

# Java Data Structures: HashSets and HashMaps

---

**CPSC 233: Introduction to Computer Science for Computer Science  
Majors II  
Winter 2022**

Jonathan Hudson, Ph.D.  
Instructor  
Department of Computer Science  
University of Calgary

**Wednesday, 10 November 2021**

*Copyright © 2021*



# HashSet

---

# Where are sets?

---

- Java's set is more syntactically verbose like an ArrayList
- It has HashSet
  - Does not have all the 'set theory' type operators like disjoint, power set, etc.

```
HashSet<Key> set = new HashSet<Key>();
```

<Key> (ex. <String>) is the generic type of the items used as keys

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/HashSet.html>

# HashSet Operations

---

- Java's set is like an ArrayList (more syntactically verbose)
- It has HashSet
  - Does not have all the 'set theory' type operators like disjoint, power set, etc.

```
HashSet<Key> set = new HashSet<Key>();
```

Java	Java (desc)	Python	Python (desc)
set.add(item);	Add item to set (no dupl)	set.add(item)	same
set.clear()	Empty set	set.clear()	same
set.contains(item)	Is item in set	item in set	same
set.remove(item)	Remove item from set	set.remove(item)	same
set.size()	Size of set	len(set)	same

# HashSet Operations

---

```
HashSet<Integer> set = new HashSet<Integer>();  
set.add(1); set.add(5); set.add(7); set.add(1);  
System.out.println(set);  
System.out.println(set.contains(5));  
System.out.println(set.contains(99));  
set.remove(5);  
System.out.println(set);
```

[1, 5, 7]

true

false

[1,7]

# Looping HashSet

---

```
HashSet<Integer> set = new HashSet<Integer>();  
  
for (Integer key : set) {  
    System.out.println(key);  
}
```

# HashMap

---

# Where are dictionaries?

---

- Java's dictionary is more syntactically verbose like an ArrayList
- It has HashMap

```
HashMap <Key, Value> map = new HashMap <Key, Value>();
```

<Key> and <Value> are the generic types to use as key and value

<https://docs.oracle.com/en/java/javase/11/docs/api/java.base/java/util/HashMap.html>



# HashMap Operations

- Java's dictionary is like an ArrayList (more syntactically verbose)
- It has HashMap

```
HashMap <Key, Value> map = new HashMap <Key, Value>();
```

Java	Java (desc)	Python	Python (desc)
map.clear()	Clear map	dict.clear()	same
map.containsKey(key)	Is key in map	key in dict	same
map.containsValue(value)	Is value in map	value in dict.values()	same
map.entrySet()	Get Entry<Key,Value> pairs	dict.items()	same
map.keySet()	Get Set<Key>	dict.key()	same
map.put(key, value)	Put value in map at key (overwrites)	dict[key] = value	same
map.remove(key)	Remove key (and value)	del dict[key]	same
map.values()	Get Collection<Value>	dict.values()	same

# HashMap Operations

```
HashMap<Integer, String> map = new HashMap<Integer, String>();
map.put(101, "Alice");
map.put(123, "Bob");
map.put(310, "Alice");
System.out.println(map);
map.put(310, "Carol");
System.out.println(map);
System.out.println(map.containsKey(101));
System.out.println(map.containsKey(999));
System.out.println(map.containsValue("Carol"));
System.out.println(map.keySet());
System.out.println(map.values());
System.out.println(map.entrySet());
System.out.println(map.remove(101));
System.out.println(map);
```

{101=Alice, 310=Alice, 123=Bob}

{101=Alice, 310=Carol, 123=Bob}

true

false

true

[101, 310, 123]

[Alice, Carol, Bob]

[101=Alice, 310=Carol, 123=Bob]

Alice

{310=Carol, 123=Bob}

# Looping HashMap

---

```
for (Integer key : map.keySet()) {  
    System.out.println(key + " stores " + map.get(key));  
}
```

```
for (Map.Entry<Integer, String> entry : map.entrySet()) {  
    System.out.println(entry.getKey() + " stores " + entry.getValue());  
}
```

# Onward to ... Software Development

---

Jonathan Hudson  
[jwhudson@ucalgary.ca](mailto:jwhudson@ucalgary.ca)  
<https://pages.cpsc.ucalgary.ca/~jwhudson/>



UNIVERSITY OF  
CALGARY