Project 15-1: Rectangle or Square Calculator

Create an object-oriented program that uses inheritance to perform calculations on a rectangle or a square.

Console

```
Rectangle Calculator

Rectangle or square? (r/s): r
Height:    5
Width:     10
Perimeter: 30
Area:      50
* * * * * * * * * *
*                 *
*                 *
*                 *
* * * * * * * * * *

Continue? (y/n): y

Rectangle or square? (r/s): s
Length:    5
Perimeter: 20
Area:      25
* * * * *
*       *
*       *
*       *
* * * * *

Continue? (y/n): n

Bye!
```

Specifications

- Use a Rectangle class that provides attributes to store the height and width of a rectangle. This class should also provide methods that calculate the perimeter and area of the rectangle. In addition, it should provide a __str__() method that returns a string representation of the rectangle.
- Use a Square class that inherits the Rectangle class. This class should include a constructor that initializes the height and width attributes of the Rectangle superclass to the length that's passed to the constructor of the Square subclass.
- The program should determine whether the user wants to enter a rectangle or a square.
- For a rectangle, the program should get the height and width from the user.
- For a square, the program should get the length of the square from the user.
- Assume that the user will enter valid data.

Project 15-2: Roshambo

Create an object-oriented program for a Roshambo game where the user can choose to compete against one of two computer players: Bart or Lisa.

Console

```
Roshambo Game

Enter your name: Joel

Would you like to play Bart or Lisa? (b/l): b

Rock, paper, or scissors? (r/p/s): r

Joel: rock
Bart: rock
Draw!

Play again? (y/n): y

Rock, paper, or scissors? (r/p/s): p

Joel: paper
Bart: rock
Joel wins!

Play again? (y/n): y

Rock, paper, or scissors? (r/p/s): s

Joel: scissors
Bart: rock
Bart wins!

Play again? (y/n): n

Thanks for playing!
```

Specifications

- Create a class named Player that provides attributes for storing the player's name and Roshambo value.
- Add a generateRoshambo() method. This method should always set the Roshambo attribute to rock.
- Add a play() method that accepts another Player object as an argument. This method should return the current Player object if that player is the winner, the Player object it receives if that player is the winner, or None if there's a tie.
- In the game of Roshambo, rock beats scissors, paper beats rock, and scissors beats paper.
- Create a class named Bart that inherits the Player class and sets its name attribute to "Bart".
- Create a class named Lisa that inherits the Player class and sets its name attribute to "Lisa". This class should override the generateRoshambo() method so it randomly sets the Roshambo attribute to rock, paper, or scissors.

Enhancement

- Keep track of wins and losses and display them at the end of each session.

Project 15-3: Customer or Employee Creator

Create an object-oriented program that allows you to enter data for customers and employees.

Console

```
Customer/Employee Data Entry

Customer or employee? (c/e): c

DATA ENTRY
First name: Frank
Last name: Wilson
Email: frank44@gmail.com
Number: M10293

CUSTOMER
Name:       Frank Wilson
Email:      frank44@gmail.com
Number:     M10293

Continue? (y/n): y

Customer or employee? (c/e): e

DATA ENTRY
First name: joel
Last name: murach
Email: joel@murach.com
SSN: 123-45-6789

EMPLOYEE
Name:       Joel Murach
Email:      joel@murach.com
SSN:        123-45-6789

Continue? (y/n): n

Bye!
```

Specifications

- Create a Person class that provides attributes for first name, last name, and email address. This class should provide a property or method that returns the person's full name.
- Create a Customer class that inherits the Person class. This class should add an attribute for a customer number.
- Create an Employee class that inherits the Person class. This class should add an attribute for a social security number (SSN).
- The program should create a Customer or Employee object from the data entered by the user, and it should use this object to display the data to the user. To do that, the program can use the isinstance() function to check whether an object is a Customer or Employee object.

Project 15-4: Random Integer List

Create an object-oriented program that uses a custom list object to automatically generate and work with a series of random integers.

Console

```
Random Integer List

How many random integers should the list contain?: 12

Random Integers
===============
Integers:  17, 34, 34, 15, 71, 44, 97, 48, 19, 12, 83, 42
Count:     12
Total:     516
Average:   43.0

Continue? (y/n): y

Random Integers
===============
Integers:  52, 88, 10, 77, 56, 91, 17, 51, 22, 14, 48, 37
Count:     12
Total:     563
Average:   46.917

Continue? (y/n): n

Bye!
```

Specifications

- Create a RandomIntList class that inherits the list class. This class should allow a programmer to create a list of random integers from 1 to 100 by writing a single line of code. For example, a programmer should be able to create a custom list that stores 12 random integers with this line of code:

  ```
  int_list = RandomIntList(12)
  ```

- To do that, you can use the self keyword to access the list superclass like this:

  ```
  self.append(rand_int)
  ```

- The RandomIntList class should contain methods or properties for getting the count, average, and total of the numbers in the list. In addition, it should contain a __str__ method for displaying a comma-separated list of integers as shown above.
- The program should use the RandomIntList class to generate the list of random integers, display the list, and get the summary data (count, total, and average).
- The program should make sure the integer entered by the user is valid.