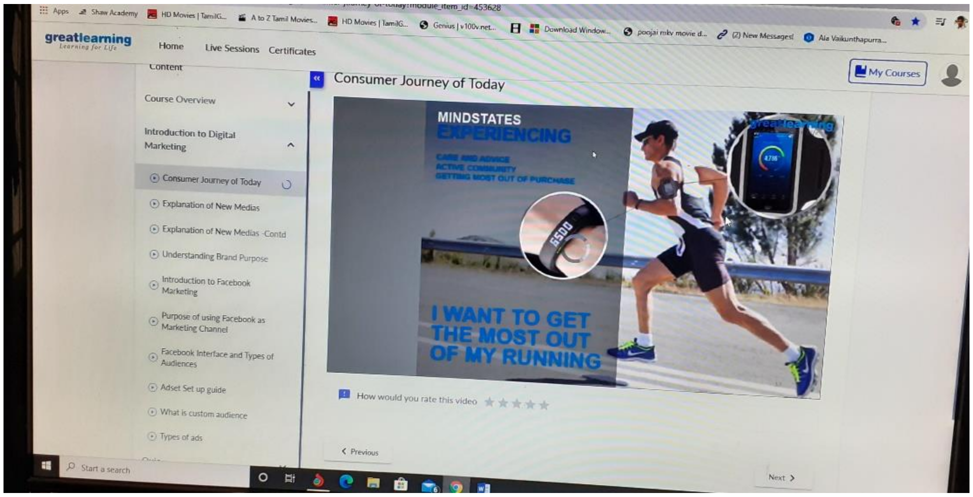
DAILY ASSESSMENT

Date:	**15-6-2020**	**Name:**	**Kavyashree m**
Course:	**Digital marketing**	**USN:**	**4al15ec036**
Topic:	**Introduction,Consumer journey of today, Explanation of new media, Understanding brand purpose, Introduction to facebook marketing, Purpose of facebook marketing, Addset set up guide, Custom audience, Types of advertisement**	**Semester & Section:**	**8th A**
Github Repository:	**kavya**		

FORENOON SESSION DETAILS


greatlearning
Learning for Life
Home
Live Sessions
Certificates
My Courses
Consumer Journey of Today
Course Overview
Introduction to Digital Marketing
Consumer Journey of Today
Explanation of New Medias
Explanation of New Medias -Contd
Understanding Brand Purpose
Introduction to Facebook Marketing
Purpose of using Facebook as Marketing Channel
Facebook Interface and Types of Audiences
Adset Set up guide
What is custom audience
Types of ads
MINDSTATES
EXPERIENCING
CARE AND ADVICE
ACTIVE COMMUNITY
GETTING MOST OUT OF PURCHASE
I WANT TO GET THE MOST OUT OF MY RUNNING
How would you rate this video
Previous
Next

DIGITAL MARKETING

Introduction to digital marketing

Digital Marketing is the term used for the targeted, measurable, and interactive marketing of products or services using digital technologies to reach the viewers, turn them into customers, and retain them. Digital marketing achieves targets of marketing a business through different online channels.

Consumer journey of today

A customer journey identifies all customer interactions with your business from the customer’s point of view.The digital customer journey combines all of the digital touchpoints a customer has with a brand and aggregates data collected such as: basic online consumer data, information about transactions, browsing history on all devices, and online customer service interaction.

Many businesses have used customer journey analytics to focus on the performance of specific touchpoints in an effort to improve customer experience. However, customer journey research by McKinsey and Co. has shown that focusing on the larger journey – interactions over time and multiple touchpoints – has a greater impact on business performance. Holistic journey performance improvement has a 30%-40% greater impact on customer satisfaction and a 20%-30% greater impact on value and business outcomes than touchpoint performance.

Explanation of new media

New media marketing centers on promoting brands and selling products and services through established and emerging online channels, harnessing these elements of new media to engage potential and current customers. New media marketing encompasses many different mediums, including display advertising, content marketing and social media promotions. The objective of all new media marketing is to get consumers to interact with the brand, engaging them in a way that increases awareness and correlates to sales.

Understanding brand purpose

Digital channels and assets are used to communicate a brand’s positioning (or purpose) as part of multichannel brand communication or engagement programmes".

So let’s rename ‘digital branding’ ‘digital communication’ and consider its role in the context of the business strategy and brand planning.

It can be difficult for companies to achieve a rewarding brand purpose on their own, especially if the product or service they offer doesn’t lend itself to a worthwhile cause. So, a lot of brands have found themselves partnering up with charities or campaigns. It’s important to note that a partnership with no common ground will become an obvious marketing move, when it comes to a charitable partnership ensure that they fit into your brand’s wider mission.For instance, Innocent recently partnered with Grow It Yourself, which is a charity that aims to teach children where their food comes from. Since Innocent’s brand is all about clean eating, fruit and vegetable to be specific, it’s a well calculated partnership.Alongside this charitable association, Innocent donate 10% of their profits to charity, use green electricity of their headquarters and use fruit from sustainable sources.

Introduction to facebook marketing

Facebook marketing refers to creating and actively using a Facebook page as a communications channel to maintain contact with and attract customers. Facebook actively provides for this, allowing users to create individual profiles or business pages for companies, organizations, or any group attempting to develop a fan base for a product, service, or brand.

Purpose of facebook marketing

1. Custom-targeted audience

Facebook advertising campaigns are built around (and for) specific audiences. For this reason, the Facebook Custom Audience Tool can work magic.

2. Maximum visibility at minimum cost

If you're planning to launch a business online, you want to build where the people are.

3. Early-stage brand awareness

Facebook can work a lot like PR if managed strategically. The Brand Awareness ad objective uses a combination of real-time proxy metrics with both the reach and attention users give to a campaign to provide brands with maximum exposure.

4. Facebook Live

Video is the undisputed champion of customer engagement and sharing. Facebook Live was introduced within the last year and has taken the internet by storm.

5. Chatbots

Ever since Mark Zuckerberg announced Facebook's 10-year roadmap, every business is trying to utilize Facebook Messenger to communicate with their customers.

Addset set up guide

1. Search engine optimization (SEO)
2. Search engine marketing (SEM)
3. Content marketing
4. Social Media Marketing (SMM)
5. Pay-per-click advertising (PPC)
6. Affiliate marketing
7. Email marketing

Custom audience

A Facebook Custom Audience is a targeted advertising service that allows businesses to import user email addresses for retargeting on the social media platform. Custom Audiences are an effective way for online businesses to interact with relevant users across multiple channels.

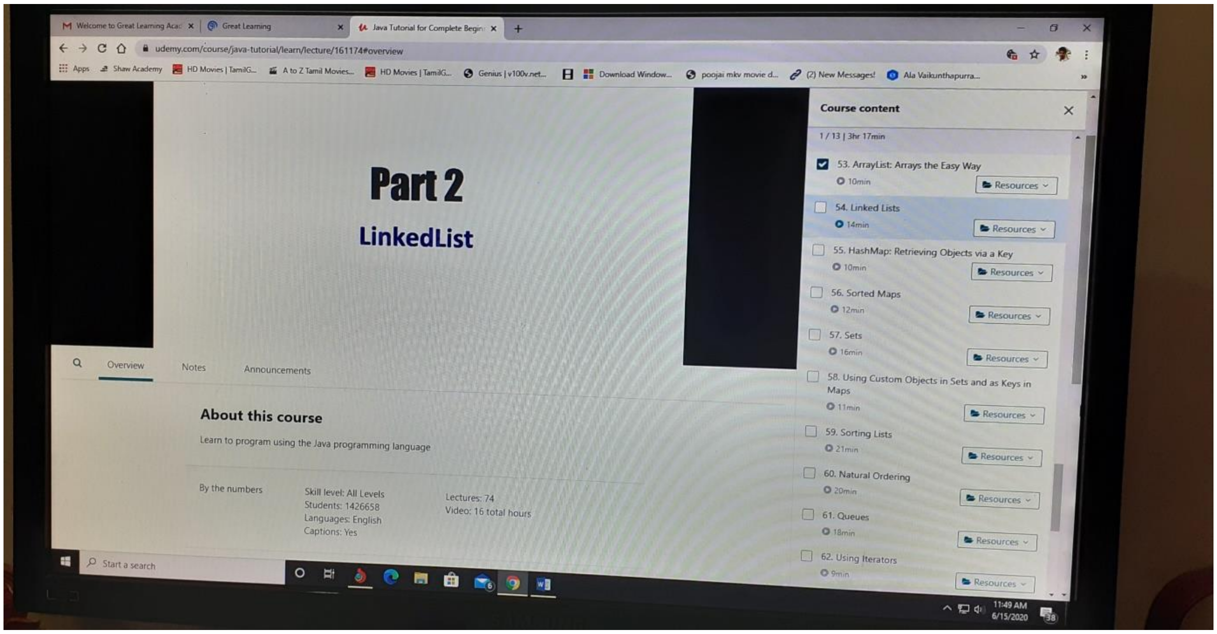
Types of advertisement

- Display ads. Display ads refer to visual advertising.
- Social media ads. The best platforms to target are Facebook, LinkedIn, and Twitter.
- Search engine marketing. Also referred to as SEM.
- Native advertising.
- Retargeting and remarketing.
- Video ads.
- Email marketing.

AFTERNOON SESSION DETAILS

Date:	15-6-2020	Name:	Kavyashree m
Course:	Java	USN:	4al15ec036
Topic:	Array list, linked list,hash maps,sorted maps,sets, Using Custom Objects in Sets and as Keys in maps,sorting lists	Semester & Section:	8th A
Github Repository:	kavya		

Image of session



Part 2

LinkedList

About this course

Learn to program using the Java programming language

By the numbers

Skill level: All Levels
Students: 1426658
Languages: English
Captions: Yes

Lectures: 74
Video: 16 total hours

Course content

1 / 13 | 3hr 17min

☑ 53. ArrayList: Arrays the Easy Way
10min Resources

□ 54. Linked Lists
14min Resources

□ 55. HashMap: Retrieving Objects via a Key
10min Resources

□ 56. Sorted Maps
12min Resources

□ 57. Sets
16min Resources

□ 58. Using Custom Objects in Sets and as Keys in Maps
11min Resources

□ 59. Sorting Lists
21min Resources

□ 60. Natural Ordering
20min Resources

□ 61. Queues
18min Resources

□ 62. Using Iterators
9min Resources

ArrayList

The ArrayList class is a resizable array, which can be found in the java.util package.

The difference between a built-in array and an ArrayList in Java, is that the size of an array cannot be modified (if you want to add or remove elements to/from an array, you have to create a new one). While elements can be added and removed from an ArrayList whenever you want. The syntax is also slightly different:

Example

Create an ArrayList object called cars that will store strings:

import java.util.ArrayList; // import the ArrayList class

ArrayList<String> cars = new ArrayList<String>(); // Create an ArrayList object

Linked list

Linked List are linear data structures where the elements are not stored in contiguous locations and every element is a separate object with a data part and address part. The elements are linked using pointers and addresses. Each element is known as a node. Due to the dynamicity and ease of insertions and deletions, they are preferred over the arrays. It also has few disadvantages like the nodes cannot be accessed directly instead we need to start from the head and follow through the link to reach to a node we wish to access. To store the elements in a linked list we use a doubly linked list which provides a linear data structure and also used to inherit an abstract class and implement list and deque interfaces.

In Java, LinkedList class implements the list interface. The LinkedList class also consists of various constructors and methods like other java collections.

Constructors for Java LinkedList:

1. LinkedList(): Used to create an empty linked list.
2. LinkedList(Collection C): Used to create a ordered list which contains all the elements of a specified collection, as returned by the collection's iterator.

Hash maps

HashMap is a Map based collection class that is used for storing Key & value pairs, it is denoted as HashMap<Key, Value> or HashMap<K, V>. This class makes no guarantees as to the order of the map. It is similar to the Hashtable class except that it is unsynchronized and permits nulls(null values and null key).

HashMap Example in Java:

In this example we have demonstrated almost all the important methods of HashMap class.

```
import java.util.HashMap;
import java.util.Map;
import java.util.Iterator;
import java.util.Set;
public class Details {

  public static void main(String args[]) {

    /* This is how to declare HashMap */
    HashMap<Integer, String> hmap = new HashMap<Integer, String>();

    /*Adding elements to HashMap*/
    hmap.put(12, "Chaitanya");
    hmap.put(2, "Rahul");
    hmap.put(7, "Singh");
    hmap.put(49, "Ajeet");
    hmap.put(3, "Anuj");

    /* Display content using Iterator*/
    Set set = hmap.entrySet();
    Iterator iterator = set.iterator();
    while(iterator.hasNext()) {
      Map.Entry mentry = (Map.Entry)iterator.next();
      System.out.print("key is: "+ mentry.getKey() + " & Value is: ");
      System.out.println(mentry.getValue());
    }

    /* Get values based on key*/
    String var= hmap.get(2);
    System.out.println("Value at index 2 is: "+var);

    /* Remove values based on key*/
    hmap.remove(3);
    System.out.println("Map key and values after removal:");
    Set set2 = hmap.entrySet();
    Iterator iterator2 = set2.iterator();
    while(iterator2.hasNext()) {
       Map.Entry mentry2 = (Map.Entry)iterator2.next();
       System.out.print("Key is: "+mentry2.getKey() + " & Value is: ");
       System.out.println(mentry2.getValue());
     }

  }
}
```

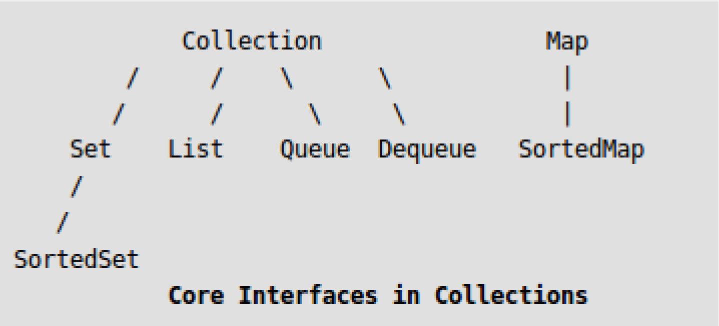
Output:

```
key is: 49 & Value is: Ajeet
key is: 2 & Value is: Rahul
key is: 3 & Value is: Anuj
key is: 7 & Value is: Singh
key is: 12 & Value is: Chaitanya
Value at index 2 is: Rahul
Map key and values after removal:
Key is: 49 & Value is: Ajeet
Key is: 2 & Value is: Rahul
Key is: 7 & Value is: Singh
Key is: 12 & Value is: Chaitanya
```

Sorted map

SortedMap Interface in Java with Examples

SortedMap is an interface in collection framework. This interface extends Map interface and provides a total ordering of its elements (elements can be traversed in sorted order of keys). Exampled class that implements this interface is TreeMap.



```
             Collection                Map
        /     /    \      \             |
       /      /     \      \            |
    Set     List    Queue  Dequeue   SortedMap
    /
   /
SortedSet
```

Core Interfaces in Collections

Code for SortedMap:

```
public interface SortedMap extends Map
{
   Comparator comparator();
   SortedMap subMap(K fromKey, K toKey);
   SortedMap headMap(K toKey);
   SortedMap tailMap(K fromKey);
   K firstKey();
   K lastKey();
```

```
}
filter_none
edit
play_arrow
brightness_4
// Java code to demonstrate SortedMap
import java.util.Iterator;
import java.util.Map;
import java.util.Set;
import java.util.SortedMap;
import java.util.TreeMap;

public class SortedMapExample
{
    public static void main(String[] args)
    {
        SortedMap<Integer, String> sm =
                    new TreeMap<Integer, String>();
        sm.put(new Integer(2), "practice");
        sm.put(new Integer(3), "quiz");
        sm.put(new Integer(5), "code");
        sm.put(new Integer(4), "contribute");
        sm.put(new Integer(1), "geeksforgeeks");
        Set s = sm.entrySet();

        // Using iterator in SortedMap
        Iterator i = s.iterator();
        // Traversing map. Note that the traversal
        // produced sorted (by keys) output .
        while (i.hasNext())
        {
            Map.Entry m = (Map.Entry)i.next();

            int key = (Integer)m.getKey();
            String value = (String)m.getValue();

            System.out.println("Key : " + key +
                            "  value : " + value);
        }
    }
}
```

Output:

```
Key : 1  value : geeksforgeeks
Key : 2  value : practice
Key : 3  value : quiz
Key : 4  value : contribute
Key : 5  value : code
```

Set in Java

- Set is an interface which extends Collection. It is an unordered collection of objects in which duplicate values cannot be stored.
- Basically, Set is implemented by HashSet, LinkedHashSet or TreeSet (sorted representation).

- Set has various methods to add, remove clear, size, etc to enhance the usage of this interface

filter_none

edit

play_arrow

brightness_4

```
// Java code for adding elements in Set
import java.util.*;
public class Set_example
{
    public static void main(String[] args)
    {
        // Set deonstration using HashSet
        Set<String> hash_Set = new HashSet<String>();
        hash_Set.add("Geeks");
        hash_Set.add("For");
        hash_Set.add("Geeks");
        hash_Set.add("Example");
        hash_Set.add("Set");
        System.out.print("Set output without the duplicates");

        System.out.println(hash_Set);

        // Set deonstration using TreeSet
        System.out.print("Sorted Set after passing into TreeSet");
        Set<String> tree_Set = new TreeSet<String>(hash_Set);
        System.out.println(tree_Set);
    }
}
```

Output:

```
Set output without the duplicates[Set, Example, Geeks, for]
Sorted Set after passing into TreeSet.
```

Using Custom Objects in Sets and as Keys in maps

To use your own objects as keys in Maps or in Sets you need to tell Java how to compare your objects, by implementing the hashCode() and equals() methods. Fortunately all decent IDEs will do this for you.

Sorting list

Sorting is one of the most common operations applied to lists and as such Java has built in mechanisms for doing it, like the Comparable and Comparator interfaces and the Collections.sort methods. These are great when you have a static list that needs to be ordered, but sometimes you want the list to remain sorted after some altering operations have been applied to it.

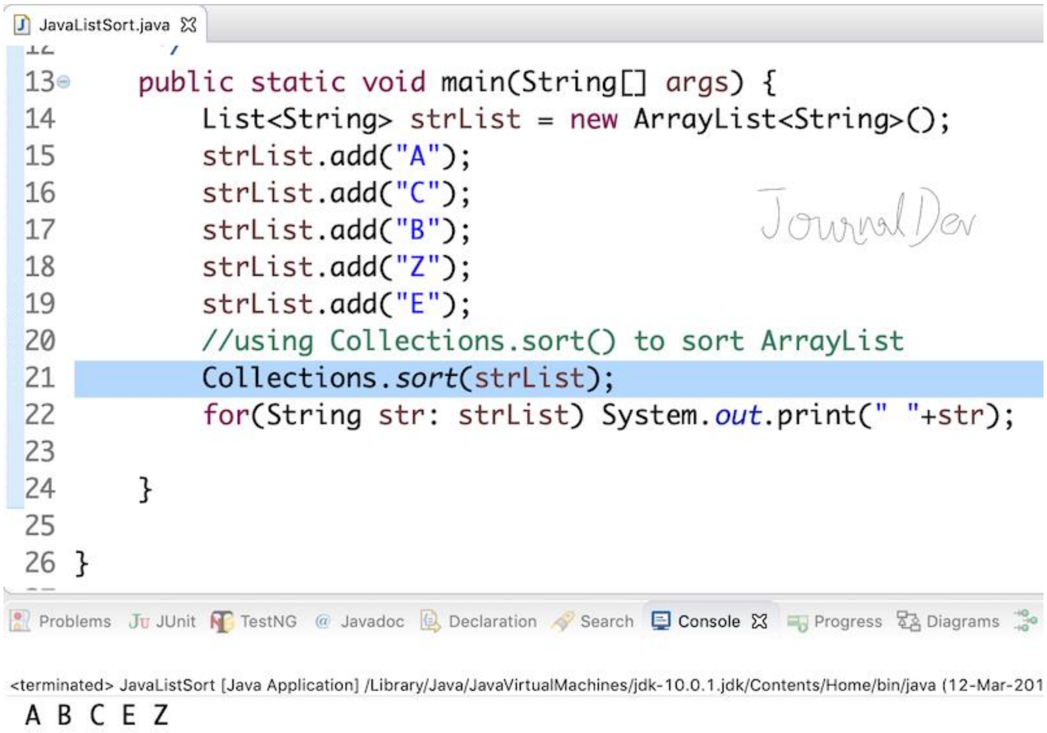
Here we will learn how to sort a list of Objects in Java. We can use Collections.sort() method to sort a list in the natural ascending order. All the elements in the list must implement Comparable interface, otherwise IllegalArgumentException is thrown.

Let’s look at a quick example to sort a list of strings.

```
package com.journaldev.sort;

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

public class JavaListSort {

    /**
     * This class shows how to sort ArrayList in java
     * @param args
     */
    public static void main(String[] args) {
        List<String> strList = new ArrayList<String>();
        strList.add("A");
        strList.add("C");
        strList.add("B");
        strList.add("Z");
        strList.add("E");
        //using Collections.sort() to sort ArrayList
        Collections.sort(strList);
        for(String str: strList) System.out.print(" "+str);
    }

}
```

As you can see that we are using Collections.sort() method to sort the list of Strings. The String class implements Comparable interface.

Output:



```
     */
    public static void main(String[] args) {
        List<String> strList = new ArrayList<String>();
        strList.add("A");
        strList.add("C");
        strList.add("B");
        strList.add("Z");
        strList.add("E");
        //using Collections.sort() to sort ArrayList
        Collections.sort(strList);
        for(String str: strList) System.out.print(" "+str);

    }

}
```

```
<terminated> JavaListSort [Java Application] /Library/Java/JavaVirtualMachines/jdk-10.0.1.jdk/Contents/Home/bin/java (12-Mar-201
 A B C E Z
```