**Assignment: 7**

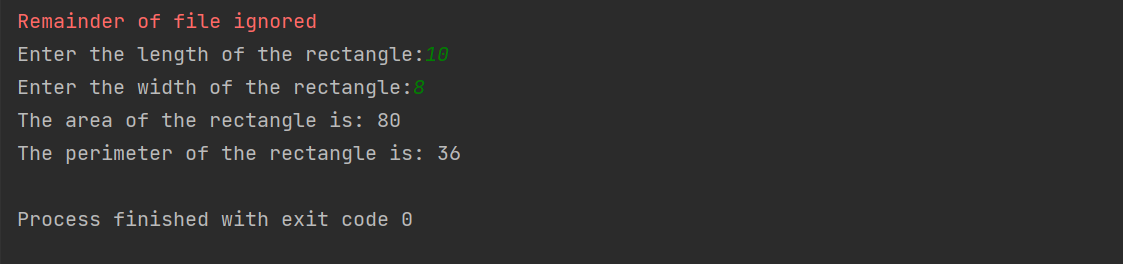
**Experiments**

1. **Write a class named Rectangle. Take the appropriate attributes and methods. Instantiate this class to use its attributes.**

**Coding:**

class Rectangle:  
 def \_\_init\_\_(self, length, width):  
 self.length = length  
 self.width = width  
  
 def area(self):  
 return self.length \* self.width  
  
 def perimeter(self):  
 return 2 \* (self.length + self.width)  
  
# Instantiate the Rectangle class  
length = int(input("Enter the length of the rectangle:"))  
width = int(input("Enter the width of the rectangle:"))  
  
rectangle = Rectangle(length, width)  
  
# Calculate the area and perimeter  
print("The area of the rectangle is:", rectangle.area())  
print("The perimeter of the rectangle is:", rectangle.perimeter())

**Output:**

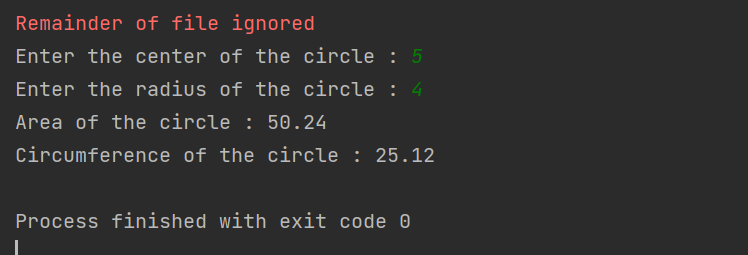


1. **Create a class Circle with attributes centre and radius. Add methods for area and circumference calculation. Now create circle objects to use these methods. Give the provision to take circle attributes from the user.**

**Coding:**

class Circle:  
 def \_\_init\_\_(self, center, radius):  
 self.center = center  
 self.radius = radius  
  
 def area(self):  
 return 3.14 \* self.radius \* self.radius  
  
 def circumference(self):  
 return 2 \* 3.14 \* self.radius  
  
  
c1 = Circle(input("Enter the center of the circle : "), int(input("Enter the radius of the circle : ")))  
print("Area of the circle :", c1.area())  
print("Circumference of the circle :", c1.circumference())

**Output:**



1. **Create a class named Account. An Account object can have following attributes:**

**Account Number**

**Account Holder**

**Account Type**

**Balance**

**credit\_account()**

**debit\_account()**

**get\_interest() (SI)**

**get\_account()**

**set\_account()**

**Add some class attributes to this Account class. Provide the appropriate constructor.**

**Create some sample bank account, use the listed methods and finally delete them.**

**Dynamically create account objects by navigating through a menu, such as:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Create account

Withdraw amount

Deposit amount

Show Interest

Delete account

\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Enter your choice?

**Coding:**

class Account:  
 # class attributes  
 bank\_name = "ABC Bank"  
 min\_deposit = 5000  
  
 # constructor  
 def \_\_init\_\_(self, acc\_num, acc\_holder, acc\_type, balance):  
 self.acc\_num = acc\_num  
 self.acc\_holder = acc\_holder  
 self.acc\_type = acc\_type  
 self.balance = balance  
  
 # Method to credit amount  
  
 def credit\_account(self, amount):  
 self.balance = self.balance + amount  
  
 # Method to debit amount  
  
 def debit\_account(self, amount):  
 self.balance = self.balance - amount  
  
 # Method to get interest  
  
 def get\_interest(self):  
 # SI = PTR  
 si = (self.balance \* 0.04 \* 1) / 12  
 return si  
  
 # Method to get account detail  
  
 def get\_account(self):  
 print("Account Number:", self.acc\_num)  
 print("Account Holder Name:", self.acc\_holder)  
 print("Account Type:", self.acc\_type)  
 print("Account Balance:", self.balance)  
  
 # Method to set account detail  
  
 def set\_account(self, acc\_num, acc\_holder, acc\_type, balance):  
 self.acc\_num = acc\_num  
 self.acc\_holder = acc\_holder  
 self.acc\_type = acc\_type  
 self.balance = balance  
  
 # Method to delete account  
  
 def delete\_account(self):  
 self.acc\_num = None  
 self.acc\_holder = None  
 self.acc\_type = None  
 self.balance = None  
  
 # creating a sample bank account  
  
  
acc1 = Account(1, "ABC", "Savings", 10000)  
acc2 = Account(2, "XYZ", "Current", 20000)  
  
# using the methods present in class  
acc1.credit\_account(5000)  
acc2.debit\_account(3000)  
  
# printing the account details  
print("Account 1 Details")  
acc1.get\_account()  
  
print("Account 2 Details")  
acc2.get\_account()  
  
# using set\_account method  
acc2.set\_account(3, "PQR", "Savings", 30000)  
  
print("Account 2 Details after updation")  
acc2.get\_account()  
  
# deleting account  
acc1.delete\_account()  
  
# printing the account details  
print("Account 1 Details after deletion")  
acc1.get\_account()  
  
  
# menu function  
def menu():  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 print("1. Create account")  
 print("2. Withdraw amount")  
 print("3. Deposit amount")  
 print("4. Show Interest")  
 print("5. Delete account")  
 print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*")  
 choice = int(input("Enter your choice : "))  
 return choice  
  
  
# main function  
while True:  
 choice = menu()  
  
 # create account  
 if choice == 1:  
 acc\_num = int(input("Enter account number: "))  
 acc\_holder = input("Enter account holder name: ")  
 acc\_type = input("Enter account type: ")  
 balance = int(input("Enter balance: "))  
  
 # creating object of class  
 acc = Account(acc\_num, acc\_holder, acc\_type, balance)  
 print("Account Created Successfully")  
 print('The account no. is :',acc\_num)  
 print('Account holder name :',acc\_holder)  
 print('Account type :',acc\_type)  
 print('Account balance :',balance)  
  
 # withdraw amount  
 elif choice == 2:  
 amount = int(input("Enter amount to withdraw: "))  
 acc.debit\_account(amount)  
 print("Amount withdrawn successfully")  
 print("Available Balance:", acc.balance)  
  
 # deposit amount  
 elif choice == 3:  
 amount = int(input("Enter amount to deposit: "))  
 acc.credit\_account(amount)  
 print("Amount deposited successfully")  
 print("Available Balance:", acc.balance)  
  
 # show interest  
 elif choice == 4:  
 si = acc.get\_interest()  
 print("Simple Interest:", si)  
  
 # delete account  
 elif choice == 5:  
 acc.delete\_account()  
 print("Account deleted successfully")  
  
  
 # invalid choice  
 else:  
 print("Invalid Choice")  
  
 # exit condition  
 exit\_choice = int(input("Enter 0 to exit "))  
 if exit\_choice == 0:  
 break

**Output:**

