In the lab you completed, you created a class called **Drinks** inside the models.py file. Inside the class, you passed **models.Model** to it as an argument. Which of the following is the correct code to create the model.

○   1   `class Drinks(Model):`

◉   1   `class Drinks(models.Model):`

○   1   `class Drinks():models.Model`

○   1   `class Drinks(models.model):`

✓Correct

Correct! In Django, each model is a Python class that subclasses the **django.db.models.Model**. To create a new instance of a model, you instantiate it like any other Python class. You define the class, and then pass the argument of **models.Model**.

2.In Django, each attribute of the model represents a database field.

In the lab you just completed, you created two attributes named **drink** and **price**. Which of the following is the correct code to create the attribute with the name of drink. The attribute contains a Form field type of **CharField**, and the maximum length set to 200 characters.

○   1   `class Drinks(models.Model): drink = CharField(max_length=200)`

○   1   `class Drinks(models.Model): drink = models.charfield(max_length=200)`

○   1   `class Drinks(models.Model): drink = models.CharField(max=200)`

◉   1   `class Drinks(models.Model): drink = models.CharField(max_length=200)`

⊘ Correct

Correct! To create the attribute with the name of **`drink`**, you first define the attribute name. Then define the built-in Field class of **`CharField()`** and pass the optional argument of **`max_length`** for validation.

3. In Django, the **`ModelAdmin`** class is the representation of a model in the admin interface.

In the lab you completed, you created a Model called **`Drink`**. For this model to be editable in the admin interface, you must register it with the admin inside the admin.py file. Which of the following are the pieces of code needed to register the model. Select all that apply.

☑   1   `admin.site.register(Drinks)`

⊘ Correct

Correct! Once you import the model, you define the **`register()`** function and pass the model name as an argument. Once registered, the model can be edited in the admin interface.

☐   1   `admin.site(Drinks)`

☑   1   `from .models import Drinks`

⊘ Correct

Correct! Using the **`import`** statement you must import the name of the required model class defined in the models.py file.

☐