

Exercise 9: Returning Methods

Use returning methods for this exercise.

1. Write a method called **calculateDistance(x1, y1, x2, y2)** that calculates distance between two points and returns that value. Check that your program's math is correct by completing a question by hand. Save your program as calculateDistance.py.

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

2. Write a compare method that:
returns 1 if a > b,
returns 0 if a == b, and
returns -1 if a < b.
Save your program as compare.py.
3. Write a method that calculates and **returns** the hypotenuse of a triangle given its two other sides. Save your program as hypotenuse.py.
4. Write a method called isEven(number) that **returns** True if the number is even and **returns** False if the number isn't. Save your program as evenNumber.py.
5. Write a method called isDigit(s) that accepts a digit s. The method should determine if s is a digit between 0 and 9. If yes, **returns** True, else, **returns** False. Save your program as digit.py.

Example/

Please enter a number: 8

Your number 8 is a digit

6. Write a program that prints your balance. I have listed five 5 methods called in the main method that you have to complete.

getDepositAmount() - should handle getting input from the user. Make sure your value is an int or float.

deposit(depositAmount, accountBalance) - should add the deposit to the account balance and return the new account balance

getWithdrawalAmount(accountBalance) - should handle getting input from the user and evaluate if the withdraw amount is less than the account balance. The method should return the withdrawl value retrieved or 0 if the withdrawl value is greater than the accountBalance.

withdraw(withdrawlAmount, accountBalance) - should subtract the withdrawl amount from the account balance and return the new account balance

print(accountBalance) - should print out the current account balance in a nicely formatted way. Maybe ...

Your account balance is: \$1090.89

Save your program as balance.py.

7. Write a method called isLeapYear(someYear) that accepts an integer value (Ex/1904) that represents a year. The method **returns** the boolean value True if the year is a leap year and False if the year is not.

A year that is evenly divided by 4 or 400 is a leap year. 1988, 1992, 1996 are leap years.

However, a year that is also evenly divided by 100 is not unless it is also evenly divided by 400. So 1800, 1900 are not leap years but 2000 is.

Save your program as leapYear.py.