")


print(10 + 15)
s1 = "Red"
s2 = "Fort"
print(s1 + s2)
a = [10, 20, 30]
b = [5, 15, -10]
print(a + b)


class BookX:
    def __init__(self, pages):
        self.pages = pages


class BookY:
    def __init__(self, pages):
        self.pages = pages


b1 = BookX(30)
```

```
b2 = BookY(20)
print('Total Pages=', b1.pages + b2.pages)


class BookX:
    def __init__(self, pages):
        self.pages = pages

    def __add__(self, other):
        return self.pages + other.pages

b1 = BookX(10)
b2 = BookX(15)
print('Total Pages=', b1 + b2)


class A:
    def __init__(self, a):
        self.a = a

    def __add__(self, o):
        return self.a + o.a

ob1 = A(1)
ob2 = A(2)
ob3 = A("Hello")
ob4 = A("World")
print(ob1 + ob2)
```

```
print(ob3 + ob4)


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

    def __add__(self, other):
        return self.a + other.a, self.b + other.b

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


class Point:
    def __init__(self, x=0, y=0):
        self.x = x
        self.y = y

    def __str__(self):
        return "({0},{1})".format(self.x, self.y)

    def __lt__(self, other):
        self_mag = (self.x ** 2) + (self.y ** 2)
        other_mag = (other.x ** 2) + (other.y ** 2)
```

```
            return self_mag < other_mag


p1 = Point(1, 1)

p2 = Point(-2, -3)

p3 = Point(1, -1)

print(p1 < p2)

print(p2 < p3)

print(p1 < p3)


class Student():
    def __init__(self, r_no, name, age, marks):
            self.r_no = r_no
            self.name = name
            self.age = age
            self.marks = marks


    def displayStudent(self):
            print("Roll no:", self.r_no, "Name:", self.name, ", Age:", self.age, ", Marks:", self.marks)


    def __str__(self):
            return "({0},{1},{2},{3})".format(self.r_no, self.name, self.age, self.marks)


    def __eq__(self, other):
            if self.marks == other.marks:
                    return self.marks == other.marks
```

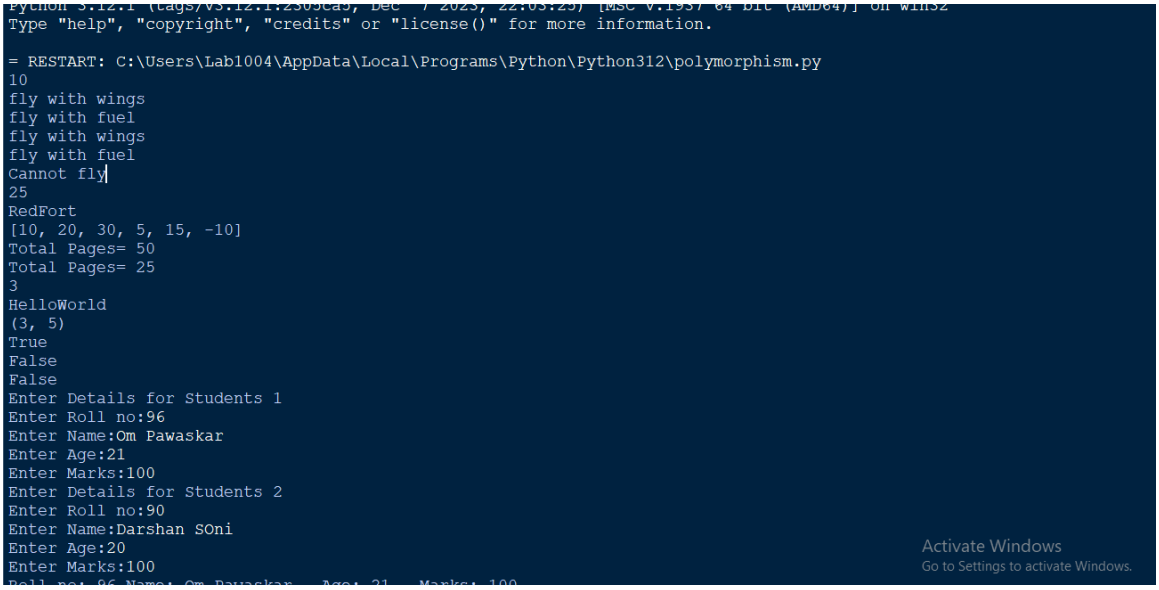
```
stu = []
for i in range(1, 3):
    print("Enter Details for Students %d" % (i))
    r_no = int(input("Enter Roll no:"))
    name = input("Enter Name:")
    age = int(input("Enter Age:"))
    marks = input("Enter Marks:")
    stu.append(Student(r_no, name, age, marks))

for s in stu:
    s.displayStudent()

class Nikhil:
    def sum(self, a=None, b=None, c=None):
        if a is not None and b is not None and c is not None:
            print("Sum of Three=", a + b + c)
        elif a is not None and b is not None:
            print("Sum Of two=", a + b)
        else:
            print('Please enter two or three Argument')

m = Nikhil()
m.sum(10, 15, 20)
m.sum(10.5, 22.5)
m.sum(10)
```

```
class Employee:
    def message(self):
        print('This message is from Employee Class')

class Department(Employee):
    def message(self):
        print('This Department class is inherited from Employee')

emp = Employee()
emp.message()
print('------------')
dept = Department()
dept.message()

class Employee:
    def message(self):
        print('This message is from Employee Class')

class Department(Employee):
    def message(self):
        print('This Department class is inherited from Employee')

class Sales(Department):
    def message(self):
        print('This Sales class is inherited from Employee')
```

```
emp = Employee()
emp.message()
print('------------')
dept = Department()
dept.message()
print('------------')
sl = Sales()
sl.message()


class Employee:
    def add(self, a, b):
        print('The Sum of Two = ', a + b)


class Department(Employee):
    def add(self, a, b, c):
        print('The Sum of Three = ', a + b + c)


emp = Employee()
emp.add(10, 20)
print('------------')
dept = Department()
dept.add(50, 130, 90)


class Employee:
    def message(self):
        print('This message is from Employee Class')
```

```
class Department(Employee):
    def message(self):
        Employee.message(self)
        print('This Department class is inherited from Employee')

emp = Employee()
emp.message()
```

Output



```
True
False
False
Enter Details for Students 1
Enter Roll no:96
Enter Name:Om Pawaskar
Enter Age:21
Enter Marks:100
Enter Details for Students 2
Enter Roll no:90
Enter Name:Darshan SOni
Enter Age:20
Enter Marks:100
Roll no: 96 Name: Om Pawaskar , Age: 21 , Marks: 100
Roll no: 90 Name: Darshan SOni , Age: 20 , Marks: 100
Sum of Three= 45
Sum Of two= 33.0
Please enter two or three Argument
This message is from Employee Class
------------
This Department class is inherited from Employee
This message is from Employee Class
------------
This Department class is inherited from Employee
------------
This Sales class is inherited from Employee
The Sum of Two =  30
------------
The Sum of Three =  270
This message is from Employee Class
```