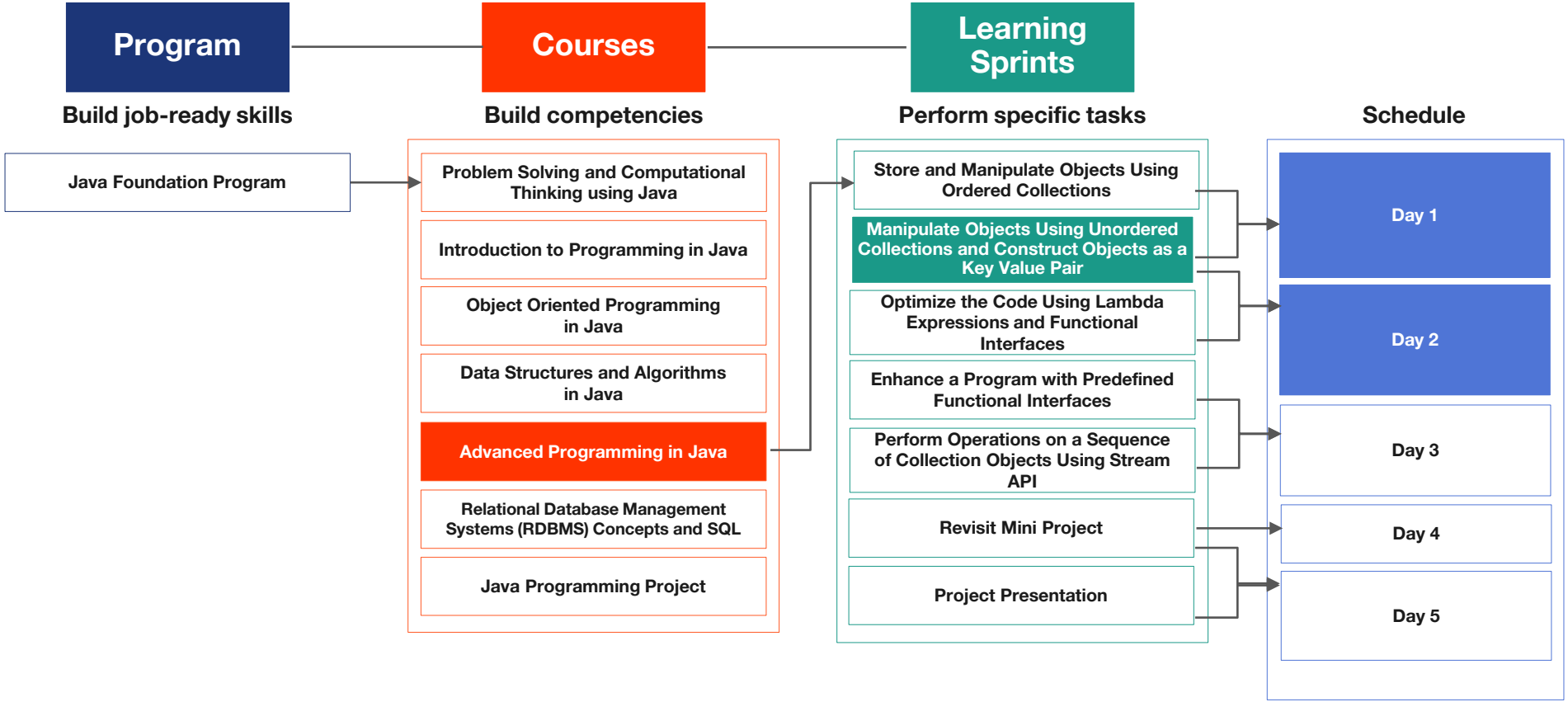
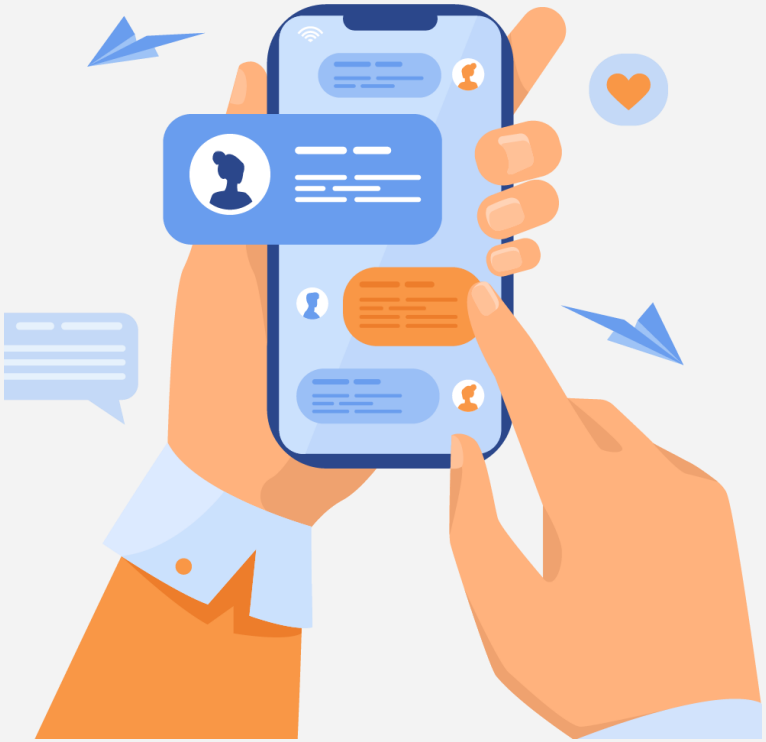
Java Program: Course 5: Plan



Think and Tell

Andrew, a software developer wishes to develop an application that will help him create a chat group dynamically from the list of available contacts.

He also wants to include a functionality that will add or remove members from the chat group.



A Chat Application

How can Andrew create a collection of contacts and ensure the uniqueness of the contacts in the chat group?

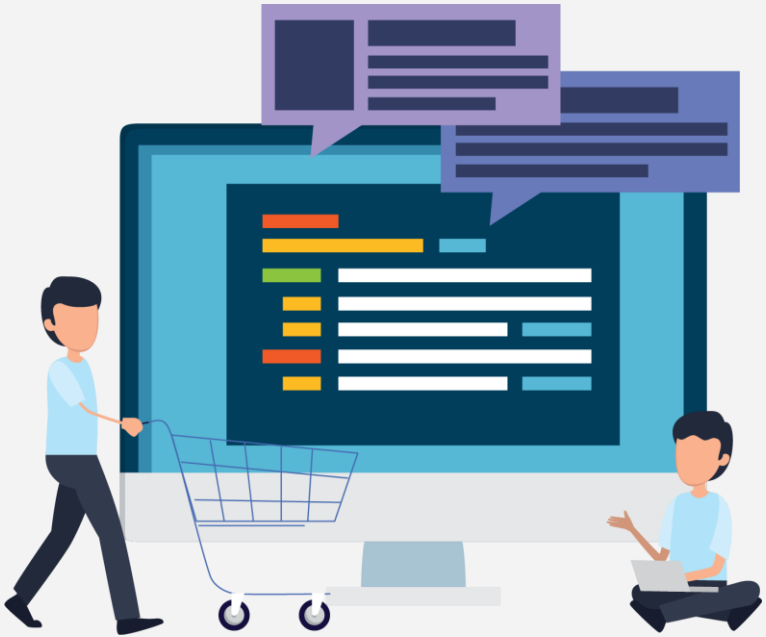
Which collection interface should be set here?

A Chat Application

Now, Andrew wishes to maintain some more details of the chat users, for example, their address, date of birth etc.

He wants to retrieve all the details based on a single key, that is, the contact number.

What type of collection should Andrew use to fulfill this requirement?



Customer Details

Andrew, further wishes to link this chat application to an e-commerce portal where the customers buy various products.
He wants to analyze the customer's purchasing history. To do this, he needs to use the contact number of the customers to retrieve their purchase history as and when required.

What type of collection should Andrew use to fulfill this requirement?

Manipulate Objects Using Unordered Collections and Construct Objects as a Key Value Pair

Learning Objectives

- Define and use the Set interface
- Describe and implement HashMap and TreeMap
- Identify and use Comparable and Comparator interface

Using the Set Interface

- Set interface is used to create a collection of unique objects
- The package, java.util, provides the Iterator interface to traverse through the set collection
- The reference of the Iterator interface can be obtained using the iterator() method of the Set interface
- The Iterator interface contains various methods, such as hasNext() and next(), which help to traverse the set collection

HashSet Class

- HashSet class provides implementation of the Set interface; it enables us to create a set in which insertion is faster because it does not sort the elements
- Some of the commonly used constructors of the HashSet class are:
  - HashSet()
  - HashSet(Collection<? extends E> c)
  - HashSet(int initialCapacity)
- Some of the commonly used methods of the HashSet class are:
  - boolean add(E e)
  - void clear()
  - boolean remove(Object o)
  - int size()

HashSet Class (contd.)

- Java also provides support for non-generic collections
- You can use the non-generic HashSet collection to store any type of object, as shown in the following code snippet:

```
Contacts a=new Contacts();
HashSet hs=new HashSet();
hs.add(a);
hs.add("String object");
hs.add(new Integer(3));
```

Interactive Demo

Write a program in Java to add, traverse and remove contact details by using HashSet collection object.

TreeSet Class

- TreeSet class provides implementation of the Set interface; it enables us to create a sorted set of objects
- The insert or add object process is slow as it sorts the objects on every insertion
- Some of the constructors of the TreeSet class are:
  - TreeSet()
  - TreeSet(Collection<? extends E> c)
- Some of the methods of the TreeSet class are:
  - boolean add(E e)
  - void clear()
  - boolean remove(Object o)
  - int size()

Quick Check!

Which one of the following classes enables you to create a collection of unique objects?

1. List
2. Vector
3. TreeSet
4. ArrayList

Interactive Demo

Write a program in Java to add, traverse and remove data from a TreeSet collection object.

Using the Map Interface

- Map interface enables you to create a collection with key-value pair objects
- You need to use the key object to access the corresponding value object
- This interface allows duplicate the value objects but the key object must be unique

Quick Check!

Fill in the blank:

A --------- interface is a collection that stores multiple key-value pairs.

HashMap Class

- HashMap class enables us to create a collection in which a value is accessible using the key
- It allows us to store objects in an unordered form; it also allows one null value for a key and any number of null values for value.
- Some of the commonly used constructors of the HashMap class are:
  - HashMap()
  - HashMap(int initialCapacity)
  - HashMap(Map<? extends K, ? extends V> m)
- Some of the commonly used methods of the HashMap class are:
  - void clear()
  - V get(Object key)
  - V put(K key, V value)
  - V remove(Object o)

Interactive Demo

Write a program in Java to add, traverse and remove data from a HashMap collection object.

TreeMap Class

- The TreeMap class enables us to create a collection of objects in a sorted order with unique keys; the sorted collection improves the retrieval process of objects
- Some of the commonly used constructors of the TreeMap class are:
  - TreeMap()
  - TreeMap(Map<? extends K, ? extends V> m)
- Some of the commonly used methods of the TreeMap class are:
  - void clear()
  - V get(Object key)
  - V put(K key, V value)
  - V remove(Object o)
  - int size()

Interactive Demo

Write a program in Java to add, traverse and remove data from a TreeMap collection object.

Hashtable Class

- Hashtable class enables us to create an unordered collection of objects, which cannot contain the null objects; the methods of the Hashtable class are synchronized
- Some the commonly used constructors of the Hashtable class are:
  - Hashtable()
  - Hashtable(int initialCapacity)
  - Hashtable(Map<? extends K, ? extends V> t)
- Some the commonly used methods of the Hashtable class are:
  - void clear()
  - V get(Object key)
  - V put(K key, V value)
  - V remove(Object o)
  - int size()

Quick Check!

Which one of the following classes contains synchronized methods?

1. Hashtable
2. HashSet
3. TreeMap
4. ArrayList

Interactive Demo

Write a program in Java to add, traverse and remove data from a Hashtable collection object.

Sorting Collection Data

Andrew wishes to create an application to maintain the contact details, such as name, contact number, and address of his friends. In addition, you need to ensure that the application should have the flexibility to add or remove any number of contacts. For this, Andrew has decided to create a collection of contacts and he also wishes to arrange the contact details as per the name.

- For this, Java provides the following interfaces to sort the objects in a collection:
  - The Comparable interface
  - The Comparator interface

Using Comparable Interface

- Comparable interface is defined in the java.lang package and is used to sort and compare a collection of objects
- This interface provides the compareTo() method that compares the references of the objects
- The compareTo() method returns a value of the int type with the following characteristics:
  - Negative: If the current object is less than the object being compared
  - Zero: If the current object is equal to the object being compared
  - Positive: If the current object is greater than the object being compared

Interactive Demo

Write a program in Java to store the names and marks of students. Next, sort the details based on the marks obtained by each student using the Comparable Interface.

Using Comparator Interface

- Multiple sorting cannot be done to sort and compare a collection of objects while working with a Comparable interface

For example, we can use the Comparator interface that is defined in the java.util package, if we wish to create a sorting logic based on the student's names and marks obtained. For this, the interface provides the compare() method that is used to compare two objects in a collection.

- The compare() method returns the value of int type with the following characteristics:
  - Negative: If the current object is less than the object being compared
  - Zero: If the current object is equal to the object being compared
  - Positive: If the current object is greater than the object being compared

Interactive Demo

Write a program in Java to store the names and marks of students. Next, sort the details based on the marks obtained by each student using the Comparator Interface.

Key Takeaways

- HashSet and TreeSet
- HashMap and TreeMap
- Comparable and Comparator