

B 1 Class Definition

Which of the following is a good and valid definition for a class representing a car?

- (A) `def class Car(object):`
- (B) `class Car(object):`
- (C) `def Car(object):`
- (D) `class A(object)`

C 2 Class Instance

Using the class definition below, which line creates a new Car object with 4 wheels and 2 doors?

```
class Car(object):  
    def __init__(self, w, d):  
        self.wheels = w  
        self.doors = d  
        self.color = ""
```

- (A) `Car(mycar, 4, 2)`
- (B) `mycar = Car(4, 2, "white")`
- (C) `mycar = Car(4, 2)`
- (D) `mycar = Car(2, 4)`

E 3 Methods

Which of the following methods changes the color of the car, based on the definition below?

```
class Car(object):  
    def __init__(self, w, d):  
        self.wheels = w  
        self.doors = d  
        self.color = ""
```

- (A) `def paint(c):
 color = c`

```
(B) def paint(self, c):  
    color = c
```

```
(D) def paint(c):  
    self.c = c
```

```
(E) def paint(self, c):  
    self.color = c
```

C 4 Method Call

You create a car with `mycar = Car(4, 2)`. Which is a line of code to change the color of mycar to "red"?

```
class Car(object):  
    def __init__(self, w, d):  
        self.wheels = w  
        self.doors = d  
        self.color = ""  
    def paint(self, c):  
        self.color = c
```

(A) `Car.paint("red")`

(B) `mycar.paint(red)`

(C) `mycar.paint("red")`

(D) `mycar.paint(Car, "red")`

B 5 Special Methods

With the code below, what does the line `print(mycar == yourcar)` print?

```
class Car(object):  
    def __init__(self, w, d):  
        self.wheels = w  
        self.doors = d  
        self.color = ""
```

```
def paint(self, c):
    self.color = c
def __eq__(self, other):
    if self.wheels == other.wheels and \
        self.color == other.color and \
        self.doors == other.doors:
        return True
    else:
        return False

mycar = Car(4, 2)
mycar.paint("red")
yourcar = Car(4,2)
print(mycar == yourcar)
```

- (A) True
- (B) False
- (C) An error