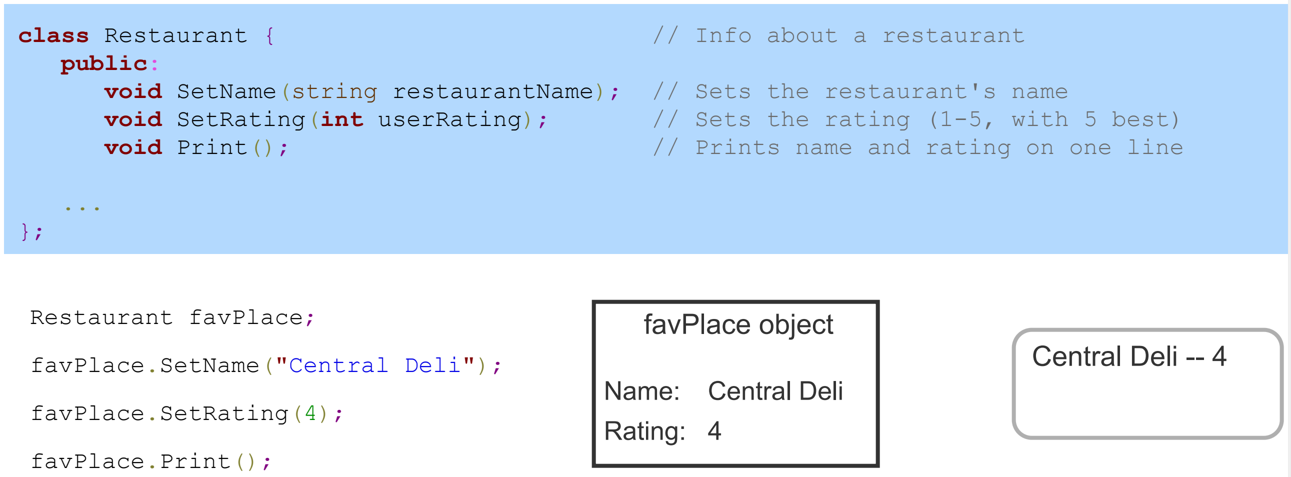
7.2 Using a class

Classes intro: Public member functions

The ***class*** construct defines a new type that can group data and functions to form an object. A class' ***public member functions*** indicate all operations a class user can perform on the object. The power of classes is that a class user need not know how the class' data and functions are implemented, but need only understand how each public member function behaves. The animation below shows a class' public member function declarations only; the remainder of the class definition is discussed later.

PARTICIPATION ACTIVITY 7.2.1: A class example: Restaurant class.

1 2 3 2x speed



```
class Restaurant {                          // Info about a restaurant
   public:
      void SetName(string restaurantName); // Sets the restaurant's name
      void SetRating(int userRating);      // Sets the rating (1-5, with 5 best)
      void Print();                        // Prints name and rating on one line

   ...
};
```

```
Restaurant favPlace;

favPlace.SetName("Central Deli");

favPlace.SetRating(4);

favPlace.Print();
```

favPlace object

Name: Central Deli

Rating: 4

```
Central Deli -- 4
```

Then, the class user can call the functions to operate on the object. A class user need not know ho
the class' data or functions are implemented.

Feedback?

PARTICIPATION ACTIVITY 7.2.2: Using a class.

Consider the example above.

1) Which operation can a class user

perform on an object of type Restaurant?

- ○ Get the name
- ○ Set the name
- ○ Get the rating

2) Calling Print() on an object of type Restaurant might yield which output?

- ○ Marias -- 5
- ○ 5
  Marias
- ○ Marias
  5

3) Although not visible in the part of the class definition shown above, how many internal data variables does the class contain?

- ○ 1
- ○ 2
- ○ Unknown

Feedback?

Using a class

A programmer can create one or more objects of the same class. Declaring a variable of a class type creates an ***object*** of that type. Ex: `Restaurant favLunchPlace;` declares a Restaurant object named favLunchPlace.

The "." operator, known as the ***member access operator***, is used to invoke a function on an object. Ex: `favLunchPlace.SetRating(4)` calls the SetRating() function on the favLunchPlace object, which sets the object's rating to 4.

PARTICIPATION ACTIVITY 7.2.3: Using the Restaurant class.

1 2 3 4 2x speed

```
int main() {
   Restaurant favLunchPlace;
```

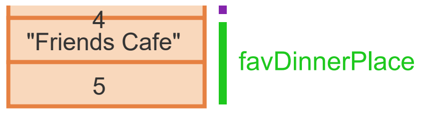
```
   Restaurant favDinnerPlace;

   favLunchPlace.SetName("Central Deli");
   favLunchPlace.SetRating(4);

   favDinnerPlace.SetName("Friends Cafe");
   favDinnerPlace.SetRating(5);

   cout << "My favorite restaurants: " << endl;
   favLunchPlace.Print();
   favDinnerPlace.Print();

   return 0;
}
```



```
My favorite restaurants:
Central Deli -- 4
Friends Cafe -- 5
```

Invoking the Print() operation on a Restaurant object, prints the restaurant's name and rating.

Feedback?

PARTICIPATION ACTIVITY 7.2.4: Using the Restaurant class.

The following questions consider *using* the Restaurant class.

1) Type a variable declaration that creates an object named favBreakfastPlace.

Check Show answer

2) Using separate variable declarations, create an object bestDessertPlace, followed an object bestIndianFood.

Check Show answer

3) Given the code below, how many objects are created?

```
Restaurant bestIndianFood;
Restaurant bestSushi;
Restaurant bestCoffeeShop;
int newRating;
```

Check Show answer

4) Object bestSushi is of type Restaurant. Type a statement that sets bestSushi's name to "Sushi Station".

Check Show answer

5) Type a statement to print bestCoffeeShop's name and rating.

Check Show answer

Feedback?

Class example: string

C++'s string type is a class. The string class stores a string's characters in memory, along with variables indicating the length and other things, but a string's user need not know such details. Instead, the string's user just needs to know what public member functions can be used, such as those shown below. (Note: size_t is an unsigned integer type).

Figure 7.2.1: Some string public member functions (many more exist).

```
char& at(size_t pos); // Returns a reference to the character at position pos in the string.

size_t length() const; // Returns the number of characters in the string

void push_back(char c); // Appends character c to the string's end (increasing length by 1).
```

Feedback?

PARTICIPATION ACTIVITY 7.2.5: Using the string class.

Consider the public member functions shown above for the string class.

1) Given string s = "Hi". How many

bytes does object s utilize in memory?

- ○ 2
- ○ 3
- ○ Unknown

2) Given string s = "Hi", how can a user append "!" to have s become "Hi!".

- ○ s.push_back('!')
- ○ s.at('!')
- ○ Unknown

3) What enables a user to utilize the string class?

- ○ Nothing; strings are built into C++
- ○ #include <string>

Feedback?