1 0 / 1 point

Which of the following is the correct way to instantiate the `Driver` class?

```
class Driver:
    def greeting(self):
        return "Hello, how are you today?"
```

- `bill = Driver()`
- ✗ `bill.Driver()`

Correct Answer: `bill = Driver()`

- `import Driver`
- `Driver.bill()`
- `driver('bill')`

5 1 / 1 point

The creation of functions in programming languages made it easier to do which of the following?

(select all that apply)

- [ ] Break a program
- [ ] Create a GOTO statement
- [x] Call a short script multiple times
- [ ] Set global variables
- [x] Reuse a subroutine

4 1 / 1 point

In the following example, which of the following outputs can you expect?

```
class Cat:

        def meow():
                return "Meow, Purr!"

fifi = Cat()
fifi.meow()
```

- [x] TypeError: meow() takes 0 positional arguments but 1 was given
- [ ] 'Meow, Purr!'
- [ ] <bound method Cat.meow of <__main__.Cat object at 0x7fdf900313a0>>
- [ ] AttributeError: 'Cat' object has no attribute 'meow'
- [ ] There is no output

3 1 / 1 point

In the following example, what is the expected output?

```
class Bike:
        def color(self):
                return "The bike is Red"

        def style(self):
                return "The bike is a Road bike"

my_new_bike = Bike()
my_new_bike.style()
```

- AttributeError: 'Bike' object has no attribute 'style'
- 'The bike is Red'
- There is no output
- <__main__.Bike object at 0x7fdf900313a0>
- ✓ 'The bike is a Road bike'

2 1 / 1 point

When writing code, what statement can be used as a placeholder to prevent an indentation error?

- `def`
- `instance`
- `class`
- ✓ `pass`
- `break`