Unit 2 - Lesson 12
Burger Class FRQ

Computer Science A

Warm Up

Retrieve

your knowledge and ideas and write it down silently

Pair

up with a neighbor and talk about your reflections

Share

your thoughts in a class discussion

Prompt

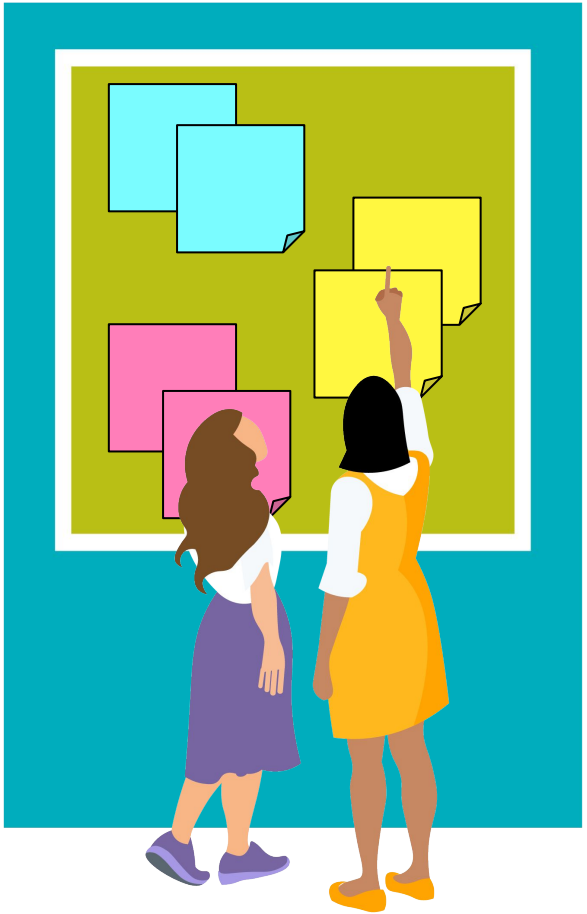
What strategies did you use while working with the **SpiralPainter FRQ** in the previous unit?

Activity

Step 1: FRQ Social

You will:

1. **Annotate** the problem
2. **Visualize** the solution
3. **Share possible solutions** with peers



Step 2: FRQ Writing

You will:

1. **Write your solution** in Java code

Step 3: Scoring

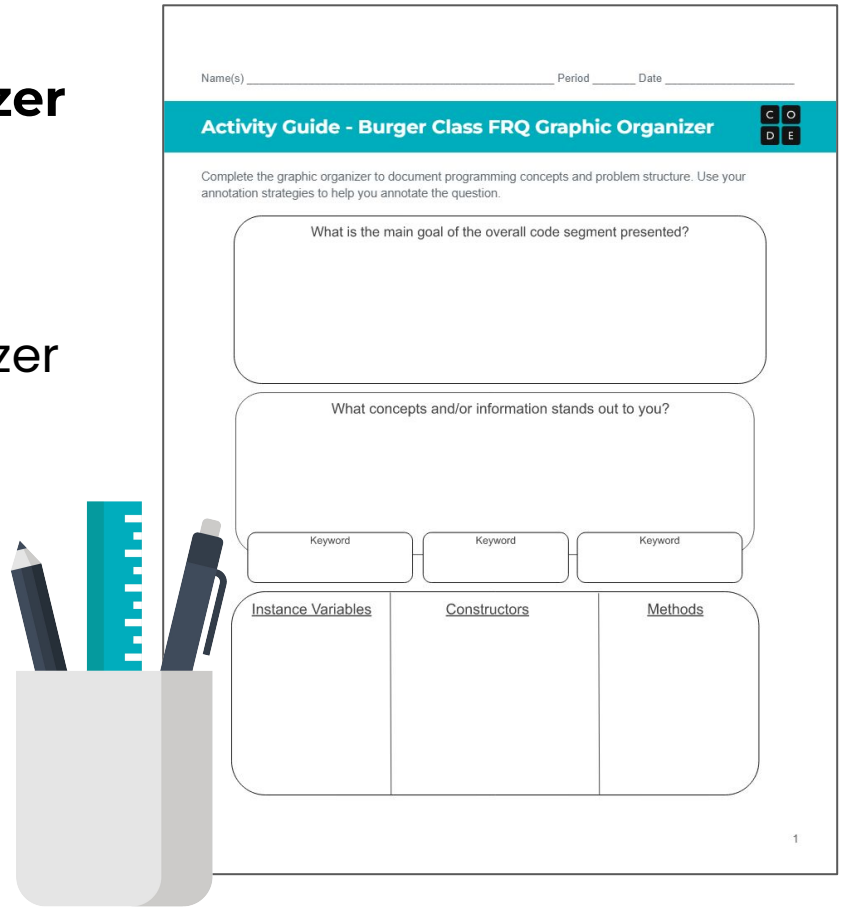
You will:

1. **Retrieve** the **scoring guidelines**
2. **Assess and reflect** on your solution and results with a partner

Burger Class FRQ Graphic Organizer

📍 You should have:

- Burger Class FRQ Graphic Organizer
- pen / pencil



Name(s) _____ Period _____ Date _____

Activity Guide - Burger Class FRQ Graphic Organizer

Complete the graphic organizer to document programming concepts and problem structure. Use your annotation strategies to help you annotate the question.

What is the main goal of the overall code segment presented?

What concepts and/or information stands out to you?

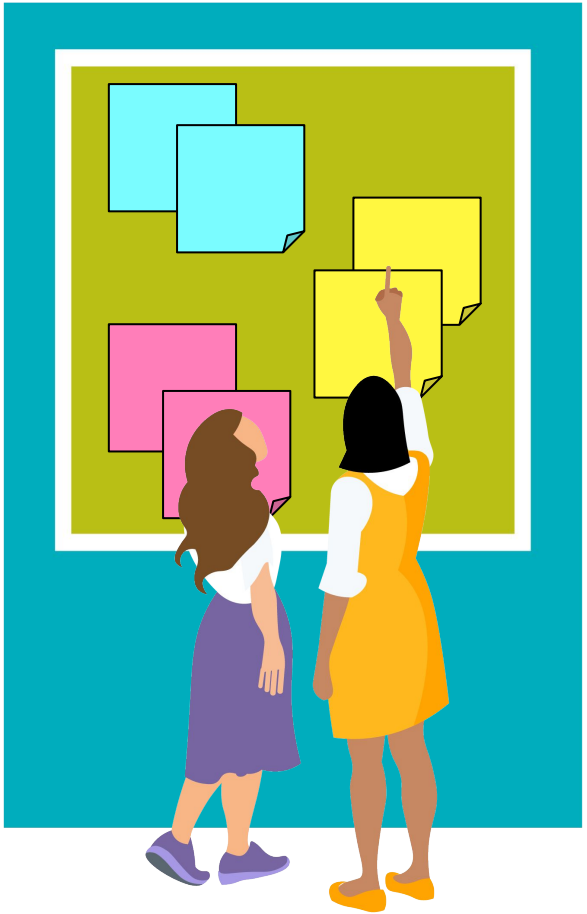
Keyword | Keyword | Keyword

Instance Variables	Constructors	Methods

1

✅ Do This:

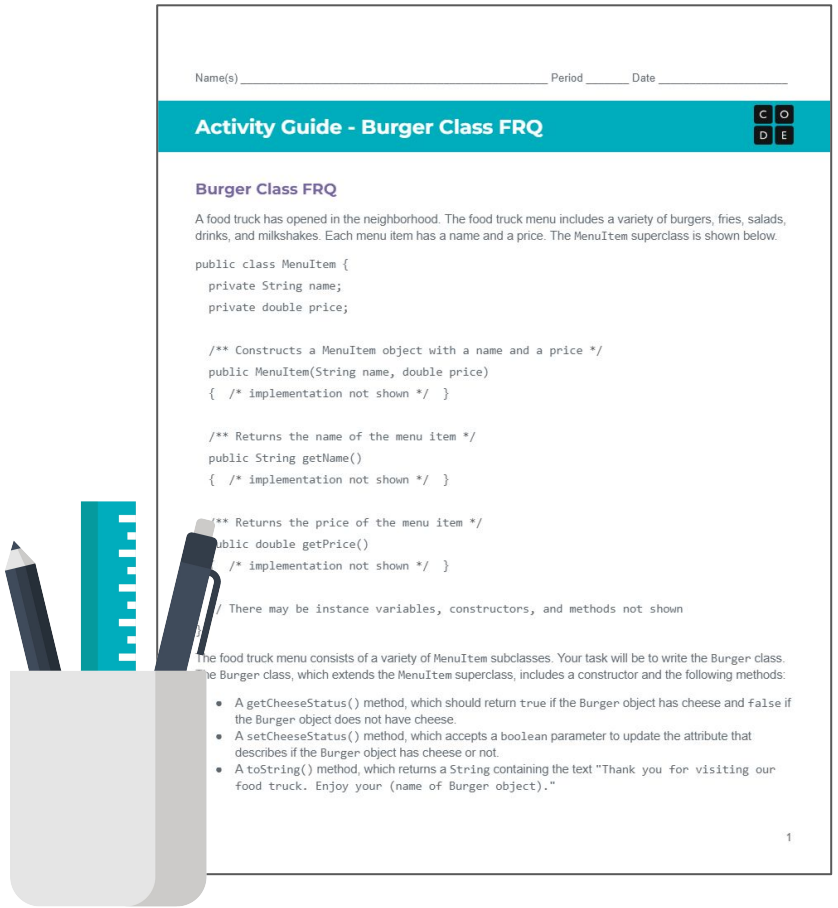
1. **Find an FRQ** posted around the room.
2. **Read the FRQ** and **share possible solutions** with your peers.
3. Complete the **Burger Class FRQ Graphic Organizer**.



Burger Class FRQ

📍 **You should have:**

- Burger Class FRQ
- pen / pencil



Name(s) _____ Period _____ Date _____

Activity Guide - Burger Class FRQ

Burger Class FRQ

A food truck has opened in the neighborhood. The food truck menu includes a variety of burgers, fries, salads, drinks, and milkshakes. Each menu item has a name and a price. The `MenuItem` superclass is shown below.

```
public class MenuItem {
  private String name;
  private double price;

  /** Constructs a MenuItem object with a name and a price */
  public MenuItem(String name, double price)
  {  /* implementation not shown */  }

  /** Returns the name of the menu item */
  public String getName()
  {  /* implementation not shown */  }

  /** Returns the price of the menu item */
  public double getPrice()
  {  /* implementation not shown */  }

  // There may be instance variables, constructors, and methods not shown
```

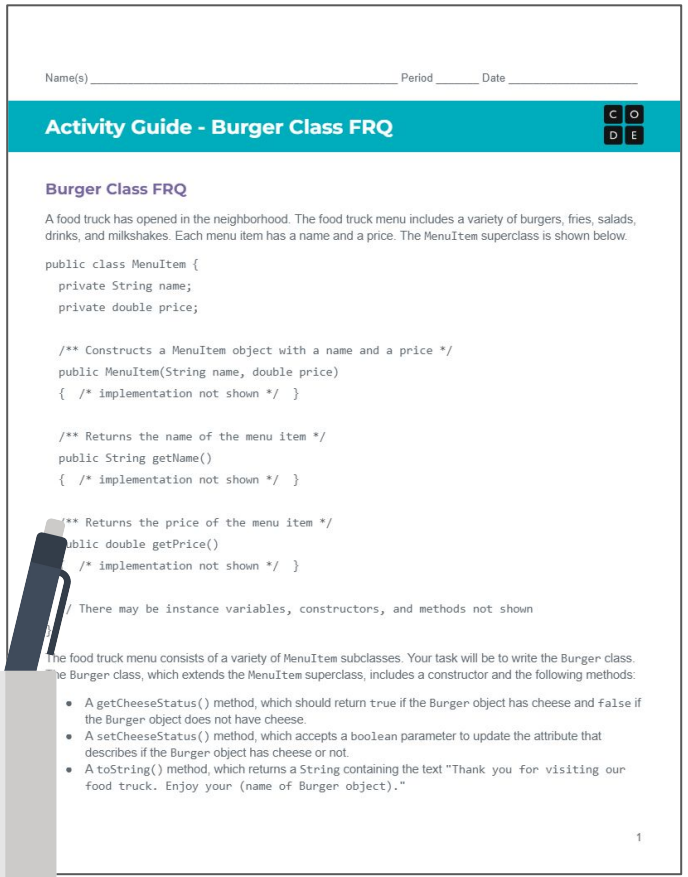
The food truck menu consists of a variety of `MenuItem` subclasses. Your task will be to write the `Burger` class. The `Burger` class, which extends the `MenuItem` superclass, includes a constructor and the following methods:

- A `getCheeseStatus()` method, which should return `true` if the `Burger` object has cheese and `false` if the `Burger` object does not have cheese.
- A `setCheeseStatus()` method, which accepts a `boolean` parameter to update the attribute that describes if the `Burger` object has cheese or not.
- A `toString()` method, which returns a `String` containing the text `"Thank you for visiting our food truck. Enjoy your (name of Burger object)."`

1

✅ Do This:

Write your solution in **Java code** for the Burger Class FRQ.



Name(s) _____ Period _____ Date _____

Activity Guide - Burger Class FRQ

Burger Class FRQ

A food truck has opened in the neighborhood. The food truck menu includes a variety of burgers, fries, salads, drinks, and milkshakes. Each menu item has a name and a price. The `MenuItem` superclass is shown below.

```
public class MenuItem {
  private String name;
  private double price;

  /** Constructs a MenuItem object with a name and a price */
  public MenuItem(String name, double price)
  {  /* implementation not shown */  }

  /** Returns the name of the menu item */
  public String getName()
  {  /* implementation not shown */  }

  /** Returns the price of the menu item */
  public double getPrice()
  {  /* implementation not shown */  }

  // There may be instance variables, constructors, and methods not shown
```

The food truck menu consists of a variety of `MenuItem` subclasses. Your task will be to write the `Burger` class. The `Burger` class, which extends the `MenuItem` superclass, includes a constructor and the following methods:

- A `getCheeseStatus()` method, which should return `true` if the `Burger` object has cheese and `false` if the `Burger` object does not have cheese.
- A `setCheeseStatus()` method, which accepts a `boolean` parameter to update the attribute that describes if the `Burger` object has cheese or not.
- A `toString()` method, which returns a `String` containing the text `"Thank you for visiting our food truck. Enjoy your (name of Burger object)."`

1

Burger Class FRQ Scoring Guidelines

📍 **You should have:**

- Burger Class FRQ Scoring Guidelines
- pen / pencil

Name(s) _____ Period _____ Date _____

Activity Guide - Burger Class FRQ Scoring Guidelines

Class: Burger	9 points

+1 Declares all appropriate `private` instance variables

+3 Constructor

- +1 Declares header: `public Burger(String ____, double ____, boolean ____)`
- +1 Calls the superclass constructor with appropriate parameters to initialize instance variables
- +1 Uses parameters and appropriate values to initialize the subclass instance variables

+2 `getCheeseStatus()` method

- +1 Declares header `public boolean getCheeseStatus()`
- +1 Returns the appropriate instance variable

+2 `setCheeseStatus(boolean ____)` method

- +1 Declares header `public void setCheeseStatus(boolean ____)`
- +1 Updates the appropriate instance variable using the parameter

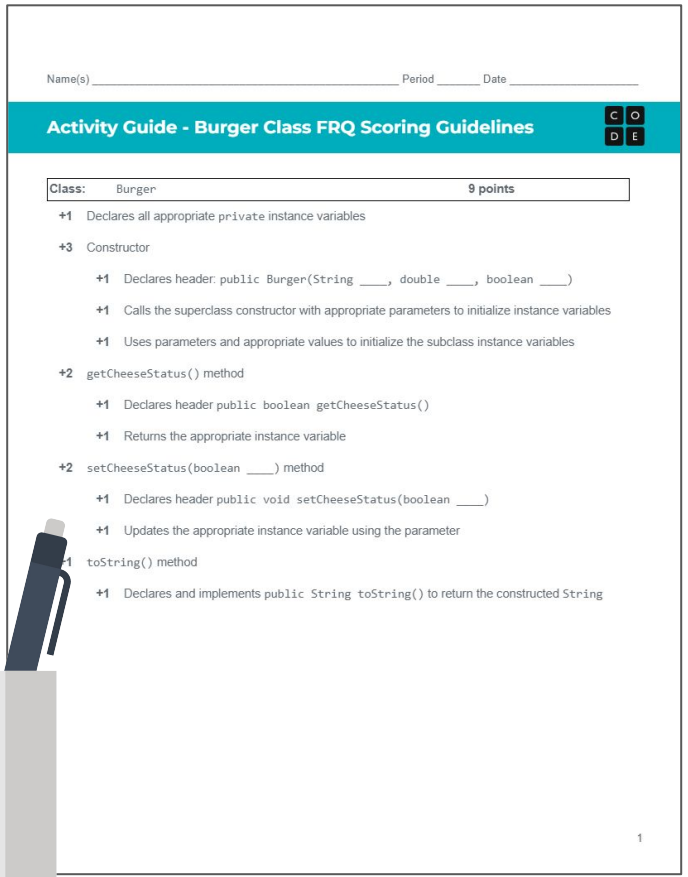
+1 `toString()` method

- +1 Declares and implements `public String toString()` to return the constructed `String`

1

✅ Do This:

1. Assess your solution using the **Burger Class FRQ Scoring Guidelines** with your partner.
2. Complete the reflection on the last page of the **Burger Class FRQ** handout.



Name(s) _____ Period _____ Date _____

Activity Guide - Burger Class FRQ Scoring Guidelines

Class: Burger	9 points

+1 Declares all appropriate `private` instance variables

+3 Constructor

- +1 Declares header: `public Burger(String ____, double ____, boolean ____)`
- +1 Calls the superclass constructor with appropriate parameters to initialize instance variables
- +1 Uses parameters and appropriate values to initialize the subclass instance variables

+2 `getCheeseStatus()` method

- +1 Declares header `public boolean getCheeseStatus()`
- +1 Returns the appropriate instance variable

+2 `setCheeseStatus(boolean ____)` method

- +1 Declares header `public void setCheeseStatus(boolean ____)`
- +1 Updates the appropriate instance variable using the parameter

+1 `toString()` method

- +1 Declares and implements `public String toString()` to return the constructed `String`

1

Class Header and Instance Variable

```
public class Burger extends MenuItem {

  private boolean hasCheese;
```

Scoring Guidelines

Declares all appropriate **`private`** instance variables

Constructor

```
public Burger(String name, double price, boolean cheese) {
  super(name, price);
  hasCheese = cheese;
}
```

Scoring Guidelines

Declares header: `public Burger(String ____, double ____, boolean ____)`

Constructor

```
public Burger(String name, double price, boolean cheese) {
  super(name, price);
  hasCheese = cheese;
}
```

Scoring Guidelines

Calls the superclass constructor with appropriate parameters to initialize instance variables

Constructor

```
public Burger(String name, double price, boolean cheese) {
  super(name, price);
  hasCheese = cheese;
}
```

Scoring Guidelines

Uses parameters and appropriate values to initialize the subclass instance variables

getCheeseStatus() Method

```
public boolean getCheeseStatus() {
    return hasCheese;
}
```

Scoring Guidelines

Declares header
`public boolean getCheeseStatus()`

getCheeseStatus() Method

```
public boolean getCheeseStatus() {
    return hasCheese;
}
```

Scoring Guidelines

Returns the appropriate instance variable

setCheeseStatus() Method

```
public void setCheeseStatus(boolean cheese) {
    hasCheese = cheese;
}
```

Scoring Guidelines

Declares header

`public void setCheeseStatus(boolean ____)`

setCheeseStatus() Method

```
public void setCheeseStatus(boolean cheese) {
    hasCheese = cheese;
}
```

Scoring Guidelines

Updates the appropriate instance variable using the parameter

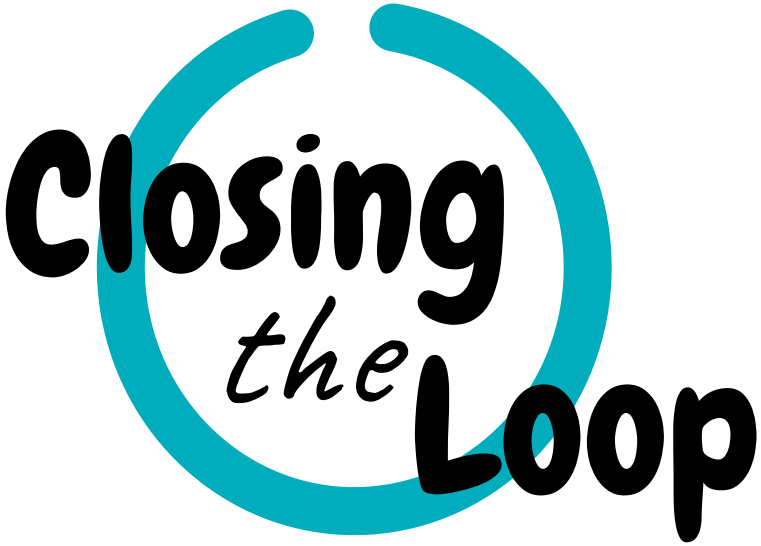
toString() Method

```
public String toString() {
    return "Thank you for visiting our food truck. Enjoy your " + getName();
}
```

Scoring Guidelines

Declares and implements **public String toString()** to return the constructed **String**

Wrap Up



What is your impression of this FRQ?

What knowledge and skills did you use that you learned in this unit?

What were you confident about? What would you like to practice?