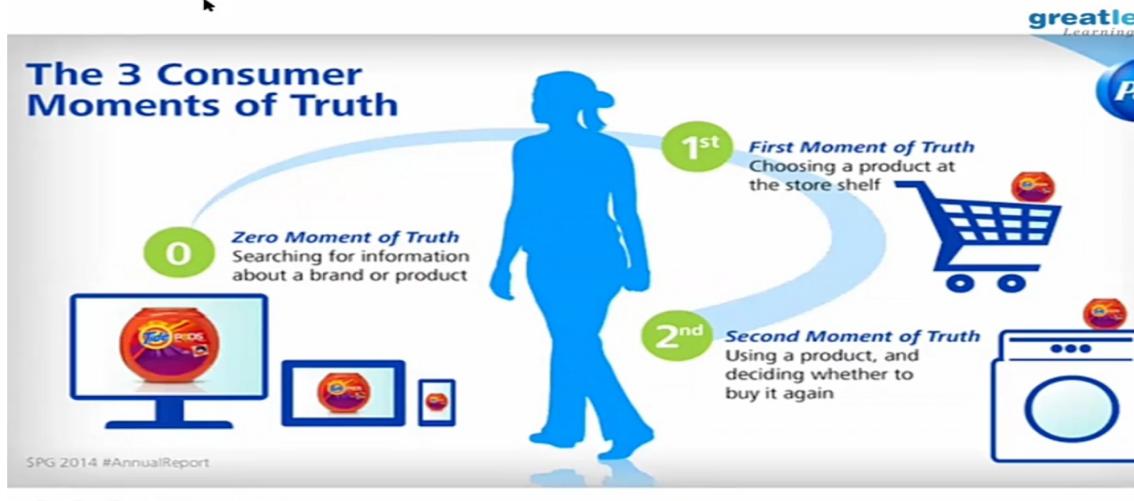
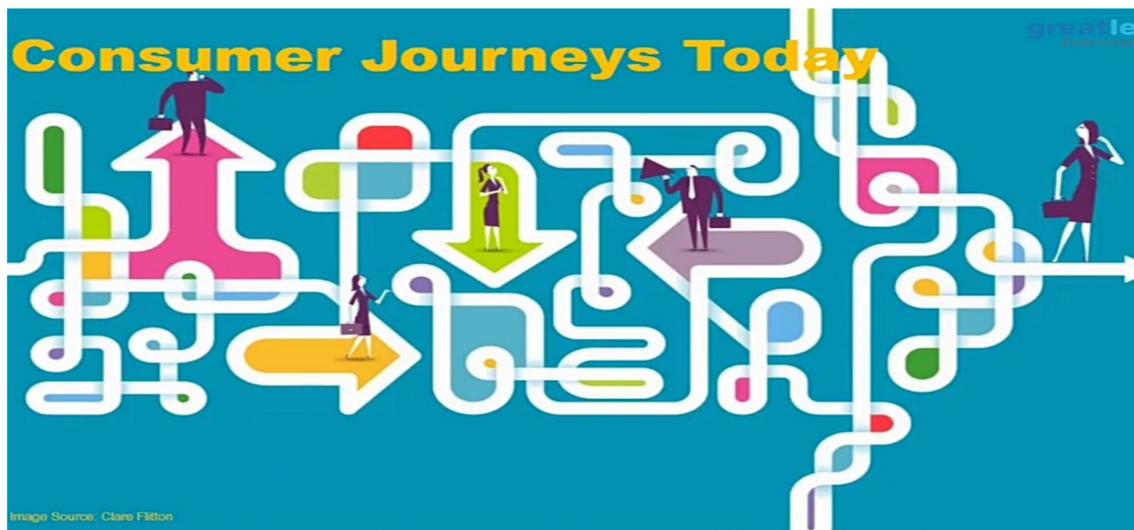


DAILY ASSESSMENT FORMAT

Date:	16/06/2020	Name:	Nichenametla Bhargavi
Course:	Digital Marketing	USN:	4AL17EC061
Topic:	Introduction and implementation	Semester & Section:	6th Sem A sec
Github Repository:	Bhargavi_Nichenametla		

FORENOON SESSION DETAILS

Image of session



Report – Report can be typed or hand written for up to two pages.

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.

The 3 consumer moments of truth:

- Zero moment of truth
- First moment of truth
- Second moment of truth

Understanding brand purpose:

- * Background
- * Demographer
- * Identifier
- * Goals
- * Challenges

Introduction to Facebook marketing

There are four steps to be followed as shown below:

- 1.Inventory
- 2.Signals
- 3.Predictions
- 4.Score

What is Custom audience?

It is a targeting option that matches people who visit your website with people on Facebook, using the Facebook pixel.

Business Objectives Delivered through Facebook



6

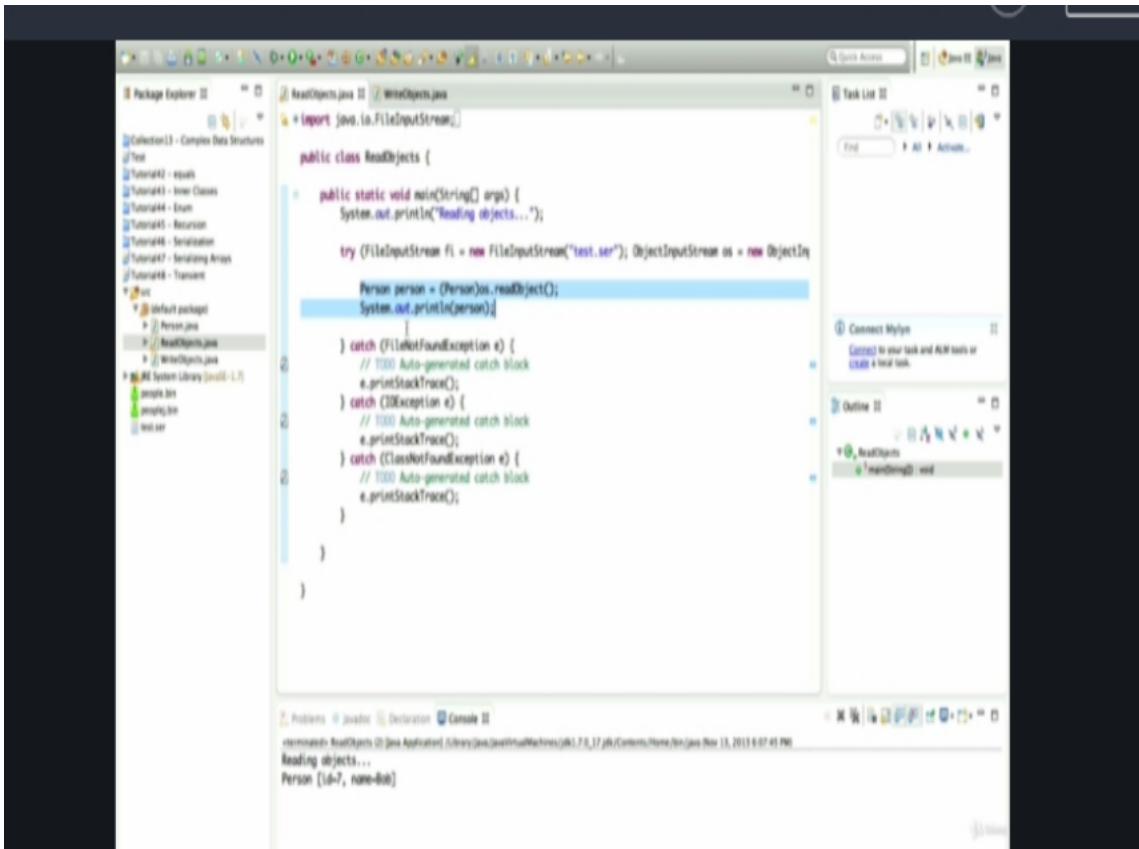
Certificate of Completion :



Date:	16/06/2020	Name:	Nichenametla Bhargavi
Course:	Udemy	USN:	4AL17EC061
Topic:	Programming in Core Java	Semester & Section:	6th Sem A sec
Github Repository:	Bhargavi_Nichenametla		

AFTERNOON SESSION DETAILS

image of the session



The screenshot shows an IDE interface with the following details:

- Project Structure:** The left sidebar shows a package named "Tutorial43 - Complex Data Structures" containing several tutorials and a "src" folder. Inside "src", there are two Java files: "Person.java" and "ReadObjects.java".
- Code Editor:** The main editor window displays the "ReadObjects.java" code. The code imports `java.io.ObjectInputStream` and defines a `main` method that reads objects from a file named "test.ser" and prints them to the console.
- Output Window:** The bottom window shows the execution results. It starts with the message "Reading objects...", followed by "Person [id=7, name=Bob]".
- Toolbars and Menus:** Standard Eclipse-style toolbars and menus are visible at the top and bottom of the interface.

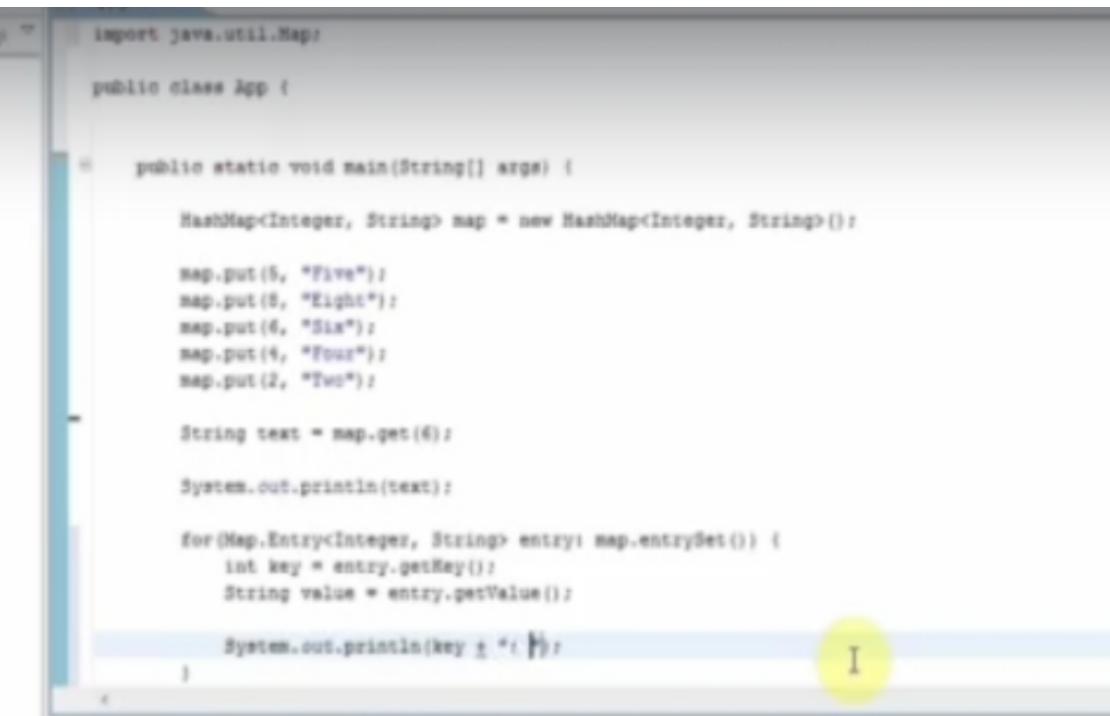
Report – Report can be typed or hand written for up to two pages.

The Java collections framework:

1. Different types of "ArrayList: Arrays the Easy Way" are discussed.
2. Came across the topic "Linked Lists" and learnt clearly.
3. Learnt about "HashMap: Retrieving Objects via a Key".
4. Also learnt about the Sorting of Maps.
5. Sets and its types are also discussed
6. Using Custom Objects in Sets and as Keys in Maps are discussed.
7. Sorting of Lists and other topics also discussed.

Examples

HashMap: Retrieving Objects via a Key:



```
import java.util.Map;

public class App {

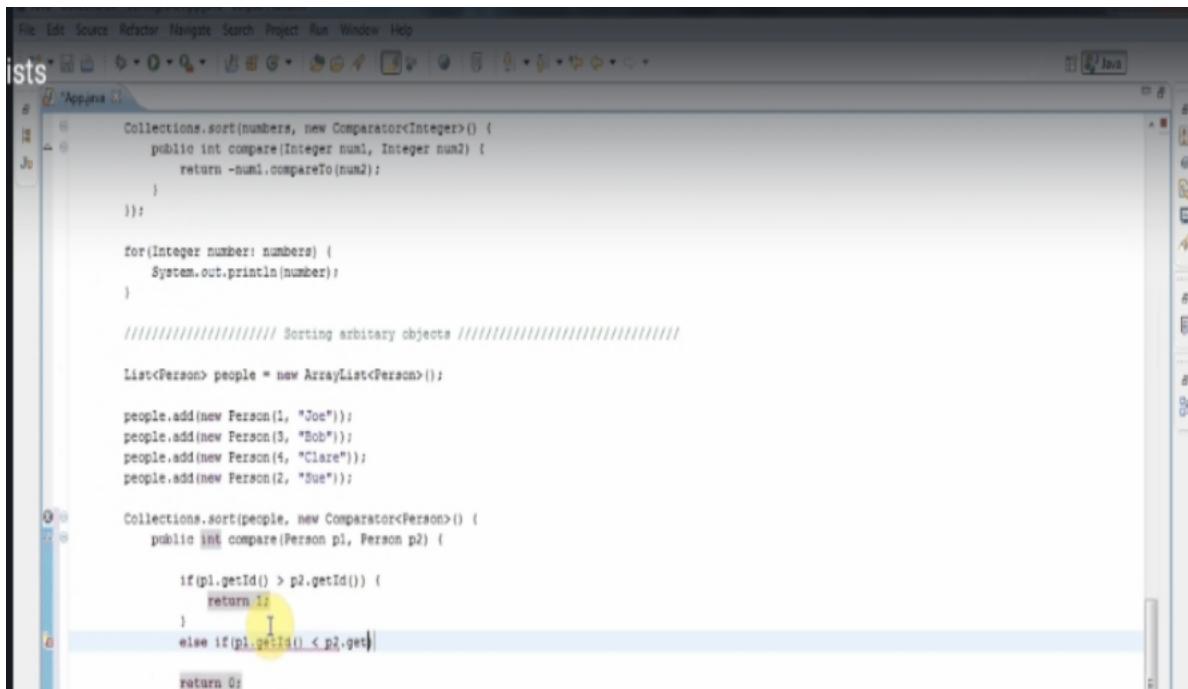
    public static void main(String[] args) {
        Map<Integer, String> map = new HashMap<Integer, String>();
        map.put(5, "Five");
        map.put(8, "Eight");
        map.put(6, "Six");
        map.put(4, "Four");
        map.put(2, "Two");

        String text = map.get(6);
        System.out.println(text);

        for(Map.Entry<Integer, String> entry: map.entrySet()) {
            int key = entry.getKey();
            String value = entry.getValue();

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

Sorting Lists:



A screenshot of an IDE showing Java code for sorting lists. The code uses the Collections.sort() method with a Comparator. It first sorts integers, then arbitrary objects (Person class). The Person comparator compares IDs.

```
File Edit Source Refactor Navigate Search Project Run Window Help
lists App.java
public class App {
    public static void main(String[] args) {
        List<Integer> numbers = new ArrayList<Integer>();
        numbers.add(3);
        numbers.add(1);
        numbers.add(4);
        numbers.add(2);

        Collections.sort(numbers, new Comparator<Integer>() {
            public int compare(Integer num1, Integer num2) {
                return -num1.compareTo(num2);
            }
        });

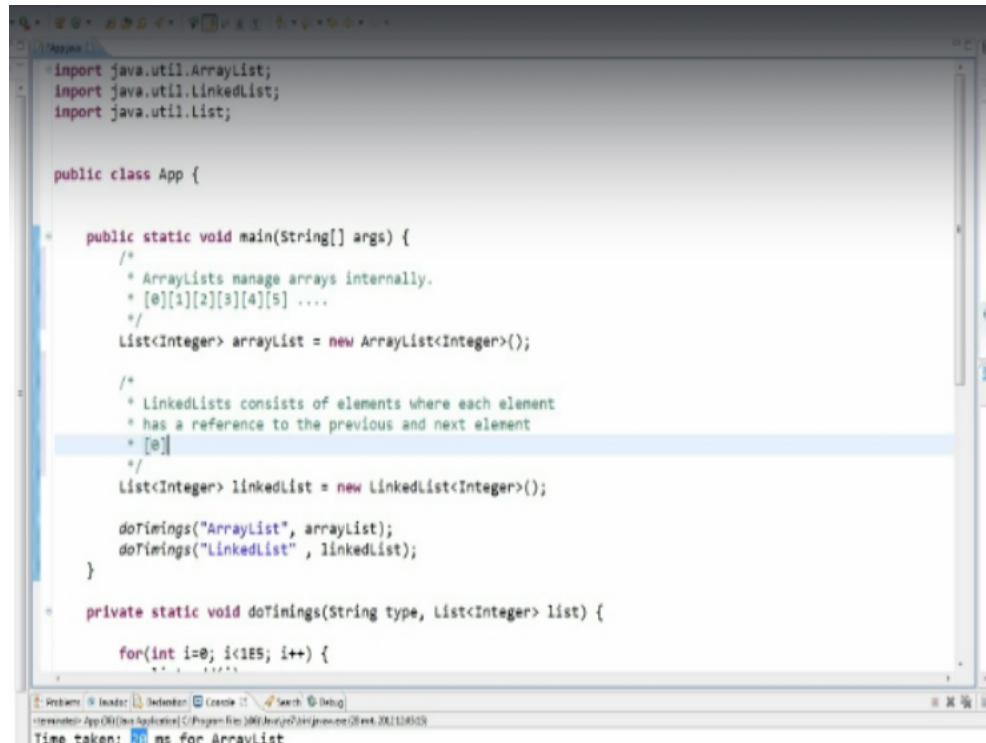
        for(Integer number: numbers) {
            System.out.println(number);
        }

        //////////////////// Sorting arbitrary objects ///////////////////
        List<Person> people = new ArrayList<Person>();

        people.add(new Person(1, "Joe"));
        people.add(new Person(3, "Bob"));
        people.add(new Person(4, "Clare"));
        people.add(new Person(2, "Sue"));

        Collections.sort(people, new Comparator<Person>() {
            public int compare(Person p1, Person p2) {
                if(p1.getId() > p2.getId()) {
                    return 1;
                } else if(p1.getId() < p2.getId()) {
                    return -1;
                }
                return 0;
            }
        });
    }
}
```

Linked Lists:



A screenshot of an IDE showing Java code for linked lists. The code imports ArrayList, LinkedList, and List. It creates an ArrayList and a LinkedList, both containing integers. A private static method doTimings() is used to time the creation of both lists.

```
import java.util.ArrayList;
import java.util.LinkedList;
import java.util.List;

public class App {

    public static void main(String[] args) {
        /*
         * ArrayLists manage arrays internally.
         * [0][1][2][3][4][5] ...
         */
        List<Integer> arrayList = new ArrayList<Integer>();

        /*
         * LinkedLists consists of elements where each element
         * has a reference to the previous and next element
         * [e]
         */
        List<Integer> linkedList = new LinkedList<Integer>();

        doTimings("ArrayList", arrayList);
        doTimings("LinkedList", linkedList);
    }

    private static void doTimings(String type, List<Integer> list) {
        for(int i=0; i<100; i++) {
        }
    }
}
```

Date:	16/06/2020	Name:	Nichenametla Bhargavi
Course:	Webinar from Cambridge University	USN:	4AL17EC061
Topic:	Business Etiquette	Semester & Section:	6th Sem A sec
Github Repository:	Bhargavi_Nichenametla		

Image of the Session

Cambridge Assessment English RECONNECT Cambridge BEC English Qualifications

Workplace etiquette

Don't get into fights/arguments

pic:www.google.com

Talk softly

Non Verbal communication: Body Language

- Leaning forward = interest
- Smiling = friendly
- Nodding = attentive and alert
- Eye contact = curious and focused
- Crossed arms = defensive
- Fidgeting hands or tapping feet = nervous or bored
- Lack of eye contact = untrustworthy
- Leaning back= discomfort

Verbal Communication

If it transpires that there is a surfeit of applications for this post, we may have cause to invite assistance from another department.

- Keep it simple

If too many people apply, we may need help.

- Asking for clarification

Excuse me, I don't understand. Could you repeat that please using different words?

Could you put that more simply please?

Can I just clarify that please? Do you mean that you want me to ?