Assignment - 25 OOPs and Inheritance

1. Write a python script to create a Profile class with 3 attributes (name, email, age).

```
class profile:
    def __init__(self):
        self.name=input("enter the name  ")
        self.age=int(input("enter the age  "))
        self.email=input("enter the mail  ")
class inherit(profile):
    def show_details(self):
        print(self.name,self.age,self.email)

stu=inherit()
#p=profile()
stu.show_details()
```

2. Write a python script to update the above Profile class with encapsulation.

```
class profile:
    def __init__(self):
        print("enter the name,age,email")
        self.name=input()
        self.age=int(input())
        self.email=input()
    def get_name(self):
        print("\n\n",self.name)
    def get_age(self):
        print(self.age)
    def get_email(self):
        print(self.email)
pro=profile()
pro.get_name()
pro.get_age()
pro.get_email()
```

3. Write a python script to update 2nd Question, change email and age to __email and __age.

```
class profile:
    def __init__(self):
        self.name=input("enter the name\n")
        self.__age=10
        self.__email="kasmeerbhalu01@gmail.com"
    def get_name(self):
        print()
        print(self.name,)
    def get_age(self):
        return self.__age
    def get_email(self):
        return self.__email
    def change_age_email(self,age,email):
        self.__age=age
        self.__email=email

pro=profile()
pro.get_name()
pro.change_age_email(25,'rohi20@gmail.com')
print(pro.get_age())
print(pro.get_email())
#print(pro._profile__age)
```

4. Write a python script to update 2nd Question, add a class variable (platform) and create a classmethod to access it.

```
class profile:
    def __init__(self):
        print("enter the name,age,email")
        self.name=input()
        self.age=int(input())
        self.email=input()
    def get_name(self):
        print("\n\n",self.name)
    def get_age(self):
        print(self.age)
    def get_email(self):
        print(self.email)
    @classmethod
    def access_platform(cls):
        cls.plateform=input("enter the plateform")
        print(cls.plateform)
pro=profile()
pro.get_name()
pro.get_age()
pro.get_email()
pro.access_platform()
```

5. Write a python script to create a Calculator class with 2 methods for adding and subtracting 2 values.

```
class calculator:
    def __init__(self, m1,m2):
        self.m1 = m1
        self.m2 = m2

    def __add__(self,other):
        total_m1 = self.m1 + other.m1
        total_m2 = self.m2 + other.m2
        s3 = calculator(total_m1,total_m2)
        return s3
    def __sub__(self,other):
        total_m1 = self.m1 - other.m1
        total_m2 = self.m2 - other.m2
        s4 = calculator(total_m1,total_m2)
        return s4



    def __str__(self):
        return str(self.m1) + " : " +  str(self.m2)

s1 = calculator(45,34)
s2 = calculator(99,77)

s3 = s1 + s2
s4=s1-s2    # calculator.__add__(s1,s2)
                # s1.__add__(s2)


#print(a)
print(s3,s4,sep="\n")
```

6. Write a python script to create a Calculator 2.0 class with 2 methods for multiplication and division of 2 values and inherit it from the Calculator class.

```
class calculator_2:
    def give_data(self, n1,n2):
        if n1>n2:
            pass
        else:
            n1,n2=n2,n1
        self.n1 = n1
        self.n2 = n2

    def mul(self):
        return self.n1*self.n2

    def complete_division(self):
        return self.n1/self.n2

    def floor_div(self):
        return self.n1//self.n2

class calculator(calculator_2):
    def __init__(self, m1,m2):
        self.m1 = m1
        self.m2 = m2

    def __add__(self,other):
        total_m1 = self.m1 + other.m1
        total_m2 = self.m2 + other.m2
        s3 = calculator(total_m1,total_m2)
        return s3
    def __sub__(self,other):
        total_m1 = self.m1 - other.m1
        total_m2 = self.m2 - other.m2
        s4 = calculator(total_m1,total_m2)
        return s4
    def __str__(self):
        return str(self.m1) + " : " +  str(self.m2)

s1 = calculator(45,34)
s2 = calculator(99,77)
s3 = s1 + s2
s4= s1 - s2
print(s3,s4,sep="\n")
s1.give_data(3,4)
print(s1.mul())
print(s1.complete_division())
print(s1.floor_div())
```

7. Write a python script to create a Phone class with 2 methods to print the features (calling and sms).

```
class phone:
    def features1(self):
        print("calling")
    def features2(self):
        print("sms")
p=phone()
p.features1()
p.features2()
```

8. Write a python script to create a SmartPhone class by inheriting Calculator 2.0 and Phone Class.

```
class phone:
    def features1(self):
        print("calling")
    def features2(self):
        print("sms")
class calculator_2:
    def give_data(self, n1,n2):
        if n1>n2:
            pass
        else:
            n1,n2=n2,n1
        self.n1 = n1
        self.n2 = n2

    def mul(self):
        return self.n1*self.n2

    def complete_division(self):
        return self.n1/self.n2

    def floor_div(self):
        return self.n1//self.n2

class smartphone(phone,calculator_2):
    def __init__(self):
        print("smart class")

s=smartphone()
```

```
s.features1()
s.features2()
s.give_data(12,10)
print(s.mul())
print(s.complete_division())
print(s.floor_div())
```

9. Write a python script to create an application like Truecaller where names and numbers are stored. Truecaller class will have 2 methods (1st to fetch the name of a number and 2nd to add a new entry).

```
class trucaller:
    def name_num(self):
        self.dis={"rashmi":5645678466,"him":564783883,"ummed":6767676767,"san":6777787878,"abhay":4535243
51}
    def access_name(self):
        for x in self.dis:
            print(x)
    def edit_name(self):
        reply=input("do you want to adda new entry[ya/na]\n")
        if reply=='ya':
            n=int(input("how many  "))
            for x in range(n):
                key=input("enter the key  ")
                value=eval(input("enter the vlaue  "))
                self.dis[key]=value
            print(self.dis)
        else:
            print("ohk")

tru=trucaller()
tru.name_num()
tru.access_name()
tru.edit_name()
```

10. Write a python script to add the new method in SmartPhone class which accepts Truecaller object as a parameter and call the fetch method of Truecaller.

```
class trucaller:
    def name_num(self):
        self.dis={"rashmi":5645678466,"him":564783883,"ummed":6767676767,"san":6777787878,"abhay":4535243
51}
    def access_name(self):
        for x in self.dis:
            print(x)
    def edit_name(self):
        reply=input("do you want to adda new entry[yes/no]\n")
        if reply=='yes':
            n=int(input("how many  "))
            for x in range(n):
                key=input("enter the key  ")
                value=eval(input("enter the vlaue  "))
                self.dis[key]=value
            print(self.dis)
        else:
            print("ohk")
class smartphone(trucaller):
    def access_trucaller(tru):
            tru.name_num()
            tru.access_name()
            tru.edit_name()


tru=trucaller()
obj=smartphone()
obj.access_trucaller()
```