Signature

Name:

**CSE102 Programming Languages II Quiz #5**

1. What is wrong with the following statement?

HashMap<String> map = **new** HashMap<>();

1. HashMaps cannot be created empty.
2. The right-hand side of assignment must be **new** HashMap<String>();
3. HashMaps must be specified with two data types, one for key and the other for value.
4. You have to specify an initial size while creating a HashMap.
5. The following function is supposed to fill a set using elements of an array, to get rid of any repetitions. What should come in place of the question marks?

**static** HashSet<String> setOf(String[] arr){

HashSet<String> set = **new** HashSet<>();

**for**(String s:??)

set.add(s);

**return** set;

}

1. set
2. arr
3. arr[i]
4. set[i]

Signature

Name:

**CSE102 Programming Languages II Quiz #5**

1. What is wrong with the following statement?

HashMap<String> map = **new** HashMap<>();

1. HashMaps cannot be created empty.
2. The right-hand side of assignment must be **new** HashMap<String>();
3. HashMaps must be specified with two data types, one for key and the other for value.
4. You have to specify an initial size while creating a HashMap.
5. The following function is supposed to to fill a set using elements of an array, to get rid of any repetitions. What should come in place of the question marks?

**static** HashSet<String> setOf(String[] arr){

HashSet<String> set = **new** HashSet<>();

**for**(String s:??)

set.add(s);

**return** set;

}

1. set
2. arr
3. arr[i]
4. set[i]

**CSE102 Programming Languages II Quiz #5**

1. Write a function which takes two mappings; *map1* and *map2* as parameters and returns their composite mapping, *map3.* So *map3(key) = value* iff *map2(map1(key)) = value.*

Some useful methods of HashMap<K, V>:

get(K key) keySet()

put(K key, V value) containsKey(K key)

**static** HashMap<String, String> compose(

HashMap<String, Integer> map1,

HashMap<Integer, String> map2)

{

HashMap<String, String> res = **new** HashMap<>();

**for**(String key: map1.keySet()) {

Integer n = map1.get(key);

**if**(map2.containsKey(n))

res.put(key, map2.get(n));

}

**return** res;

}

**CSE102 Programming Languages II Quiz #5**

1. Write a function which takes two mappings; *map1* and *map2* as parameters and returns their composite mapping, *map3.* So *map3(key) = value* iff *map2(map1(key)) = value.*

Some useful methods of HashMap<K, V>:

get(K key) keySet()

put(K key, V value) containsKey(K key)

**static** HashMap<String, String> compose(

HashMap<String, Integer> map1,

HashMap<Integer, String> map2)

{

}