

```
class Klass:
    def __init__(self, x):
        self.xcoord = x

    def act_on(self, value):
        self.xcoord += value

k = Klass(4)
```

Which of the following best describes the code elements above

- A. Klass is a class, xcoord an instance variable, act_on a method, k an instance
- B. Klass is a class, xcoord a method, act_on an instance variable, k an instance
- C. Klass and k are instances, xcoord an instance variable, act_on a method
- D. Klass and k are classes, xcoord an instance variable, act_on a method
- E. Klass is a class, xcoord and act_on are methods, k is an instance

Answer: A

Klass is a class definition, xcoord is an instance variable and act_on a method of that class, while k is an instance of that class.

```
class Klass:
    def __init__(self, x):
        self.xcoord = x

    def act_on(self, value):
        self.xcoord += value
```

```
k = Klass(4)
k.act_on(2)
```

After the above code executes, what is the value of `k.xcoord`?

- A. "x"
- B. 0
- C. 2
- D. 4
- E. 6

Answer: E

After initialization, the value of `k.xcoord` is 4. After the `act_on` method, `k.xcoord` is incremented by 2 to a final value of 6

```
class A:
    def __init__(self, x):
        self.x = x

    def act_on(self, value):
        self.x += value

class B(A):
    def __init__(self, x, y):
        super().__init__(x)
        self.y = y

    def act_on(self, value):
        self.x += self.y*value

val = B(4, 2)
val.act_on(2)
```

After the above code executes, what is the value of `val.x`?

- A. 6
- B. 8
- C. 10
- D. 16
- E. 20

Answer: B

After initialization, the value of `val.x` is 4. The `act_on` method in B overrides that in A, so the final expression is $4 + 2 * 2$