What is the role of ‘models.Manager’ in the following code snippet?

Why do we set ‘objects’ equal to an instance of the ‘StudentManager’ class? Is this overwriting the ‘objects’ we were previously using with Django’s ORM (eg Student.objects.all() )?

What could we use the method ‘customMethod’ for (hint: it may be useful for login and registration)?

What is printed when the view function ‘process’ is run? Which one is printed first?

*# in models.py*

class StudentManager(models.Manager):

def customMethod(self, first, last):

print(“The student is named {} {}.”.format(first, last))

Student.objects.create(first\_name=first, last\_name=last)

return “Where is this returned?”

class Student(models.Model):

first\_name = models.CharField(max\_length=255)

last\_name = models.CharField(max\_length=255)

objects = StudentManager()

*# in views.py*

from .models import Student

def process(request):

print(Student.objects.customMethod(“Testy”, “McTesterson”))

return redirect(‘/’)