

1. Explain the four different constructors to create a HashMap with examples?

HashMap(): This constructor constructs a default HashMap.

```
HashMap<datatype1, datatype2> hm=new HashMap<datatype1, datatype2>();
```

HashMap(Map m): This constructor initializes the hash map by using the elements of the given Map object m.

```
Ex:HashMap<datatype1, datatype2> hm=new HashMap<datatype1, datatype2>();
```

```
HashMap<datatype1, datatype2> hm1=new HashMap<datatype1, datatype2>(hm);
```

HashMap(int capacity):

This constructor initializes the capacity of the hash map to the given integer value, capacity.

```
Ex: HashMap<datatype1, datatype2> hm=new HashMap<datatype1, datatype2>(capacity);
```

HashMap(int capacity, float fillRatio):

This constructor initializes both the capacity and fill ratio of the hash map by using its arguments.

```
ex: HashMap<datatype1, datatype2> hm=new HashMap<datatype1, datatype2>(2,2.5f);
```

2. What will happen if you try to store a key which is already present in HashMap?

If you store an existing key in the HashMap then it will override the old value with the new value and put() will return the old value. There will not be any exception or error.

3. Is it possible to store null key and null values in a HashMap?

Yes, HashMap allows one null key which is stored at the first location of bucket array e.g. bucket[0] = value. The HashMap doesn't call hashCode() on null key because it will throw NullPointerException, hence when a user call get() method with null then the value of the first index is returned.

Yes, HashMap also allows null value, you can store as many null values as you want as shown in the hashmap example post in this blog.