

## Nicholas Graham, CPRG 216, Lab 7 – Classes

ID: 956736

```
lab7_classes.py •
lab7_classes.py > Student > set_id
1 class Student:
2     def __init__(self, name="", id=0, address=""):
3         self.__name = name
4         self.__id = int(id)
5         self.__address = address
6
7     def get_name(self):
8         return self.__name
9     def get_id(self):
10        return self.__id
11    def get_address(self):
12        return self.__address
13
14    def set_name(self, name):
15        self.__name = name
16    def set_id(self, id):
17        self.__id = id
18    def set_address(self, address):
19        self.__address = address
20
21    def __str__(self):
22        return f"Name: {self.__name}\nID: {self.__id}\nAddress: {self.__address}\n"
23
24    my_info = Student("Nick Graham", 956736, "123 Street St.")
25    print(my_info)
26
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Drive - Southern Alberta Institute of Technology/OOP 1/Lab7-Classes/lab7\_classes.py"

Name: Nick Graham  
ID: 956736  
Address: 123 Street St.

PS C:\Users\nsgra\OneDrive - Southern Alberta Institute of Technology\OOP 1\Lab7-Classes> |

```
lab7_classes.py X
lab7_classes.py > Circle > display_circle_perimeter

28 class Circle:
29     def __init__(self, radius=0):
30         self.__radius = radius
31
32     def get_radius(self):
33         return self.__radius
34
35     def get_circle_area(self, radius):
36         self.__circle_area = 3.14 * (radius**2)
37         return self.__circle_area
38
39     def get_circle_perimeter(self, radius):
40         self.__circle_perimeter = radius * 2 * 3.14
41         return self.__circle_perimeter
42
43     def set_circle_radius(self, radius):
44         self.__radius = radius
45
46
47     def display_circle_perimeter(self, radius):
48         self.__radius = radius
49         print(f"for circle with radius = {self.__radius} units:\nPerimeter = {self.get_circle_perimeter(self.__radius):.2f} units\n")
50
51     def display_circle_area(self, radius):
52         self.__radius = radius
53         print(f"for circle with radius = {self.__radius} units:\nArea = {self.get_circle_area(self.__radius):.2f} units^2\n")
54
55     def __str__(self):
56         return f"for circle with radius = {self.__radius} units:\nArea = {self.get_circle_area(self.__radius)} units^2\nPerimeter = {self.get_circle_perimeter(self.__radius)} units"
57
58 my_circle = Circle()
59 my_circle.display_circle_perimeter(5)
60 my_circle.display_circle_area(5)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Python + - [ ] ... ^ x

For circle with radius = 5 units:
Perimeter = 31.40 units

For circle with radius = 5 units:
Area = 78.50 units^2

Ln 48, Col 31 Spaces: 4 UTF-8 CRLF Python 3.12.4 64-bit (Microsoft Store) Go Live
12:21 PM 2024-07-26
```

```

lab7_classes.py
lab7_classes.py > BankAccount

61
62 class BankAccount:
63     def __init__(self, account_number=0, customer_name="", balance=0):
64         self.__account_number = account_number
65         self.__customer_name = customer_name
66         self.__balance = balance
67
68     def get_account_number(self):
69         return self.__account_number
70     def get_customer_name(self):
71         return self.__customer_name
72     def get_balance(self):
73         return self.__balance
74
75     def set_account_number(self, account_number):
76         self.__account_number = account_number
77     def set_customer_name(self, customer_name):
78         self.__customer_name = customer_name
79     def set_balance(self, balance):
80         self.__balance = balance
81
82     def deposit(self):
83         deposit_amount = float(input("How much money would you like to deposit? $"))
84         self.__balance += deposit_amount
85         print("Deposit Successful.\n")
86         self.display()
87
88     def withdraw(self):
89         withdraw_amount = float(input("How much money would you like to withdraw? $"))
90         self.__balance -= withdraw_amount
91         print(f"Withdrawl successful.\n")
92         self.display()
93
94     def display(self):
95         print(f"Account information:\nCustomer Name: {" " * 2}{self.__customer_name}\nAccount

```

```

94     def display(self):
95         print(f"Account information:\nCustomer Name: {" " * 2}{self.__customer_name}\nAccount Number: {" " * 2}{self.__account_number}\nAccount
          balance: ${self.__balance}")
96
97     customer = BankAccount()
98
99     customer.set_account_number(123)
100    customer.set_customer_name("Nick Graham")
101    customer.set_balance(100)
102
103    customer.deposit()
104
105    customer.withdraw()
106

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Python + -

```

How much money would you like to deposit? $50
Deposit Successful.

Account information:
Customer Name:  Nick Graham
Account Number:  #123
Account balance: $150.0
How much money would you like to withdraw? $25
Withdrawl successful.

Account information:
Customer Name:  Nick Graham
Account Number:  #123
Account balance: $125.0
PS C:\Users\nsgra\OneDrive - Southern Alberta Institute of Technology\OOP 1\Lab7-Classes>

```