1. Create a class called "BankAccount" that has the following attributes:

* Account number (integer)
* Account holder's name (string)
* Account balance (float)

The class should have the following methods:

* Deposit: This method should allow the user to deposit an amount into their account.
* Withdraw: This method should allow the user to withdraw an amount from their account. The user should not be able to withdraw more than their current account balance.
* Display balance: This method should display the user's current account balance.

import unittest

class BankAccount:

def \_\_init\_\_(self,Account\_number,Account\_holders\_name,Account\_balance):

self.Account\_number=Account\_number

self.Account\_holders\_name=Account\_holders\_name

self.Account\_balance=Account\_balance

def deposit(self,deposited\_money):

if deposited\_money>=0:

self.Account\_balance+=deposited\_money

else:

print("Unsupported Request")

def withdraw(self,withdraw\_money):

if withdraw\_money > self.Account\_balance:

print("Insufficient Funds")

else:

self.Account\_balance-=withdraw\_money

def display\_balance(self):

print("The Account balance is "+ str(self.Account\_balance))

return ("The Account balance is "+ str(self.Account\_balance))

class Test\_BankAccount(unittest.TestCase):

def setUp(self):

self.my\_account = BankAccount(10012, "Pinky", 1000.0)

def test\_deposit(self):

self.my\_account.deposit(500.0)

self.assertEqual(self.my\_account.Account\_balance, 1500.0)

def test\_deposit\_negative(self):

amount=-500

self.my\_account.deposit(amount)

self.assertEqual(self.my\_account.Account\_balance, 1000.0)

def test\_withdraw\_sufficient\_balance(self):

self.my\_account.withdraw(200.0)

self.assertEqual(self.my\_account.Account\_balance, 800.0)

def test\_withdraw\_insufficient\_balance(self):

self.my\_account.withdraw(2000.0)

self.assertEqual(self.my\_account.Account\_balance, 1000.0)

def test\_display\_balance(self):

self.assertEqual(self.my\_account.display\_balance(),"The Account balance is 1000.0")

if \_\_name\_\_ == '\_\_main\_\_':

unittest.main(argv=['first-arg-is-ignored'], exit=False)