JAVA ASSIGNMENT 1

Name:- Umme Kulsum

Roll No.:- 25

Div:- A

---

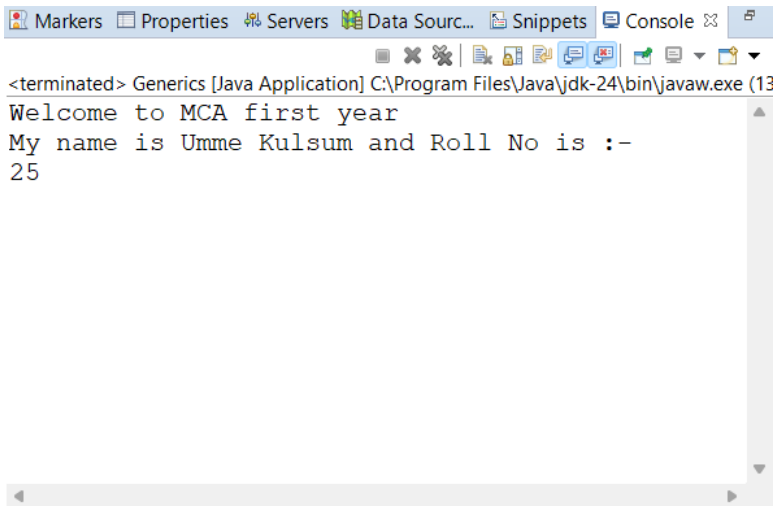
1) Assignment on Java Generics:-

1.1 Write a java program to demonstrate the Generic Class.

CODE:-

```
package Generics;

class Geg<T>{
T obj;
Geg(T obj) { this.obj = obj; }
public T get() { return this.obj;}
}

public class Generics {
     public static void main(String[] args) {
          Geg <String> s = new Geg<String>("Welcome to MCA
first year");
          System.out.println(s.get());
          Geg <String> S =new Geg<String>("My name is Umme
Kulsum and Roll No is :-");
          System.out.println(S.get());
          Geg <Integer> i = new Geg<Integer>(25);
          System.out.println(i.get());;
     }
}
```

OUTPUT

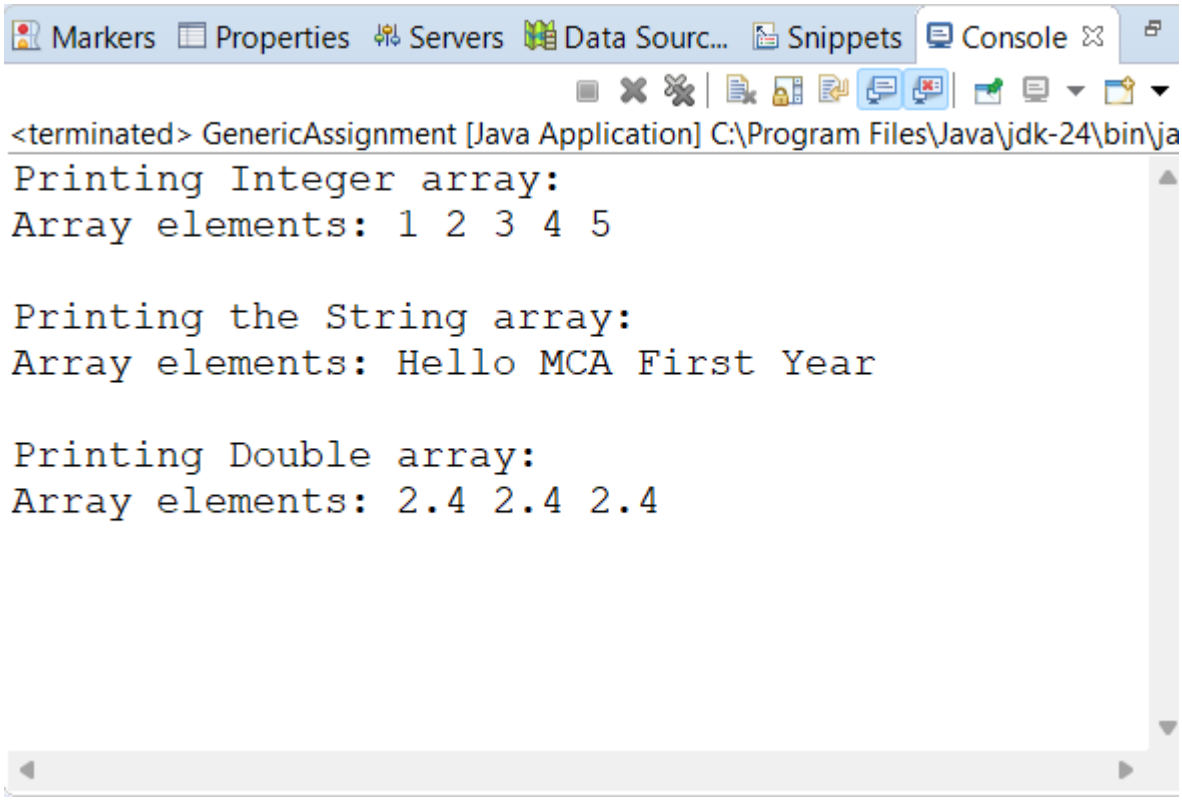


1.2Write a java program to demonstrate Generic Methods.

CODE:

```
package GenericAssignment;

public class GenericAssignment {
     // Generic method definition
    public static <T> void printArray(T[] array) {
        System.out.print("Array elements: ");
        for (T element : array) {
            System.out.print(element + " ");
        }
        System.out.println();
    }

    public static void main(String[] args) {

        Integer[] intArray = {1, 2, 3, 4, 5};
        System.out.println("Printing Integer array:");
        printArray(intArray);

        String[] stringArray = {"Hello", "MCA", "First",
"Year"};
        System.out.println("\nPrinting the String array:");
        printArray(stringArray);

        Double[] doubleArray = {2.4, 2.4, 2.4};
        System.out.println("\nPrinting Double array:");
        printArray(doubleArray);
    }
}
```

OUTPUT:

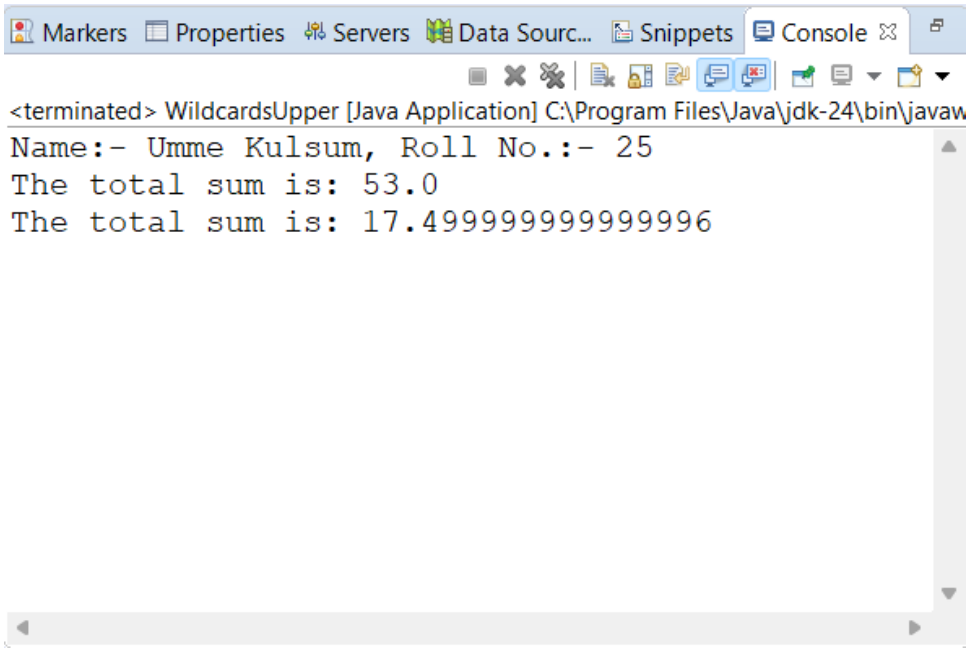


1.3Write a java program to demonstrate Wildcards in Java Generics.

(a) Upper bounded wildcards:-

```
package WildcardsUpper;
import java.util.*;

public class WildcardsUpper {
     public static void main(String[] args) {
         System.out.println("Name:- Umme Kulsum, Roll No.:-
25");

         List<Integer> l1 = Arrays.asList(2, 8, 9, 6, 7, 8, 9,
4);
         System.out.println("The total sum is: " + sum(l1));

         List<Double> l2 = Arrays.asList(4.1, 4.3, 6.7, 2.4);
         System.out.println("The total sum is: " + sum(l2));
     }

     private static double sum(List<? extends Number> list) {
         double sum = 0.0;
         for (Number num : list) {
             sum += num.doubleValue();
         }
         return sum;
     }
}
```

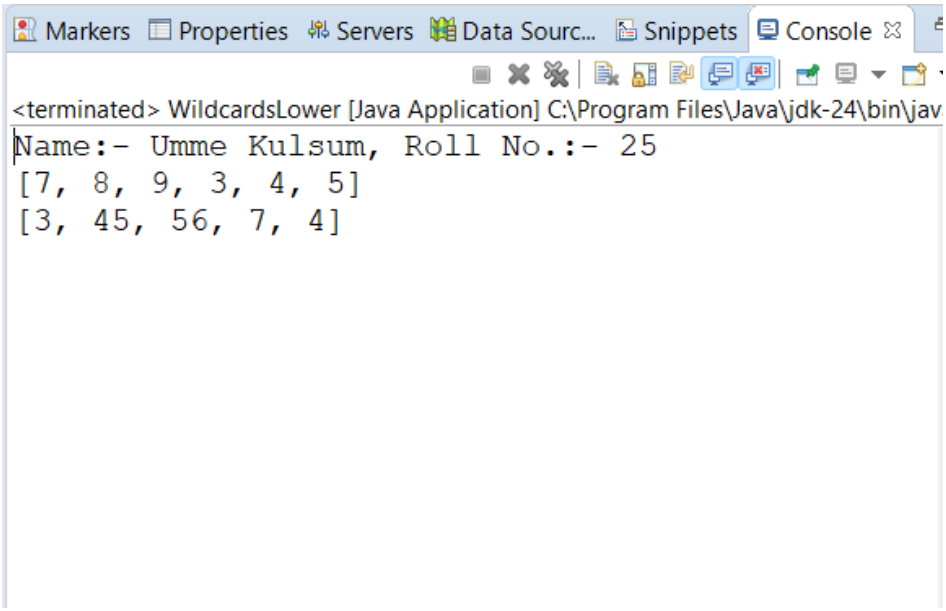
OUTPUT:



(b) Lower bounded wildcards:-

```
package WildcardsLower;
import java.util.*;

public class WildcardsLower {
     public static void main(String[] args) {
         System.out.println("Name:- Umme Kulsum, Roll No.:-
25");
         List<Integer> list1 = Arrays.asList(7, 8, 9, 3, 4, 5);
         print1(list1);
         List<Number> list2 = Arrays.asList(3, 45, 56, 7, 4);
         print1(list2);
     }

    public static void print1(List<? super Integer> list) {
         System.out.println(list);
     }

}
```

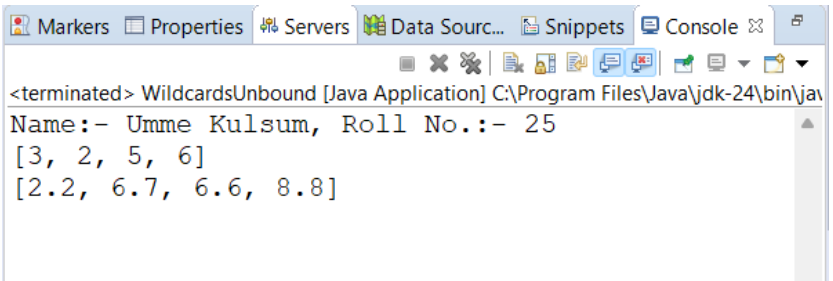
OUTPUT:



(c) Unbounded wildcards:-

```
package WildcardsUnbound;
import java.util.*;

public class WildcardsUnbound {
     public static void main(String[] args) {
         System.out.println("Name:- Umme Kulsum, Roll No.:-
25");
         List<Integer> list1 = Arrays.asList(3, 2, 5, 6);
         List<Double> list2 = Arrays.asList(2.2, 6.7, 6.6,
8.8);
         printlist(list1);
         printlist(list2);
     }
    private static void printlist(List<?> list) {
         System.out.println(list);
     }
}
```

OUTPUT:



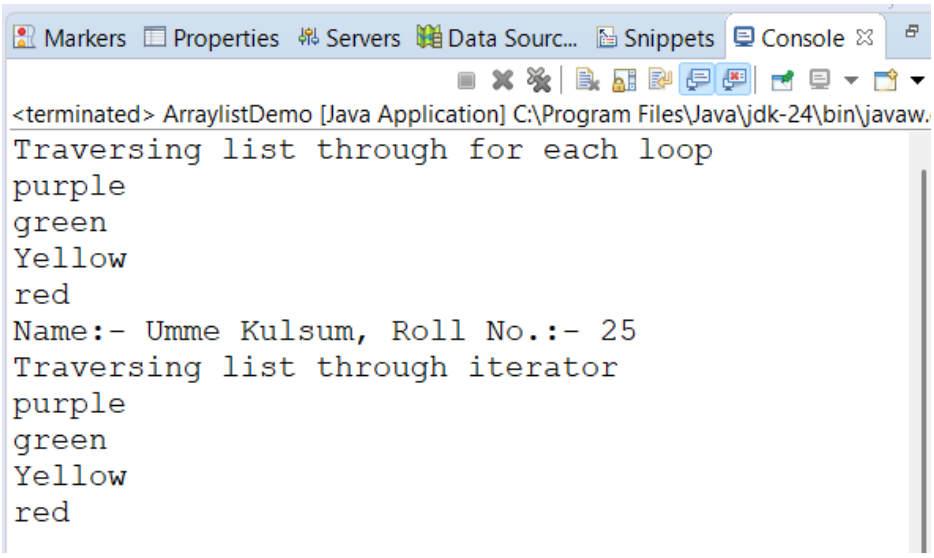
2) Assignment on List Interface:-

(2.1) Write a Java program to create List containing list of items of type String and use for-each loop to print the items of the list.

CODE:

```
package ArraylistDemo;
import java.util.*;

import java.util.ArrayList;
import java.util.Iterator;

public class ArraylistDemo {
     public static void main(String args[]) {
        int a;
        ArrayList<String> list = new ArrayList<String>();
        list.add("purple");
        list.add("green");
        list.add("Yellow");
        list.add("red");
        System.out.println(list);
        System.out.println("Traversing list through for each
loop");
        for (String color : list)
            System.out.println(color);
        System.out.println("Name:- Umme Kulsum, Roll No.:-
25");
        System.out.println("Traversing list through
iterator");
        Iterator itr = list.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

OUTPUT:

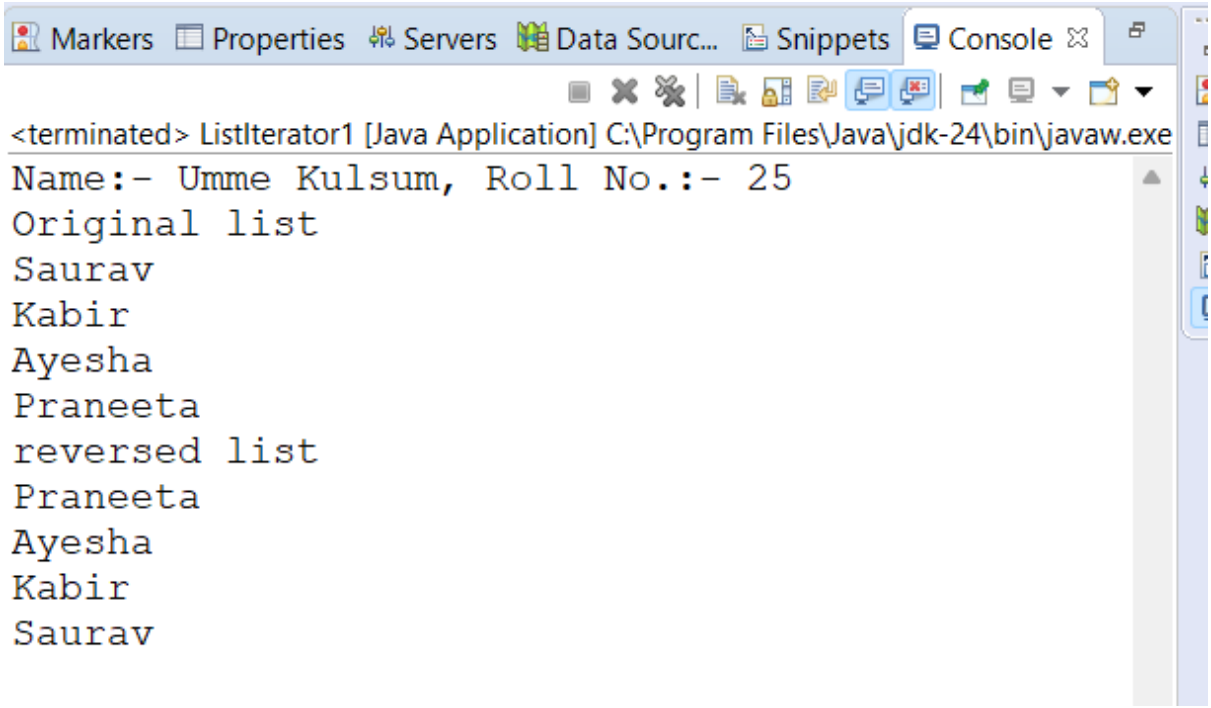


2.2 Write a Java program to create List containing list of items and use ListIterator interface to print items present in the list. Also print the list in reverse/backward direction

Code:

```
package ListIterator1;
import java.util.*;

public class ListIterator1 {
     public static void main(String args[]) {
         List<String> mylist = new ArrayList<String>();
         mylist.add("Saurav");
         mylist.add("Kabir");
         mylist.add("Ayesha");
         mylist.add("Praneeta");
         System.out.println("Name:- Umme Kulsum, Roll No.:-
25");
         System.out.println("Original list");
         Iterator<String> itr = mylist.iterator();
         while (itr.hasNext()) {
             System.out.println(itr.next());
         }
         Collections.reverse(mylist);
         System.out.println("reversed list");
         Iterator<String> itr1 = mylist.iterator();
         while (itr1.hasNext()) {
             System.out.println(itr1.next());
         }
     }
}
```

OUTPUT:



3) Assignment on SET Interface:-

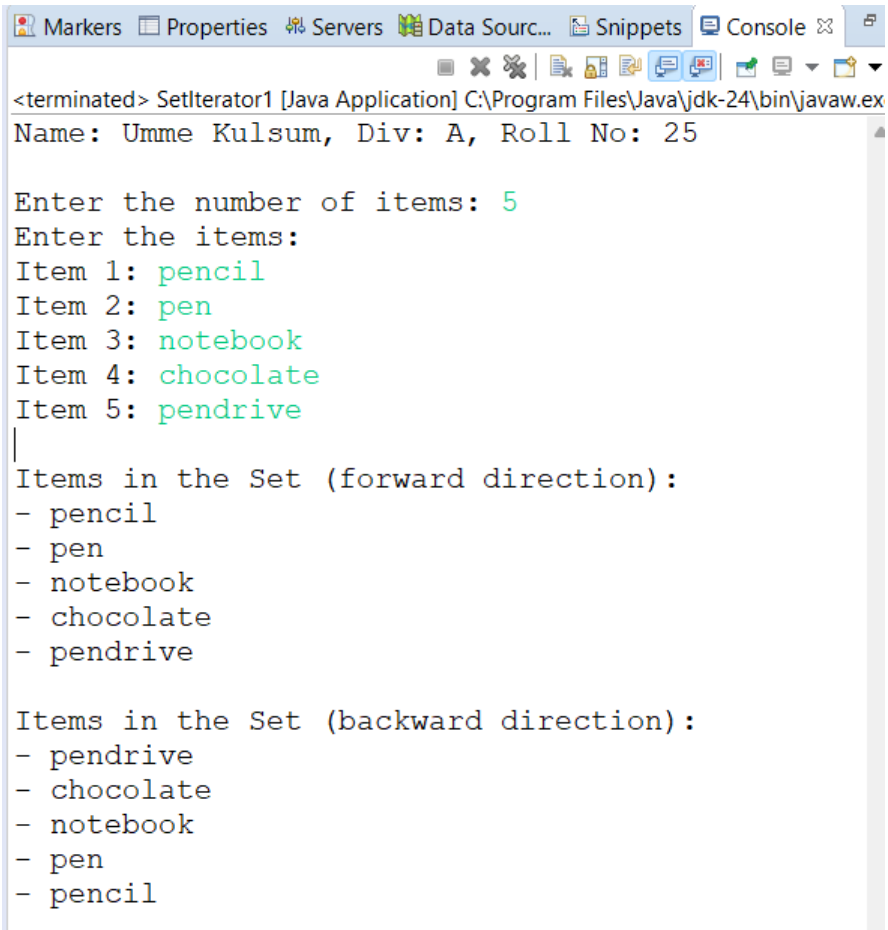
(3.1) Write a Java program to create a Set containing list of items of type String and print the items in the list using Iterator interface. Also print the list in reverse/backward direction.

CODE:-

```
package SetIterator1;
import java.util.*;

public class SetIterator1 {
    public static void main(String[] args) {
        System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");
        Scanner sc = new Scanner(System.in);
        Set<String> itemSet = new LinkedHashSet<>();
        System.out.print("Enter the number of items: ");
        int n = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter the items:");
        for (int i = 0; i < n; i++) {
            System.out.print("Item " + (i + 1) + ": ");
            String item = sc.nextLine();
            itemSet.add(item);
        }
```

```
        LinkedList<String> itemList = new
LinkedList<>(itemSet);
        ListIterator<String> iterator =
itemList.listIterator();
        System.out.println("\nItems in the Set (forward
direction):");
        while (iterator.hasNext()) {
            System.out.println("- " + iterator.next());
        }
        System.out.println("\nItems in the Set (backward
direction):");
        while (iterator.hasPrevious()) {
            System.out.println("- " + iterator.previous());
        }
        sc.close();
    }
}
```

OUTPUT:



(3.2) Write a Java program using Set interface containing list of items and Perform the following operations:

a. Add items in the set.

b. Insert items of one set in to other set.

c. Remove items from the set

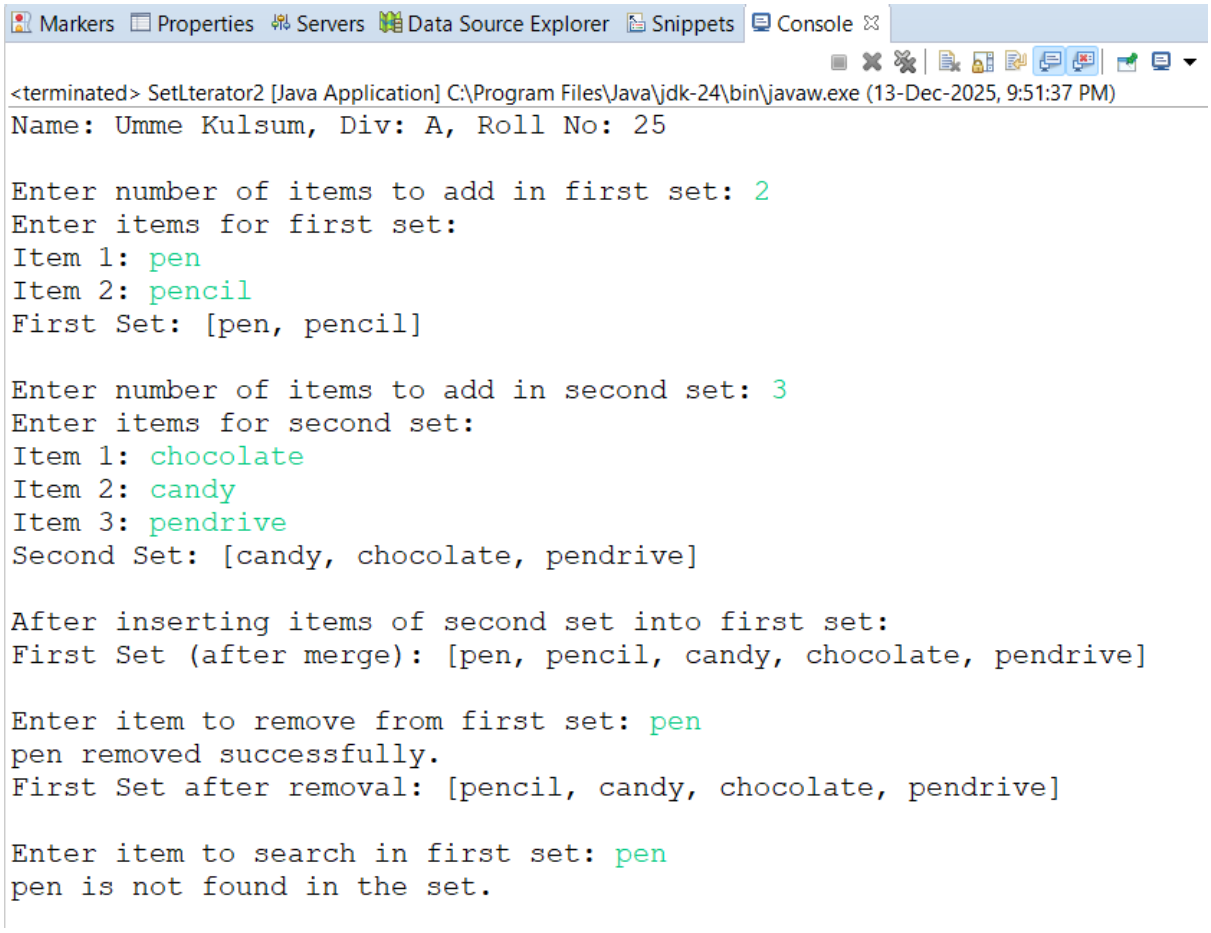
d. Search the specified item in the set

CODE:

```
package SetLterator2;
import java.util.*;

public class SetLterator2 {
     public static void main(String[] args) {
        System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");

        Set<String> set1 = new LinkedHashSet<>();
        System.out.print("Enter number of items to add in
first set: ");
        Scanner sc = new Scanner(System.in);
        int n1 = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter items for first set:");
        for (int i = 0; i < n1; i++) {
            System.out.print("Item " + (i + 1) + ": ");
            set1.add(sc.nextLine());
        }
        System.out.println("First Set: " + set1);

        Set<String> set2 = new TreeSet<>();
        System.out.print("\nEnter number of items to add in
second set: ");
        int n2 = sc.nextInt();
        sc.nextLine();
        System.out.println("Enter items for second set:");
        for (int i = 0; i < n2; i++) {
            System.out.print("Item " + (i + 1) + ": ");
            set2.add(sc.nextLine());
        }
        System.out.println("Second Set: " + set2);

        set1.addAll(set2);
        System.out.println("\nAfter inserting items of second
set into first set:");
        System.out.println("First Set (after merge): " +
set1);
```

```
        System.out.print("\nEnter item to remove from first
set: ");
        String removeItem = sc.nextLine();
        if (set1.remove(removeItem)) {
            System.out.println(removeItem + " removed
successfully.");
        } else {
            System.out.println(removeItem + " not found in the
set.");
        }
        System.out.println("First Set after removal: " +
set1);

        System.out.print("\nEnter item to search in first set:
");
        String searchItem = sc.nextLine();
        if (set1.contains(searchItem)) {
            System.out.println(searchItem + " is present in
the set.");
        } else {
            System.out.println(searchItem + " is not found in
the set.");
        }

        sc.close();
    }

}
```

OUTPUT:



4) Assignment on MAP Interface:

(4.1) Write a Java program using Map interface containing list of items having keys and associated values and perform the following operations:

a. Add items in the map.

b. Remove items from the map

c. Search specific key from the map

d. Get value of the specified key

e. Insert map elements of one map in to other map.

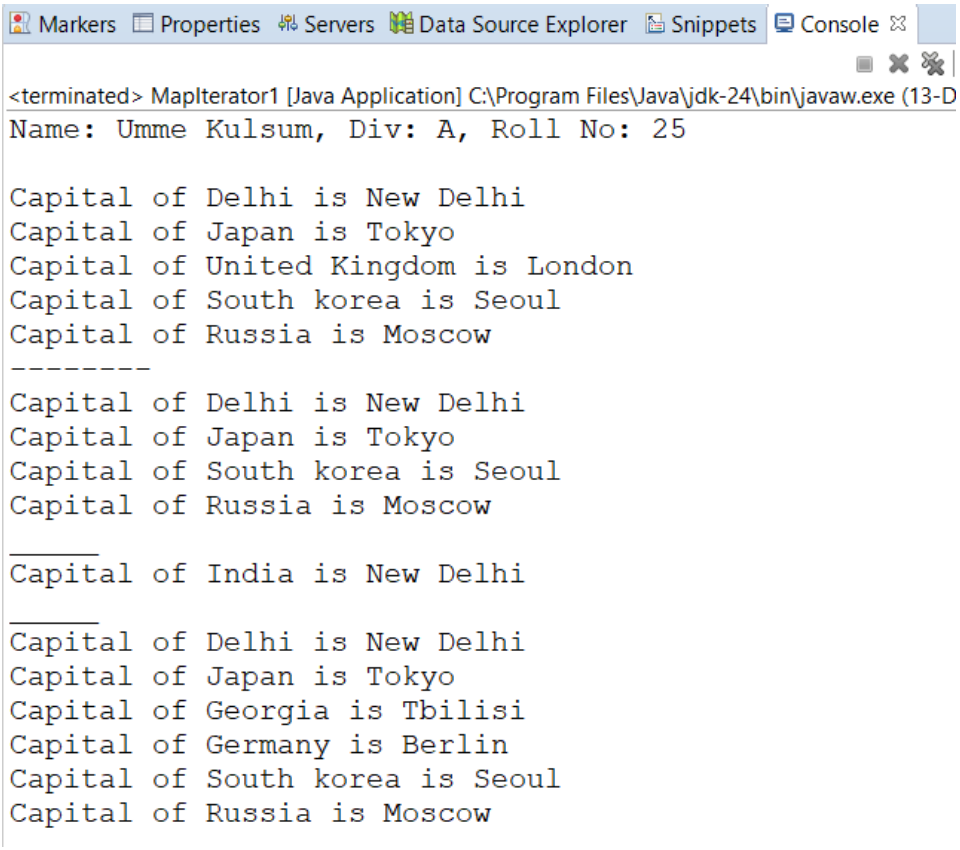
f. Print all keys and values of the map.

CODE:

```
package MapIterator1;
import java.util.*;

public class MapIterator1 {
    public static void main(String args[]) {
        System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");
```

```
        Map<String,String> hmap = new HashMap<>();
        hmap.put("Delhi", "New Delhi");
        hmap.put("South korea", "Seoul");
        hmap.put("Japan", "Tokyo");
        hmap.put("Russia", "Moscow");
        hmap.put("United Kingdom", "London");

        for (Map.Entry<String, String> m : hmap.entrySet()) {
            System.out.println("Capital of " + m.getKey() + "
is " + m.getValue());
        }

        System.out.println("--------");
        hmap.remove("United Kingdom");

        for (Map.Entry<String, String> m : hmap.entrySet()) {
            System.out.println("Capital of " + m.getKey() + "
is " + m.getValue());
        }

        System.out.println("_____");
        System.out.println("Capital of India is " +
hmap.get("Delhi"));
        System.out.println("_____");

        Map<String, String> hmap2 = new HashMap<>();
        hmap2.put("Germany", "Berlin");
        hmap2.put("Georgia", "Tbilisi");

        hmap.putAll(hmap2);

        for (Map.Entry<String, String> m : hmap.entrySet()) {
            System.out.println("Capital of " + m.getKey() + "
is " + m.getValue());
        }
    }
}
```

OUTPUT:



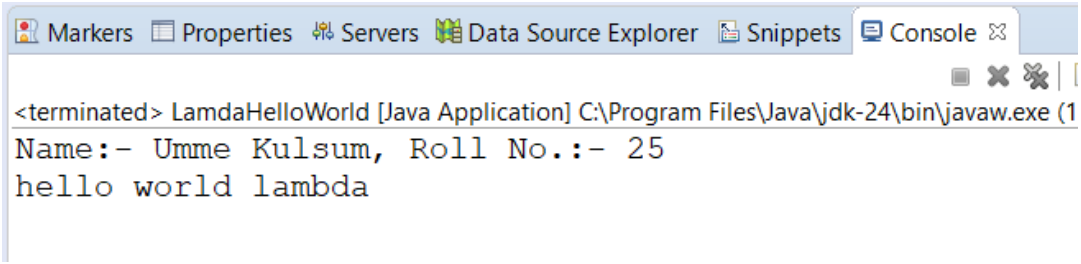
(5) LAMBDA EXPRESSIONS

(5.1) Write a Java program using Lambda Expression to print ”Hello World”.

CODE :-

```
package LamdaHelloWorld;

interface hello1 {
    void hi();
}

public class LamdaHelloWorld {
     public static void main(String[] args) {
        System.out.println("Name:- Umme Kulsum, Roll No.:-
25");
        hello1 h1 = () -> {
            System.out.print("hello world lambda");
        };
        h1.hi();
    }
}
```

OUTPUT:

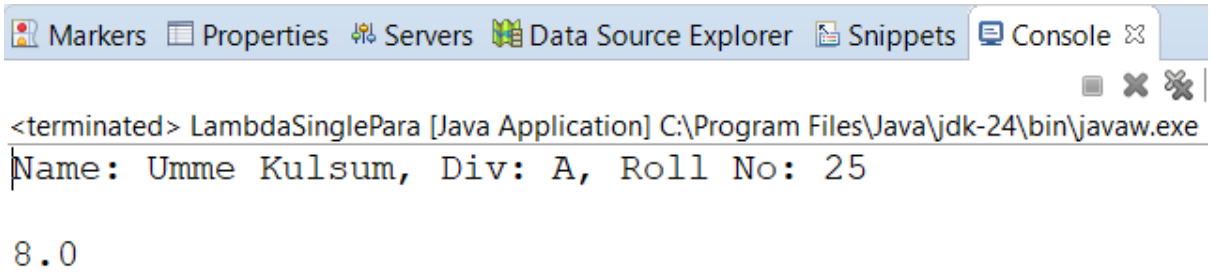


(5.2) Write a Java program using Lambda Expression with single parameters.

CODE:-

```
package LambdaSinglePara;

interface cube {
    double cubevolume(double L);
}

public class LambdaSinglePara {
     public static void main(String[] args) {
         System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");
         cube c2 = (L) -> {
             return L * L * L;
         };
         System.out.print(c2.cubevolume(2));
    }
}
```

OUTPUT:



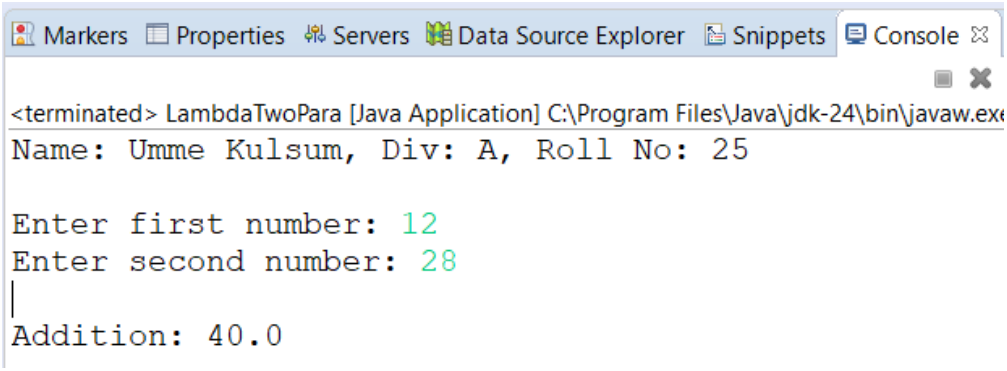
(5.3) Write a Java program using Lambda Expression with multiple parameters to add two numbers.

CODE:

```
package LambdaTwoPara;
import java.util.*;

interface Lambda {
    double addNum(double a, double b);
}

public class LambdaTwoPara {
     public static void main(String[] args) {
```

```
        System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first number: ");
        double num1 = sc.nextDouble();
        System.out.print("Enter second number: ");
        double num2 = sc.nextDouble();

        Lambda m = (a, b) -> a + b;
        System.out.println("\nAddition: " + m.addNum(num1,
num2));
        sc.close();
    }
}
```

OUTPUT:



5.4 Write a Java program using Lambda Expression to calculate the following:

a. Convert Fahrenheit to Celcius

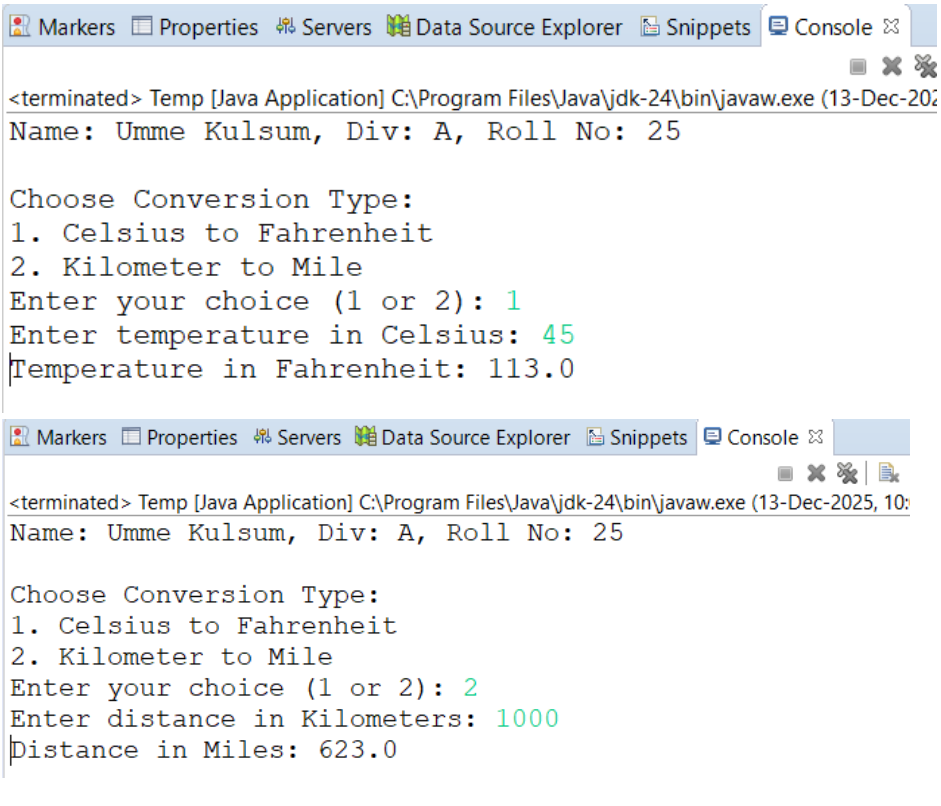
b. Convert Kilometers to Miles.

CODE:-

```
package ASSIGNMENT;
import java.util.Scanner;
interface TemperatureConverter {
   double convert(double celsius);
}
interface DistanceConverter {
   double convert(double kilometers);
}
public class Temp {
   public static void main(String[] args) {
      System.out.println("Name: Ritesh Shinde, Div: B, Roll No: 54\n");
      Scanner sc = new Scanner(System.in);

      TemperatureConverter tempConverter = (celsius) -> (celsius * 1.8) + 32;
```

```
        DistanceConverter distanceConverter = (kilometers) -> kilometers * 0.623;

        System.out.println("Choose Conversion Type:");
        System.out.println("1. Celsius to Fahrenheit");
        System.out.println("2. Kilometer to Mile");
        System.out.print("Enter your choice (1 or 2): ");

        int choice = sc.nextInt();

        if (choice == 1) {
            System.out.print("Enter temperature in Celsius: ");
            double celsius = sc.nextDouble();
            System.out.println("Temperature in Fahrenheit: " + tempConverter.convert(celsius));
        } else if (choice == 2) {
            System.out.print("Enter distance in Kilometers: ");
            double kilometers = sc.nextDouble();
            System.out.println("Distance in Miles: " + distanceConverter.convert(kilometers));
        } else {
            System.out.println("Invalid choice! Please select 1 or 2.");
        }
        sc.close();
    }
}
```

OUTPUT:

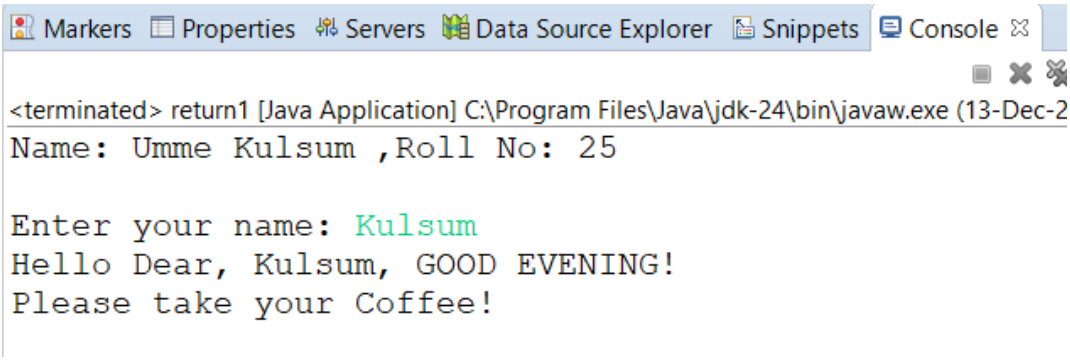


(5.5) Write a Java program using Lambda Expression with or without return keyword.

CODE :-

```
package return1;

import java.util.Scanner;

interface GreetingMessage {
   String showMessage(String name);
}

interface MorningGreeting {
   void greet();
}

public class return1 {
   public static void main(String[] args) {
      System.out.println("Name: Umme Kulsum ,Roll No: 25\n");
      Scanner sc = new Scanner(System.in);

      GreetingMessage personalizedGreeting = (name) -> {
         return "Hello Dear, " + name + ", GOOD EVENING!";
      };

      MorningGreeting morningGreeting = () -> {
         System.out.println("Please take your Coffee!");
      };

      System.out.print("Enter your name: ");
      String userName = sc.nextLine();
      System.out.println(personalizedGreeting.showMessage(userName));
      morningGreeting.greet();

      sc.close();
   }
}
```

OUTPUT:



5.6 Write a Java program using Lambda Expression to concatenate two strings.

CODE :-

```
package strConcat;

import java.util.*;

interface ConcatStr {
    public String concat(String s1, String s2);
}

public class StrConcat {
    public static void main(String[] args) {
        System.out.println("Name: Umme Kulsum, Div: A, Roll
No: 25\n");
        ConcatStr cs = (String s1, String s2) -> s1 + s2;
        System.out.println("Concatenating \"hello,\" and \"
World\" = " + cs.concat(" Hello,", "World"));
    }
}
```

OUTPUT: