3. Given the Map declaration below, how do you write a for-each loop to iterate through its elements?

```
Map<String, Integer> petAges = new HashMap<String, Integer>();
petAges.put("Mo", 14);
petAges.put("Quill", 12);
petAges.put("Tuxedo", 8);
```

Show Results 22/22 Students Answered

A for (Map.Entry<String, Integer> petAge : petAges)

B for (Map.Entry<String, Integer> petAge : petAges.entrySet())

C for (Map.Entry<String, Integer> petAge in petAges)

D for (petAges.entrySet() => Map.Entry<String, Integer> petAge)

i Show explanation

4. What happens when the following code executes?

```
Map<String, Integer> animalNumberLegs = new HashMap<String,
Integer>();
animalNumberLegs.put("Dog", 4);
animalNumberLegs.put("Fish", 0);
animalNumberLegs.put("Bear", 2);
animalNumberLegs.put("Bear", 4);
```

"Fish" => 0

"Dog" => 4

Show Results 21/22 Students Answered

"Bear" => 4

A The value for "Bear" is overwritten with 4.

B A second "Bear" is added, with a value of 4.

C A compiler error occurs because the syntax is invalid.

D The value of bear increases by 4 and now equals 6.

i Show explanation

6. How do you determine if the key "Dave" exists in the Map below?

```
Map<String, String> people = new HashMap<String, String>();
```

Show Results 20/22 Students Answered

A people.getKey("Dave") ✗

B people.hasKey("Dave") ✗

C people.keyExists("Dave") ✗

D people.containsKey("Dave") ← ✓

i Show explanation

```
 *
 */
public Map<String, Integer> peterPaulPartnership(Map<String, Integer> peterPaul) {

    if (peterMoney >= 5000 && paulMoney >= 10000) {
        calculate 1/4 of peter's money
        calculate 1/4 or paul's money
        subtract 1/4 from peter's money
        subtract 1/4 from paul's money
        add a new Map element called PeterPaulPartnership that has a value
            of 1/4 of peters and 1/4 of pauls money
    }
```

peter >= 5000 paul >=10000 peterpaulpartnership 1/4 of peter's money + 1/4 of paul's money

```
*/
public Map<String, Integer> peterPaulPartnership(Map<String, Integer> peterPaul) {

    if (peterMoney >= 5000 && paulMoney >= 10000) {
        calculate 1/4 of peter's money
        calculate 1/4 or paul's money
        subtract 1/4 from peter's money
        subtract 1/4 from paul's money
        add a new Map element called PeterPaulPartnership that has a value
            of 1/4 of peters and 1/4 of pauls money
    }
```

peter >= 5000 paul >=10000 peterpaulpartnership 1/4 of peter's money + 1/4 of paul's money

get the original value, subtract the 1/4 and put the new value back in