

Questions About Classes

Here is a rudimentary class:

```
public class Person {  
    private string name;  
    private int age;  
    .....  
}
```

In some other class I construct a person with name "bob" stored in variable x. Could I then say x.age=70?

- A. Yes
- B. No, but I could if I removed "private" from the declaration of age.
- C. No, but I could if I changed the declaration of age to "public int age".
- D. No, people aren't allowed to be that old.

Answer C: No, but I could if I changed the declaration of age to "public int age". But this is still a bad thing to do. We will often access private variables of a class through "getters and setters". In other words, the class will contain public methods

```
public int getAge() {  
    return age;  
}
```

```
public void setAge(int a) {  
    age = a;  
}
```

To set the age of x we'll say `x.setAge(69);`

Note that answer (B) in the first question (just removing the word “private”, giving us default access) would allow other classes in the same package to access variable age, but not classes in other packages. This still isn't a good solution; use getters and setters.

I have a class Critter that has constructor

```
public Critter(String species);
```

and a method

```
public int numLegs();
```

What code can I put in `public static void main(String [] args)` to test this?

A. `System.out.println(numLegs);`

B. `System.out.println(numLegs());`

C. `System.out.println(new Critter("cat").numLegs());`

D. `Critter x = new Critter("cat");`
`System.out.println(x.numLegs());`

Answer:

C and D both work, though C is a bit odd -- it constructs a new cat, prints its number of legs, and then throws out the cat it constructed.