Practical Lab 9

The tasks below are based around the lecture notes entitled *Classes and Objects.*

Task 1

Create a Circle class to include the following instance property: radius

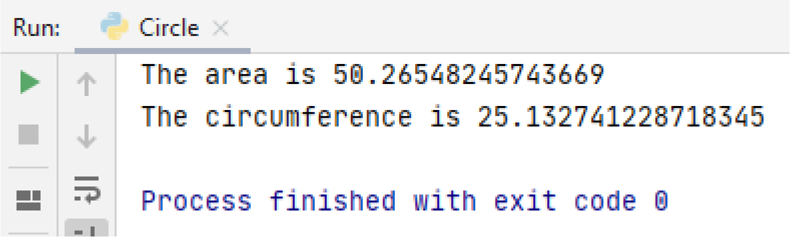
The Circle Class should have an **_ _init_ _()** method which takes in the radius.

The Circle class should have the following methods (you can **import math** and use **math.pi**)

1) **area ()** which calculates and returns the area of the circle
2) **circumference()** which calculates and returns the length of the circumference

Once you have created this class, see if you can successfully create a Circle instance and get the area and circumference for that Circle object.

If the radius of the circle is 4 then you would get the following output.



```
The area is 50.26548245743669
The circumference is 25.132741228718345

Process finished with exit code 0
```

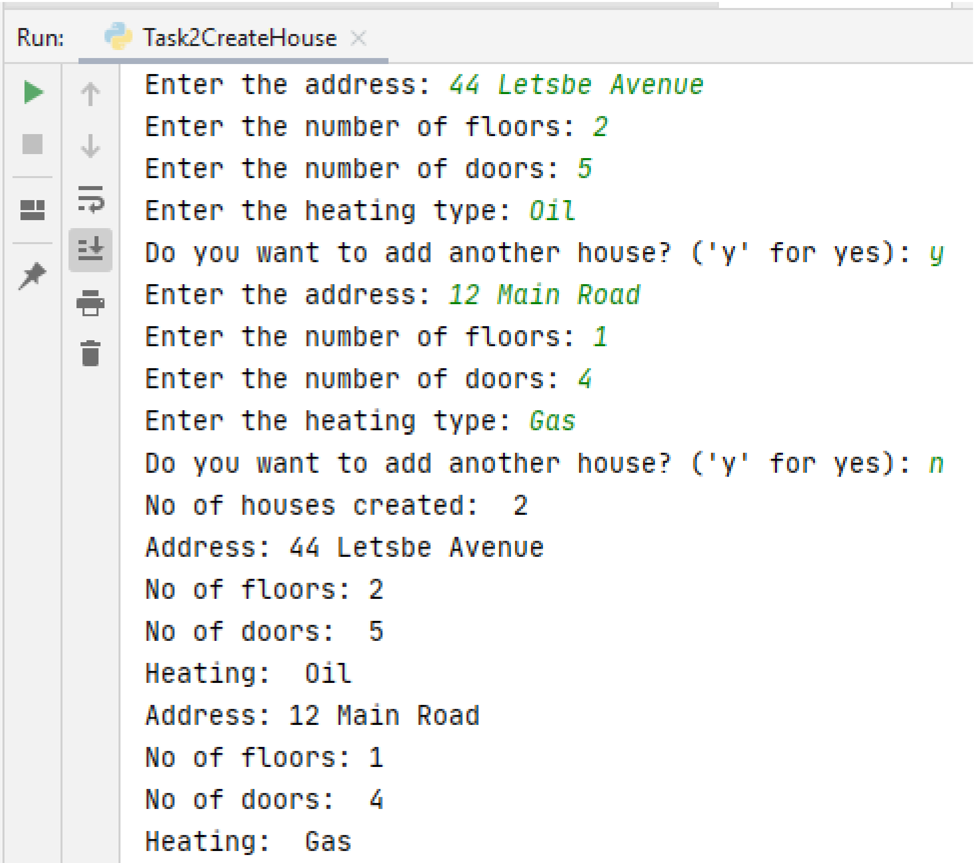
Task 2

Create a House class to include the following instance properties: address, number of floors, number of doors, type of heating

- The House Class should have an **_ _init_ _()** method which takes in these attributes
- The House Class should have a method which prints out the details of the House.

In another Python file called Task2CreateHouse, allow the user to create as many House objects as they want and print out the details of the Houses. The user should be informed as to how many House objects have been created.

Your program should look like this when run

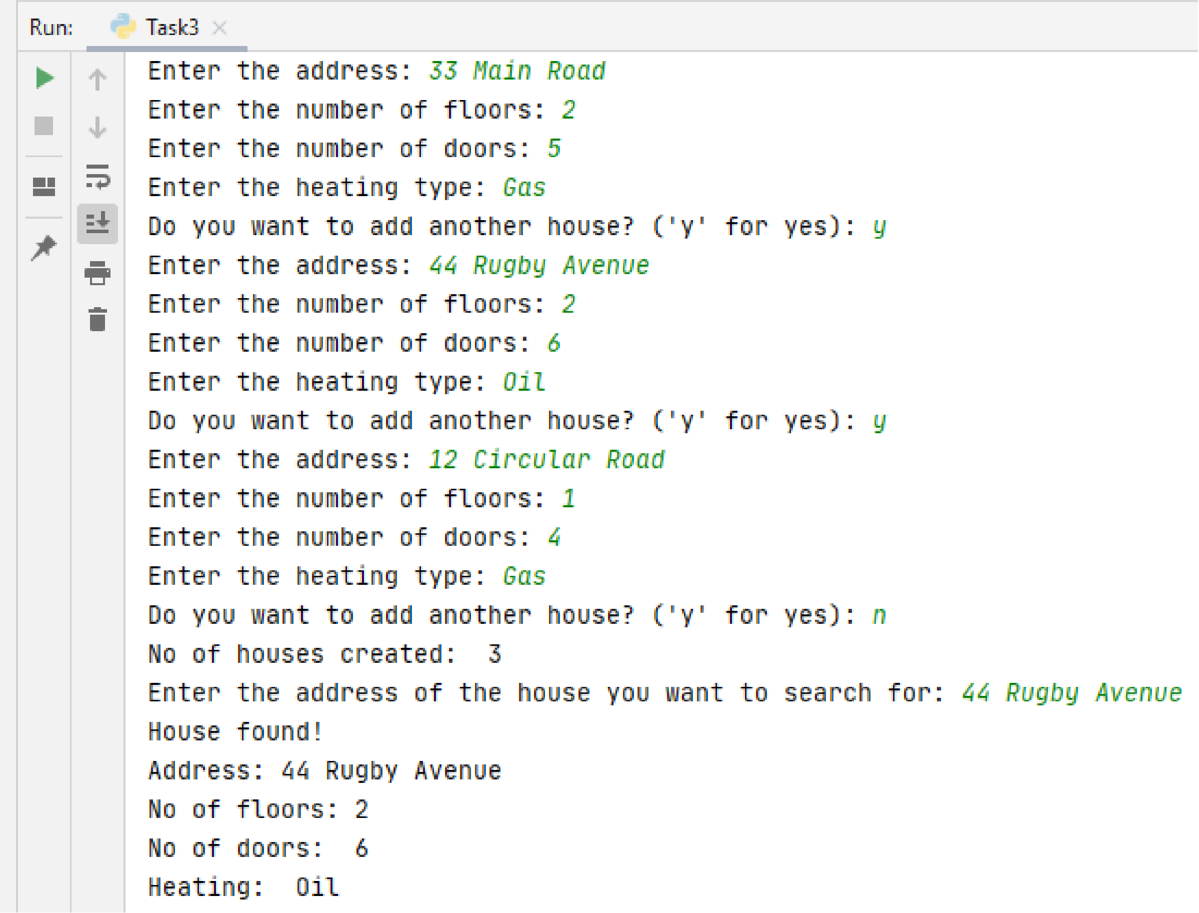


```
Enter the address: 44 Letsbe Avenue
Enter the number of floors: 2
Enter the number of doors: 5
Enter the heating type: Oil
Do you want to add another house? ('y' for yes): y
Enter the address: 12 Main Road
Enter the number of floors: 1
Enter the number of doors: 4
Enter the heating type: Gas
Do you want to add another house? ('y' for yes): n
No of houses created:  2
Address: 44 Letsbe Avenue
No of floors: 2
No of doors:  5
Heating:  Oil
Address: 12 Main Road
No of floors: 1
No of doors:  4
Heating:  Gas
```

Task 3

Amend your Task2CreateHouse Python file to allow the user to search for a particular address and have the house details printed out. There should be no additional code written for the House class.

The code should look like this when run.



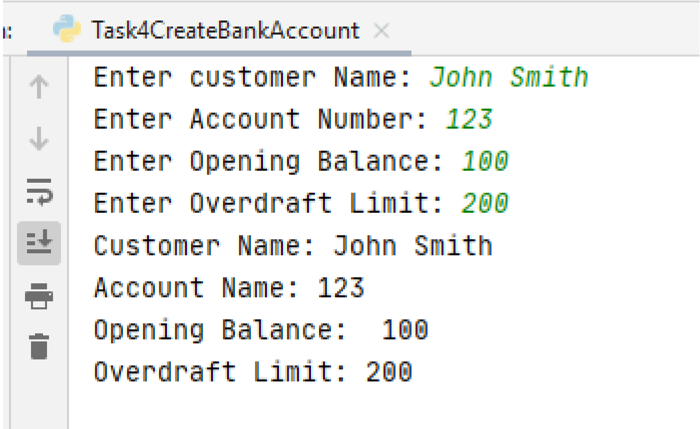
```
Enter the address: 33 Main Road
Enter the number of floors: 2
Enter the number of doors: 5
Enter the heating type: Gas
Do you want to add another house? ('y' for yes): y
Enter the address: 44 Rugby Avenue
Enter the number of floors: 2
Enter the number of doors: 6
Enter the heating type: Oil
Do you want to add another house? ('y' for yes): y
Enter the address: 12 Circular Road
Enter the number of floors: 1
Enter the number of doors: 4
Enter the heating type: Gas
Do you want to add another house? ('y' for yes): n
No of houses created:  3
Enter the address of the house you want to search for: 44 Rugby Avenue
House found!
Address: 44 Rugby Avenue
No of floors: 2
No of doors:  6
Heating:  Oil
```

Task 4

(a) Create a BankAccount class which will have the following instance properties: Customer name, Bank Account Number, Opening Balance, Overdraft Limit

The BankAccount Class should have an **_ _init_ _()** method which takes these properties and a method which prints these details out

Create a BankAccount object in another Python file called Task4CreateBankAccount and print out the details. Your code should look like this when run.

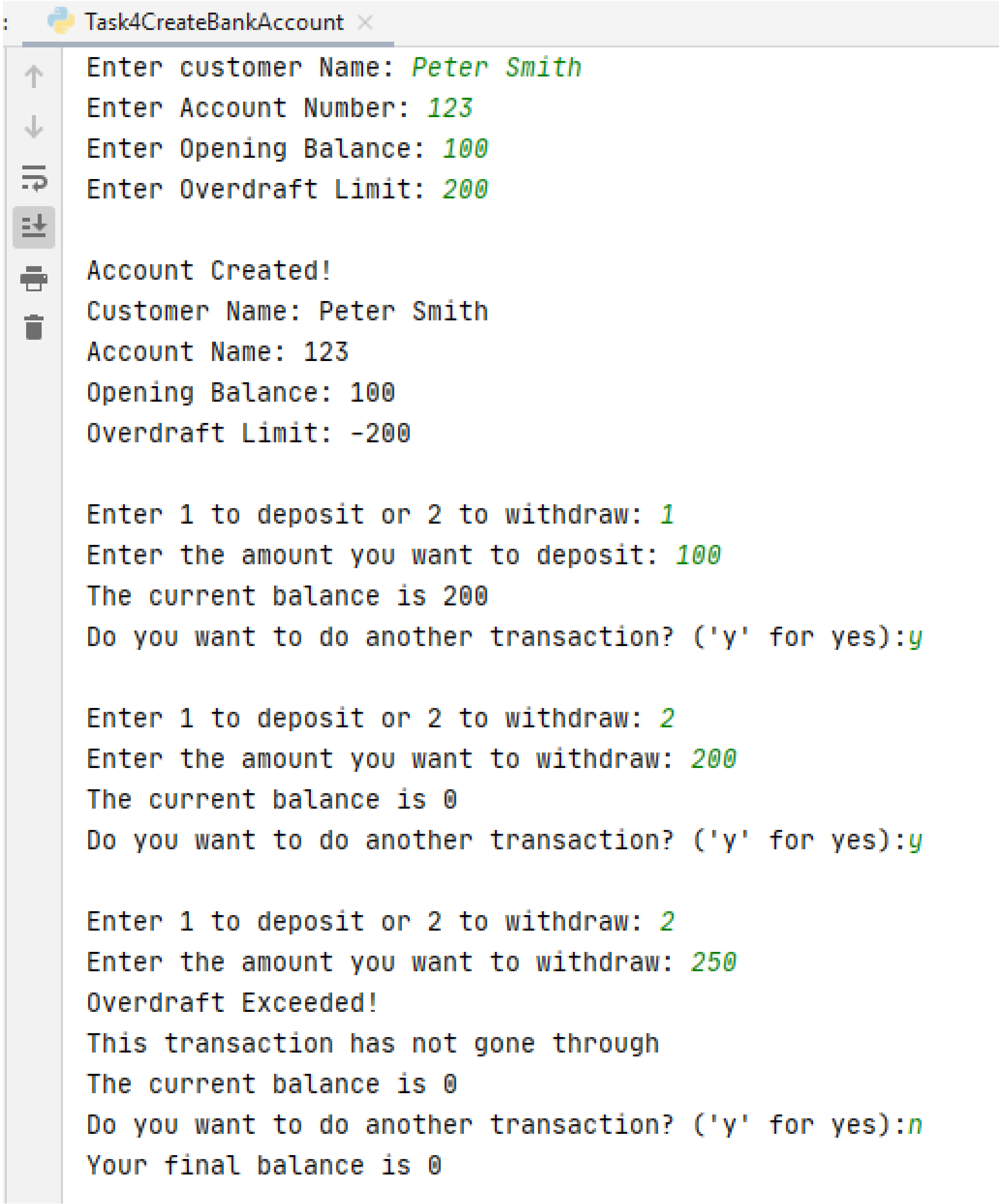


```
Enter customer Name: John Smith
Enter Account Number: 123
Enter Opening Balance: 100
Enter Overdraft Limit: 200
Customer Name: John Smith
Account Name: 123
Opening Balance:  100
Overdraft Limit: 200
```

(b) Using a menu system, allow the user to deposit and withdraw money. If the user hits the overdraft limit, then a warning message should be displayed, and the transaction should not take place. To do all this, have the following in your BankAccount class:

- A new instance property that will represent current balance; this should be set initially to the opening balance in the bank account
- A deposit function – this will adjust the current balance by adding to it and printing out the new current balance
- A withdraw function – this will adjust the current balance by subtracting from it and print out the new current balance (note this function must call the check overdraft limit function first for the transaction to be valid)
- A check overdraft limit function which checks to see if the user is within their overdraft limit. Note that this is a value method which will either return True or False.

See the next page as to how this program could look when run



```
Enter customer Name: Peter Smith
Enter Account Number: 123
Enter Opening Balance: 100
Enter Overdraft Limit: 200

Account Created!
Customer Name: Peter Smith
Account Name: 123
Opening Balance: 100
Overdraft Limit: -200

Enter 1 to deposit or 2 to withdraw: 1
Enter the amount you want to deposit: 100
The current balance is 200
Do you want to do another transaction? ('y' for yes):y

Enter 1 to deposit or 2 to withdraw: 2
Enter the amount you want to withdraw: 200
The current balance is 0
Do you want to do another transaction? ('y' for yes):y

Enter 1 to deposit or 2 to withdraw: 2
Enter the amount you want to withdraw: 250
Overdraft Exceeded!
This transaction has not gone through
The current balance is 0
Do you want to do another transaction? ('y' for yes):n
Your final balance is 0
```