

Group A

1. Given the following Java program:

```
import java.util.*;
public class Main
{
    public static void main(String[] args) {
        List num = new ArrayList(Arrays.asList(23, 16, 14, 33, 19, 6, 1));
        System.out.println("List is "+num);
    }
}
```

(a) Give the index values of all the odd numbers assuming zero-based indexing

= The index values of the odd numbers in the list are:

- Index (0) =23
- Index (3) =33
- Index (4) =19
- Index (6) =1

(b) How many elements would be looked at when the list is traversed (from start to finish) until the value 19 was found?

= Four elements would be looked at when the list is traversed (from start to finish) until the value is 19 was found.

2. Which of the following lists are syntactically correct in Java?

Try them out in to see if you were correct.

(a) List num = new ArrayList(Arrays.asList(1, 2, 3, 'four'));

(b) List num = new ArrayList(Arrays.asList(1, 2, [3, 4]));

= Both lists are syntactically incorrect in Java. In list 'a', the character 'four' should be surrounded by quotation marks to be a string, and in the list 'b', the square brackets are not a valid syntax in java.

```
import java.util.ArrayList;
public class ArrayLista{
    public static void main(String[] args){
        List num = new ArrayList(Arrays.asList(1, 2, 3, 'four'));
        System.out.println("List is " + num);
    }
}
```

```
import java.util.ArrayList;
public class ArrayListb{
    public static void main(String[] args){
        List num = new ArrayList(Arrays.asList(1, 2, [3, 4]));
        System.out.println("List is " + num);
    }
}
```

Right answer:

```
import java.util.*;
public class Arraylist{
    public static void main(String[] args){
        List num = new ArrayList (Arrays.asList(1, 2, 3, 4));
        System.out.println("List is " + num);
    }
}
```

Output:

List is [1, 2, 3, 4]

3. Perform a series of list operations on the following list:

List fruit = new ArrayList (Arrays.asList('apple', 'banana', 'pear', 'cherry')); to produce this updated list:

['Grapefruit', 'banana', 'Date', 'cherry', 'Orange']

```
import java.util.ArrayList;
public class ListOperations{
    public static void main(String[] args){
        ArrayList<String> fruit1 = new ArrayList<String>();
        fruit1.add(0, "apple");
        fruit1.add(1, "banana");
        fruit1.add(2, "pear");
        fruit1.add(3, "cherry");
        //create a list with the initial values
        ArrayList<String> fruit2= new ArrayList<String>();
        // add a new element at the beginning of the list
        fruit2.add(0, "Grapefruit");
        //replace the second element in the list with a new value
        fruit2.add(1, "banana");
        fruit2.add(2, "date");
        fruit2.add(3, "cherry");
        //add a new
        fruit2.add(4, "orange");
        System.out.println(fruit2);
    }
}
```

Output:

```
[Grapefruit, banana, date, cherry, orange]
```

1. Write a program to find out whether a given integer is present in an array or not.

```
import java.util.Scanner;
public class arrayinteger{
    public static void main(String[] args){
        Scanner abc = new Scanner(System.in);
        int[] array = {1, 2, 3, 4, 5};
        System.out.print("enter tan integer to search for: ");
        int num = abc.nextInt();//integer to search for
        boolean found = false; //flag to indicate if the integer was found
        //search for the integer in the array
        for(int i=0; i<array.length; ++i){
            if(array[i] == num){
                found= true;
                break;
            }
        }
        if (found){
            System.out.println("the integer was found in the array");
        }
        else{
            System.out.println("the integer was not found in the array");
        }
    }
}
```

Output:

```
enter tan integer to search for: 6
the integer was not found in the array
```

2. Calculate the average marks from an array containing marks of all students in physics using a for-each loop.

```
import java.util.*;
public class physicsmarks{
    public static void main(String[] args){
        int i;
        Scanner abc = new Scanner(System.in);
        System.out.print("enter number of students");
        int n = abc.nextInt();
        int [] a = new int [n];
        double average = 0;
        System.out.println("enter the marks");
        for (i=0;i<n;i++){
            a[i]=abc.nextInt();
        }
        for (i=0;i<n;i++){
            average=average + a[i];
        }
        System.out.print("average marks of physics is: ");
        for (i=0;i<n-1;i++){
            System.out.println(a[i]+", ");
        }
        System.out.println(a[i]+"=" +average/n);
    }
}
```

Output:

```
enter number of students8
enter the marks
56
67
76
86
96
33
46
21
average marks of physics is: 56,
67,
76,
86,
96,
33,
46,
21=60.125
```

3. Write a Java program to reverse an array.

```
import java.util.*;
public class reversearray{
    public static void main(String[] args){
        int[] array={1,2,3,4,5};
        System.out.println("real array: ");
        for (int i=0;i<array.length;i++){
            System.out.print(array[i]+" ");
        }
        System.out.println();
        System.out.println("array in reverse: ");
        for(int i=array.length-1; i>=0;i--){
            System.out.print(array[i]+" ");
        }
    }
}
```

Output: real array:

1 2 3 4 5

array in reverse:

5 4 3 2 1

4. Write a Java program to find the maximum element in an array.

```
import java.util.*;
public class maxielement{
    public static void main(String[] args){
        int[] array={1,2,3,4,5,6,8};
        int max= array[0];
        Scanner nao=new Scanner(System.in);
        System.out.print("enter number of elements in an array");
        max=nao.nextInt();
        int a[] = new int[max];
        System.out.println("enter elements of an array");
        for (int i=1; i < array.length; i++){
            if(array[i]>max){
                max=array[i];
            }
        }
        System.out.println("the maximum number of element in an array is: "+max);
    }
}
```

Output:

```
enter number of elements in an array4
enter elements of an array
the maximum number of element in an array is: 4
the maximum number of element in an array is: 4
the maximum number of element in an array is: 4
the maximum number of element in an array is: 5
the maximum number of element in an array is: 6
the maximum number of element in an array is: 8
```

5. Write a Java program to find whether an array is sorted or not.


```
import java.util.*;
public class sortedornot{
    public static void main(String[] args){
        List fruits=new ArrayList();
        fruits.add("banana");
        fruits.add("watermelon");
        fruits.add("kiwi");
        fruits.add("strawberry");
        System.out.println("unsorted list of fruits: "+fruits);
        Collections.sort(fruits);
        System.out.println("Sorted List of fruits: "+fruits);
    }
}
```

Output:

```
unsorted list of fruits: [banana, watermelon, kiwi, strawberry]
Sorted List of fruits: [banana, kiwi, strawberry, watermelon]
```

Group C

1. Write a Java program to append the specified element to the end of a hash set.

```
import java.util.HashSet;
public class endofhashset{
    public static void main(String[] args){
        HashSet set=new HashSet();
        //adding elements to the hash set
        set.add("kiwi");
        set.add("tangerine");
        set.add("banana");
        //displaying the initial set
        System.out.println("Initial Set: "+set);
        //adding element to the set
        set.add("pineapple");
        //displaying the modified set
        System.out.println("modified set: "+set);
    }
}
```

Week 4 Workshop Naomi Thing

```
count the number of key-value mappings: 5
```

0u00000000000000

Output:

```
Initial Set: [banana, kiwi, tangerine]
modified set: [banana, kiwi, tangerine, pineapple]
```

2. Write a Java program to compare two sets and retain elements which are same on both sets.

```
import java.util.HashSet;
public class compare2sets{
    public static void main(String[] args){
        //create 2 sets
        HashSet<String> set1= new HashSet<String>();
        HashSet<String> set2= new HashSet<String>();
        //add elements to set1
        set1.add("social");
        set1.add("science");
        set1.add("math");
        //add elements to set2
        set2.add("arts");
        set2.add("lgbtqia+");
        set2.add("science");
        //create a new set to store the common elements
        HashSet<String> common= new HashSet<>();
        //iterate through set1 and check if the element is present in set2
        for (String element:set1){
            if(set2.contains(element)){
                common.add(element);
            }
        }
        System.out.println("common elements: "+common);
    }
}
```

Output:

```
common elements: [science]
```

3. Write a Java program to count the number of key-value mappings in a hash table

```
import java.util.Hashtable;
public class HashTable{
    public static void main(String[] args){
        //create a hash table
        Hashtable<String, Integer> table=new Hashtable<>();
        //add key-value pairs to the hash table
        table.put("A",1);
        table.put("B",2);
        table.put("C",3);
        table.put("D",4);
        table.put("E",5);
        //count the number of key-value mappings in the hash table
        int count = table.size();
        System.out.println("count the number of key-value mappings: "+ table.size());
    }
}
```

count the number of key-value mappings: 5

4. Write a Java program to get a collection view of the values contained in this map

```
import java.util.*;
public class values{
    public static void main(String[] args){
        //create a map
        Map<String, Integer> map = new HashMap<>();
        map.put("A", 1);
        map.put("B", 2);
        map.put("C", 3);
        map.put("D", 4);
        map.put("E", 5);
        //get a collective view of the values contained in the map
        Collection<Integer> values = map.values();
        System.out.println("values in the map: "+values);
    }
}
```

Output:

```
values in the map: [1, 2, 3, 4, 5]
```

(Optional)

Group D

1. Building a Rock Paper Scissor game in java

Ask the user to enter in their move.

Make a list of valid moves.

Check if the user entered a valid move by looking at the list of valid moves. (If the move is in the list, it is valid move)

Randomly generate the opponent's move. (Randomly choose one move from the list of valid moves)

Display the result to user

Use a loop to continue asking the user for their move.

Check if the user wants to quit.

Week 4 Workshop Naomi Thing

```
import java.util.Scanner;
public class Rockpapersci{
    public static void main(String[] args){
        Scanner abc = new Scanner(System.in);
        int playerScore=0;
        int computerScore=0;
        while(true){
            System.out.println("enter rock, paper, scissor, or quit");
            String playerMove=abc.nextLine().toLowerCase();
            if (playerMove.equals("quit")){
                break;
            }
            if(!playerMove.equals("rock")&& !playerMove.equals("paper")&&!playerMove.equals("scissor")){
                System.out.println("Invalid input. Try again.");
                continue;
            }
            //generate the computer's move
            int random =(int) (Math.random() *3);
            String computerMove;
            if(random==0){
                computerMove="rock";
            } else if (random ==1){
                computerMove="paper";
            } else{
                computerMove="scissor";
            }
            System.out.println("Computer's Move: " +computerMove);
            if (playerMove.equals(computerMove)){
                System.out.println("DRAW!!!!");
            }else if (playerMove.equals("rock")&&computerMove.equals("paper")||
                playerMove.equals("paper")&&computerMove.equals("scissor")||
                playerMove.equals("scissor")&&computerMove.equals("rock")){
                System.out.println("YOU WIN!!!");
                playerScore++;
            }else{
                System.out.println("YOU LOSE..");
                computerScore++;
            }
            System.out.println("current score: YOU" +playerScore+"-"+computerScore+"Computer");
            System.out.println();
        }
    }
}
```

Output:

enter rock, paper, scissor, or quit

rock

Computer's Move: paper

YOU WIN!!!

current score: YOU1-0Computer

enter rock, paper, scissor, or quit

Week 4 Workshop Naomi
Thing