
1. Write a Class Painting with method calculatePaintingCost.
Write a Class FlatPainting with noOfRooms which is inheritated from Class Painting.
Override calculatePaintingCost.(Assume per room painting cost is 10000)
Write a Class BulidingPainting with noOfFlats which is inheritated from Class Painting.
Override calculatePaintingCost.(Assume per room painting cost is 25000)
Write main() method to accpet painting assignment with option 1.Flat 2.Building and then calculate the cost according to the type.

2.Class Account with name ,accNumber,balance
Class Bank with bankName, branch,accountList(5 accounts)
Implement following functions::
A.accept_account_info
B.print_account_info
C.accept_bank_info
D.print_bank_info
E.delete_account #Return 1 if deleted else 0
F.float deposit #Return updated balance
G.float withdraw #Return updated balance

Write a menu driven program using above details and perform following operations:
i. Create New Bank and 5 records of account in accountList.
ii. Depost Amount from Given Account No.
iii. Withdraw Amount from Given Account No.
iv. Delete Account

3.Write a Matrix class. Write Accept () and Print () functions. Also provide Addition (), Subtraction () and Multiplication () function. (Operator Overloading)

4. Create a class called Rational for performing comparison with fractions.
Use integer variables to represent the private data of the class numerator and denominator.
Provide a __init__ that enables an object of this class to be initialized when it is declared. The __init__ should contain default values in case no initializers are provided and should store the fraction in reduced form.
For example, the fraction would be stored in the object as 1 in the numerator and 2 in the denominator.

(Operator Overloading)
Provide Methods that perform each of the following tasks:
1. Adding two Rational numbers. The result should be stored in reduced form.
2. Subtracting two Rational numbers. The result should be stored in reduced form.
3. Multiplying two Rational numbers. The result should be stored in reduced form.
4. Dividing two Rational numbers. The result should be stored in reduced form.
5. Less than and greater than for two Rational numbers # Return 1 if true else 0
6. Equal to and not equal to for two Rational numbers # Return 1 if true else 0
5. Printing Rational numbers in the form a/b, where a is the numerator and b is the denominator.

Write a Menu Driven program to test your class.
Hint: In mathematics a rational number is any number that can be expressed as the quotient a/b of two integers, with the denominator b not equal to zero.