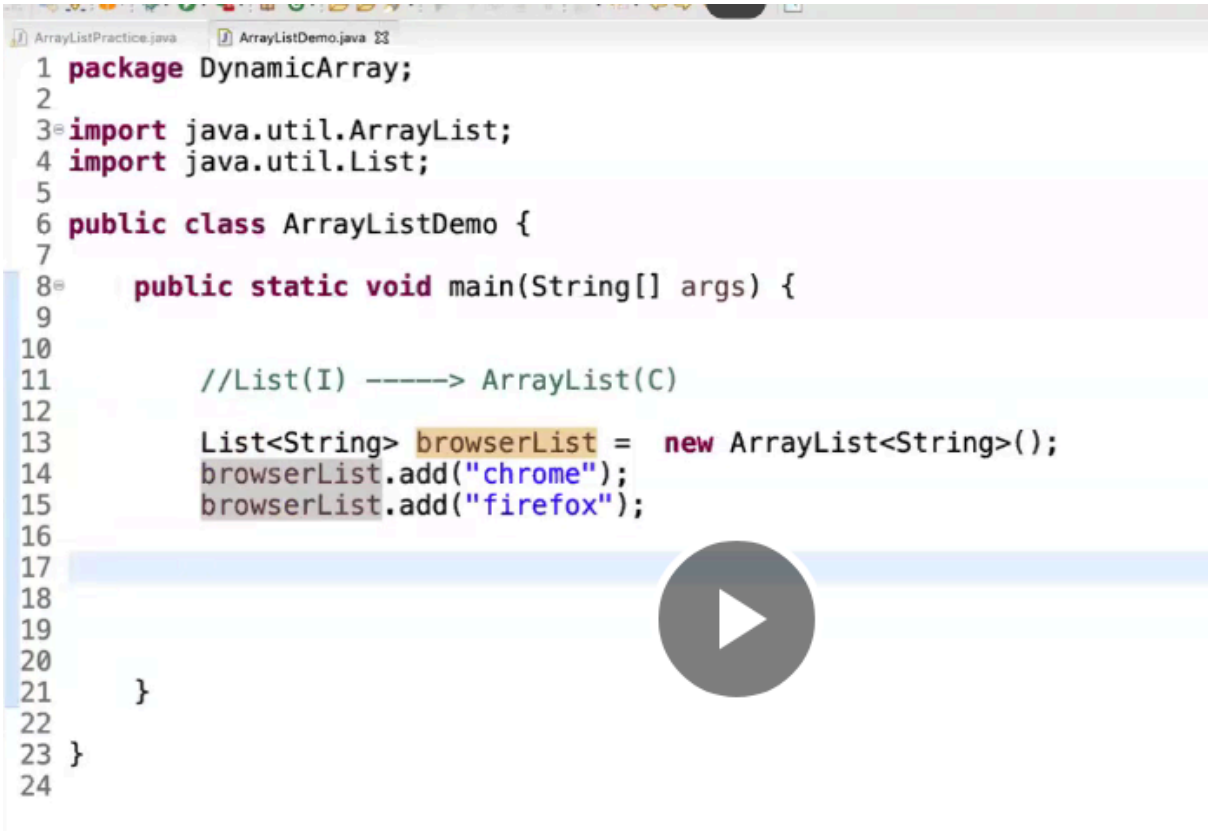
Top casting-



paste topcasting

topcasting.java

```java
package com.day29;

import java.util.ArrayList;
import java.util.List;

public class topcasting {

    public static void main(String[] args) {

        //top casting.
        List<String> browserlist=new ArrayList<>();
        browserlist.add("chrome");
        browserlist.add("firefox");
        System.out.println(browserlist);
    }

}

//[chrome, firefox]
```

map-

ArrayListPractice.java | ArrayListDemo.java | *MapConcept.java

```java
import java.util.HashMap;

public class MapConcept {

    public static void main(String[] args) {

        //collection: <key,value> pair
        //HashMap(C) ---> Map(I)
        //its non order based collection -- no order

        HashMap<String, Integer> map = new HashMap<String, Integer>();

        map.put("chrome", 101);
        map.put("firefox", 96);
        map.put("edge", 111);
```

Other data types-

```
19
20      HashMap<Integer, Integer> map1 = new HashMap<Integer, Integer>();
21      map1.put(100, 1);
22
23
24      HashMap<String, Object> data = new HashMap<String, Object>();
25
26      data.put("name", "veena");
27      data.put("age", 35);
28      data.put("designation", "SDET2");
29      data.put("isperm", true);
30      data.put("gender", 'f');
```

paste map1

map1.java

```
1  package com.day29;
2
3  import java.util.HashMap;
4
5  public class map1 {
6
7      public static void main(String[] args) {
8
9          HashMap<String, Integer> h1= new HashMap<>();
10         h1.put("chrome", 10);
11         h1.put("firefox", 324324);
12
13         System.out.println(h1);
14
15         HashMap<Integer, Integer> h2= new HashMap<>();
16         h2.put(2343, 34324);
17         h2.put(32424, 3423423);
18
19         System.out.println(h2);
20
21         HashMap<String, Object> h3= new HashMap<>();
22         h3.put("test", 34324);
23         h3.put("karan", 25.78);
24         h3.put("james", 45.67f);
25         h3.put("bond", true);
26         h3.put("camila", "giraffe");
27         h3.put("charles", 't');
```

```
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}
//{chrome=10, firefox=324324}
//{2343=34324, 32424=3423423}
//{test=34324, camila=giraffe, james=45.67, karan=25.78, charles=t, bond=true}
```

Writable | Smart Insert | 1 : 1 : 0

```
        System.out.println(data.get("name"));
```

Object java.util.HashMap.get(Obje

Veena

Key not there-

```
        System.out.println(data.get("phone"));
```

Null

paste map4-

map4.java

```
package com.day29;

import java.util.HashMap;

public class map4 {

    public static void main(String[] args) {

        //duplicate key allowed.
        //latest value overrides all.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put("test", 34324);
        h3.put("test", 23432.32432);
        h3.put("karan", 25.78);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{test=23432.32432, camila=giraffe, james=45.67, karan=25.78, charles=t, bond=true}
```

paste map5-

map5.java

```java
package com.day29;

import java.util.HashMap;

public class map5 {

    public static void main(String[] args) {

        //duplicate value allowed.

        HashMap<String, Object> h3= new HashMap<>();
        h3.put("test", 34324);
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{test=34324, camila=giraffe, james=45.67, karan=34324, charles=t, bond=true}
```

paste map6-

map6.java

```java
package com.day29;

import java.util.HashMap;

public class map6 {

    public static void main(String[] args) {

        //duplicate key and duplicate value allowed.

        HashMap<String, Object> h3= new HashMap<>();
        h3.put("test", 34324);
        h3.put("test", 34324);
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{test=34324, camila=giraffe, james=45.67, karan=34324, charles=t, bond=true}
```

Key is null-

Allowed.

```
31          data.put(null, "pune");

38          System.out.println(data.get(null));
```

Pune

Duplicate null key-

Allowed.

```
32          data.put(null, "India");

38          System.out.println(data.get(null));
```

India

Overwrite the old value.

Null values-

Allowed.

```
34          data.put("phone", null);

39          System.out.println(data.get("age"));
40          System.out.println(data.get("phone"));//null
```

Null

Multiple null values-

Allowed.

```
33 //      data.put(null, "India");
34         data.put("phone", null);
35         data.put("city", null);

41         System.out.println(data.get("phone"));//null
42         //System.out.println(data.get(null));//India
43         System.out.println(data.get("city"));//null
```

Null

Null

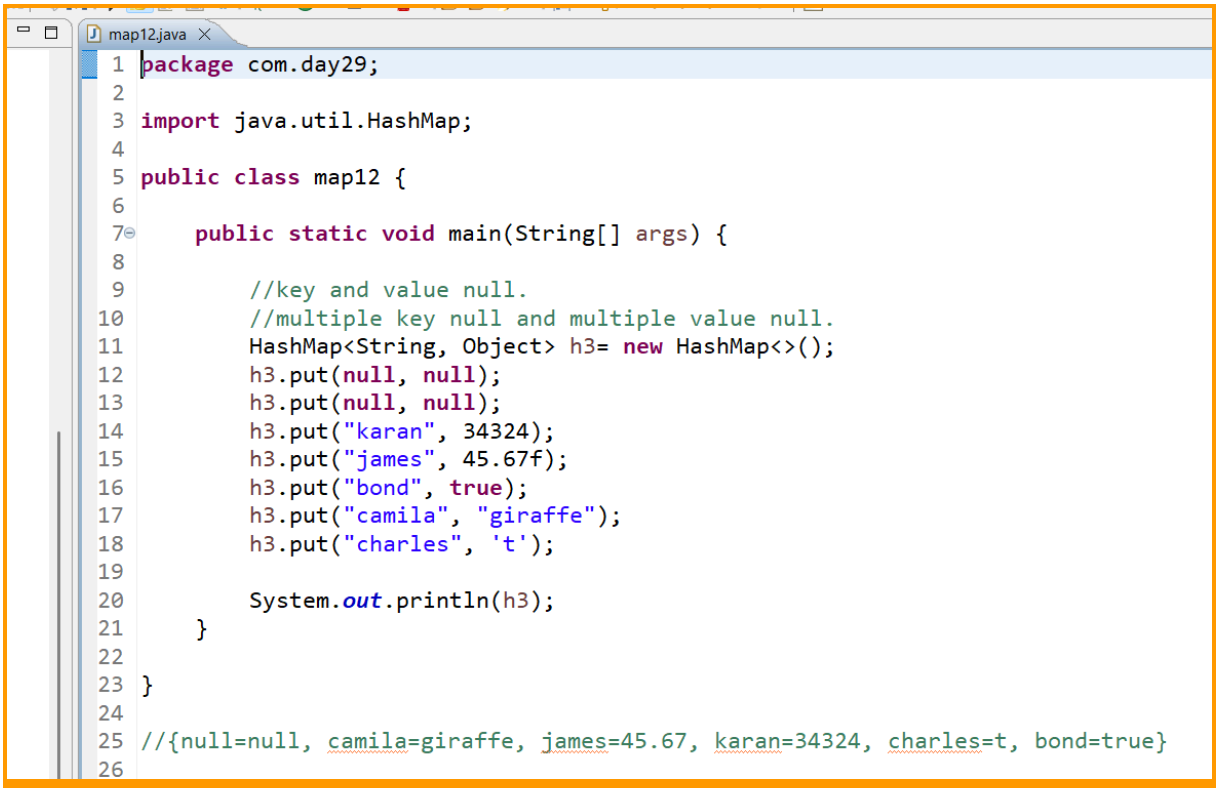
```
 9         //collection: <key,value> pair
10         //HashMap(C) ---> Map(I)
11         //its non order based collection -- no order
12         //one null key is allowed
13         //but multiple values can be allowed
```

Key and value null-

```
37         data.put(null, null);

44         System.out.println(data.get("phone"));//null
45         System.out.println(data.get(null));//null
```

Null

paste map12



map12.java

```java
package com.day29;

import java.util.HashMap;

public class map12 {

    public static void main(String[] args) {

        //key and value null.
        //multiple key null and multiple value null.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{null=null, camila=giraffe, james=45.67, karan=34324, charles=t, bond=true}
```

Blank value-

```
38          data.put("address", "");

48          System.out.println(data.get("address"));//null
```

paste map14

map14.java

```java
package com.day29;

import java.util.HashMap;

public class map14 {

    public static void main(String[] args) {

        //duplicate blank value.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("Tester", "");
        h3.put("Tester1", "");
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{null=null, Tester=, camila=giraffe, james=45.67, karan=34324, charles=t, Tester1=, bond=true}
```

Space value-

```java
38          data.put("address", " ");
```

```java
48          System.out.println(data.get("address"));//null
```

-
-

```java
 9          //collection: <key,value> pair
10          //HashMap(C) ---> Map(I)
11          //its non order based collection -- no order
12          //one null key is allowed
13          //but multiple null values can be allowed
```

paste [map16]

map16.java

```java
package com.day29;

import java.util.HashMap;

public class map16 {

    public static void main(String[] args) {

        //multiple space values.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("Tester", "");
        h3.put("Tester1", " ");
        h3.put("Tester2", " ");
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{null=null, Tester=, camila=giraffe, james=45.67, karan=34324, charles=t,
//Tester2= , Tester1= , bond=true}
```

Blank key-

```java
39          data.put("", "32.33");

50          System.out.println(data.get(""));
```

```
32.33
```

paste map18

map18.java

```java
package com.day29;

import java.util.HashMap;

public class map18 {

    public static void main(String[] args) {

        //multiple blank key.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("", "karan");
        h3.put("", "jigar");
        h3.put("Tester1", " ");
        h3.put("karan", 34324);
        h3.put("james", 45.67f);
        h3.put("bond", true);
        h3.put("camila", "giraffe");
        h3.put("charles", 't');

        System.out.println(h3);
    }

}

//{null=null, =jigar, camila=giraffe, james=45.67, karan=34324, charles=t, Tester1= , bond=true}

```

key and value blank-

paste map19-

map19.java

```java
package com.day29;

import java.util.HashMap;

public class map19 {

    public static void main(String[] args) {

        //key and value blank.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("", "");


        System.out.println(h3);
    }

}

//{null=null, =}

```

paste map20-

map20.java

```java
package com.day29;

import java.util.HashMap;

public class map20 {

    public static void main(String[] args) {

        //key and value blank.
        //duplicates.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put("", "");
        h3.put("", "");


        System.out.println(h3);
    }

}

//{null=null, =}

```

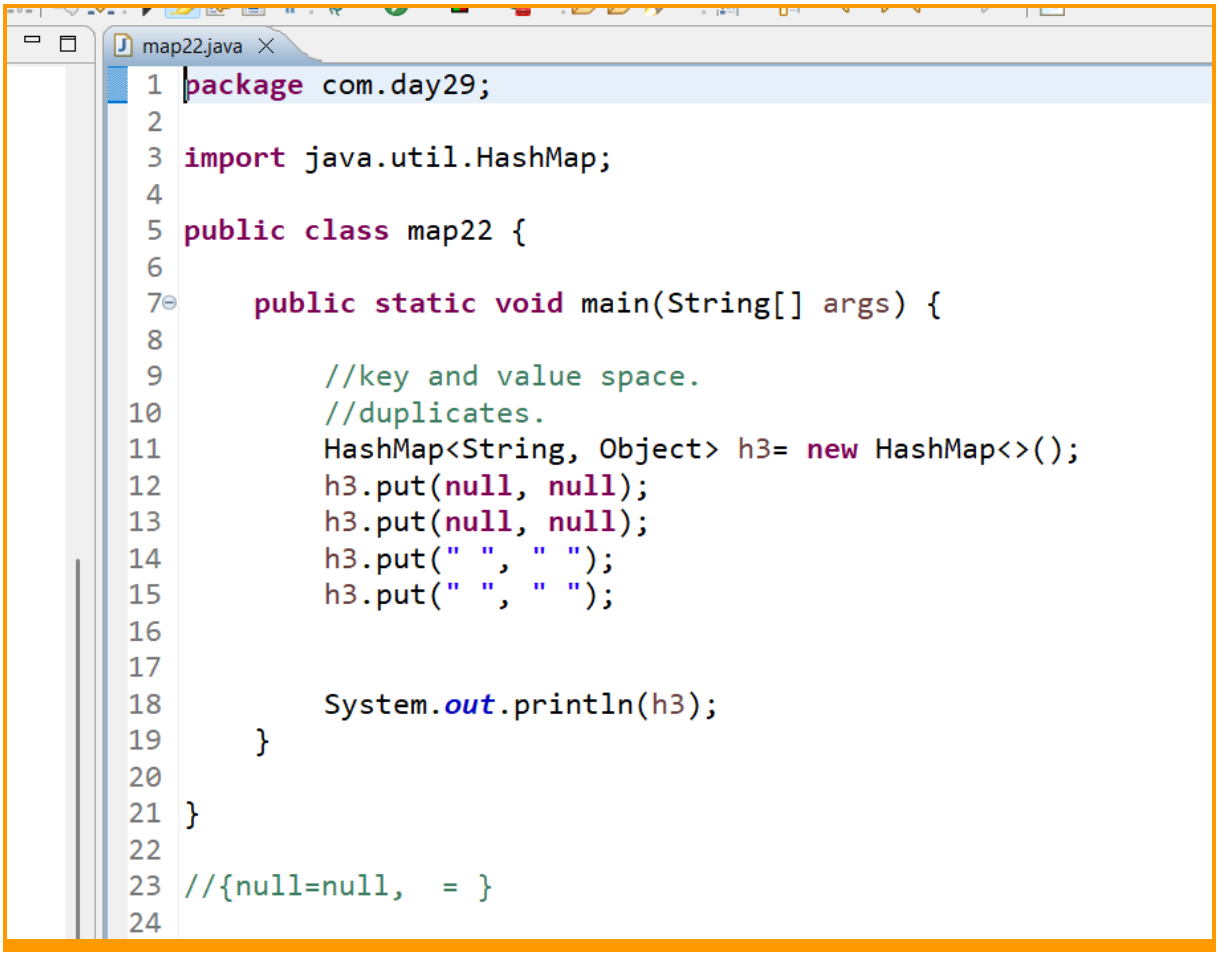
key and value space-

paste map21-

map21.java

```java
package com.day29;

import java.util.HashMap;

public class map21 {

    public static void main(String[] args) {

        //key and value space.
        HashMap<String, Object> h3= new HashMap<>();
        h3.put(null, null);
        h3.put(null, null);
        h3.put(" ", " ");


        System.out.println(h3);
    }

}

//{null=null,  = }
```

paste map22-



map22.java

```
1  package com.day29;
2
3  import java.util.HashMap;
4
5  public class map22 {
6
7      public static void main(String[] args) {
8
9          //key and value space.
10         //duplicates.
11         HashMap<String, Object> h3= new HashMap<>();
12         h3.put(null, null);
13         h3.put(null, null);
14         h3.put(" ", " ");
15         h3.put(" ", " ");
16
17
18         System.out.println(h3);
19     }
20
21 }
22
23 //{null=null,  = }
24
```

Duplicate-

Overwrite the old value.

```
40          data.put("age", 37);

46          System.out.println(data.get("age"));
```

37

Print hash map directly-

```
53          System.out.println("-----");
54          System.out.println(data);
```

```
-----
{null=pune, address= , gender=f, phone=null, city=null, name=vee
```

No order in output.

Null keys and null values comes first.

```
 9          //collection: <key,value> pair
10          //HashMap(C) ---> Map(I)
11          //its non order based collection -- no order
12          //one null key is allowed
13          //but multiple null values can be allowed
14          //<null,value> : it will stored at first position
```

Default capacity of hash map is 15.

Every block called as node in hash map. Every node has three parts.

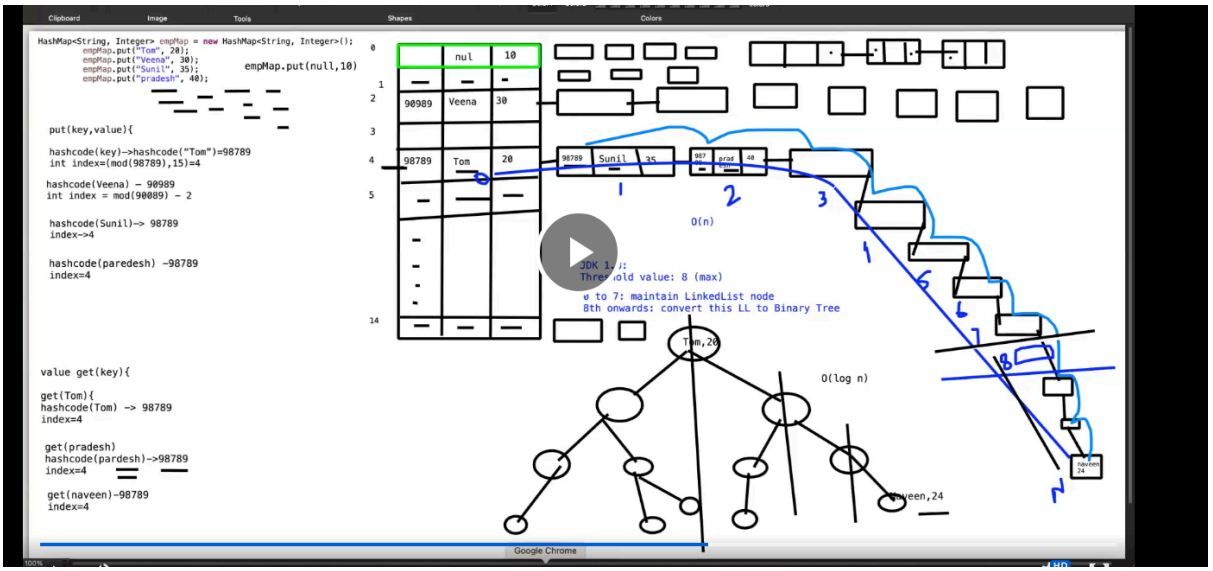
Hash code – special code for the key. Duplicate hash codes can be present.

Upto jdk 1.7 we can add n number of nodes and make it link list when the hash code is same. This is collision.

Changed from jdk 1.8.

Link list is linear search. Binary tree is faster to search.

When key is null, then it always go in the first index.




```
57 //
58         HashMap<String, Integer> empMap = new HashMap<String, Integer>();
59         empMap.put("Tom", 20);
60         empMap.put("Veena", 30);
61         empMap.put("Sunil", 35);
62         empMap.put("pradesh", 40);
63
64         System.out.println(empMap);
65
66
67
```

```
{Veena=30, Tom=20, Sunil=35, pradesh=40}
```

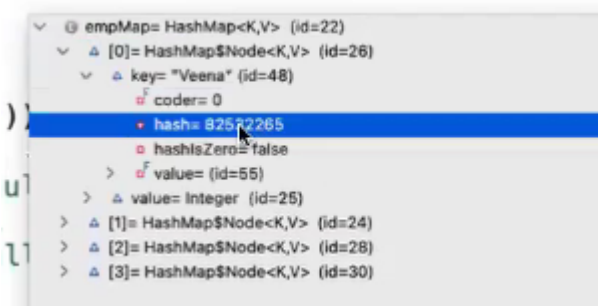
How to check hash code-

Put debugger on line 59.

Mouse hover empMap in line 64.

```
57 //
58         HashMap<String, Integer> empMap = new HashM
59         empMap.put("Tom", 20);
60         empMap.put("Veena", 30);
61         empMap.put("Sunil", 35);
62         empMap.put("pradesh", 40);
63
64         System.out.println(empMap);
65
```

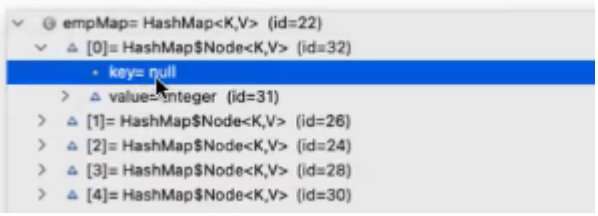
82532265



Lets debug with null key-

```
57 //
58         HashMap<String, Integer> empMap = new HashMap<String, Integer>();
59         empMap.put("Tom", 20);
60         empMap.put("Veena", 30);
61         empMap.put("Sunil", 35);
62         empMap.put("pradesh", 40);
63         empMap.put(null, 34);
64
65
66         System.out.println(empMap);
```

Null at first place. No hash code calculated.



Size-

```
66          System.out.println(empMap.size());
```

5

Print hash map-

```
66          System.out.println(empMap);
```

```
{null=34, Veena=45, Tom=20, Sunil=35, pradesh=40}
```

Min and max value of character- 1.52

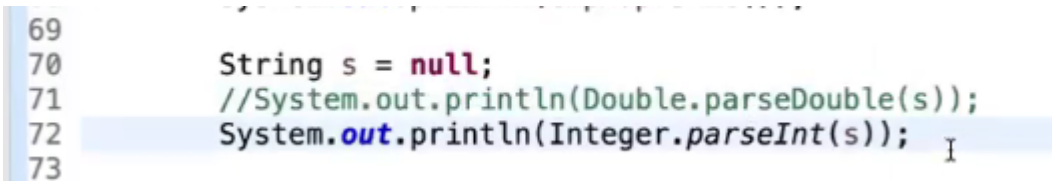
Cant say because there are different character sets like number, alphabets and so on, so we cant have definite answer for this.

```
69
70          System.out.println(Character.MIN_VALUE);
71          System.out.println(Character.MAX_VALUE);
```

nothing comes in output.

```
69
70          String s = null;
71          System.out.println(Double.parseDouble(s));
```

Null pointer exception.



Number format exception.