JAVA ASSIGNMENT 1

Name:- Susmit Mahesh Naik

Roll No.:- 35

Div:- A2

---

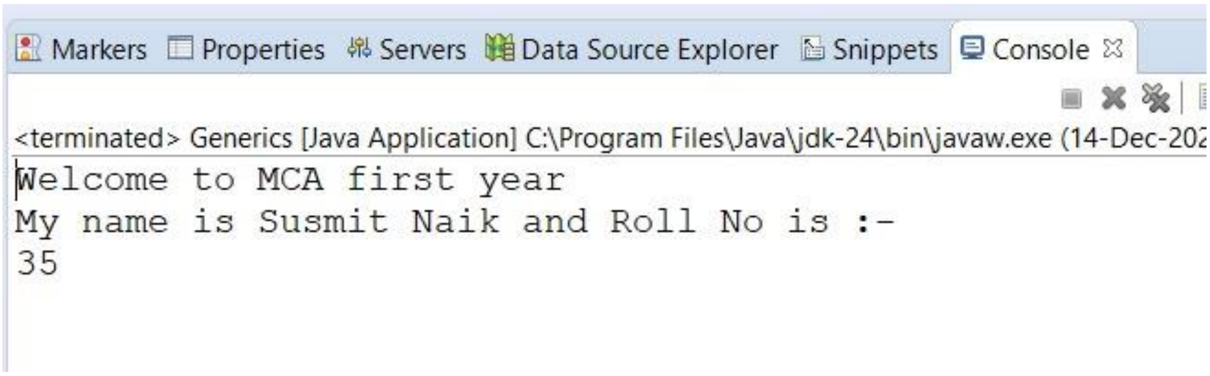
1) Assignment on Java Generics:-

1.1 Write a java program to demonstrate the Generic Class.

CODE:-

```
class Geg<T>{
T obj;
Geg(T obj) { this.obj = obj; }
public T get() { return this.obj;}
}
public class GenericsExample {
	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 Susmit Naik and Roll No
is :-");
		System.out.println(S.get());
		Geg <Integer> i = new Geg<Integer>(35);
		System.out.println(i.get());;
	    }

}
```

OUTPUT



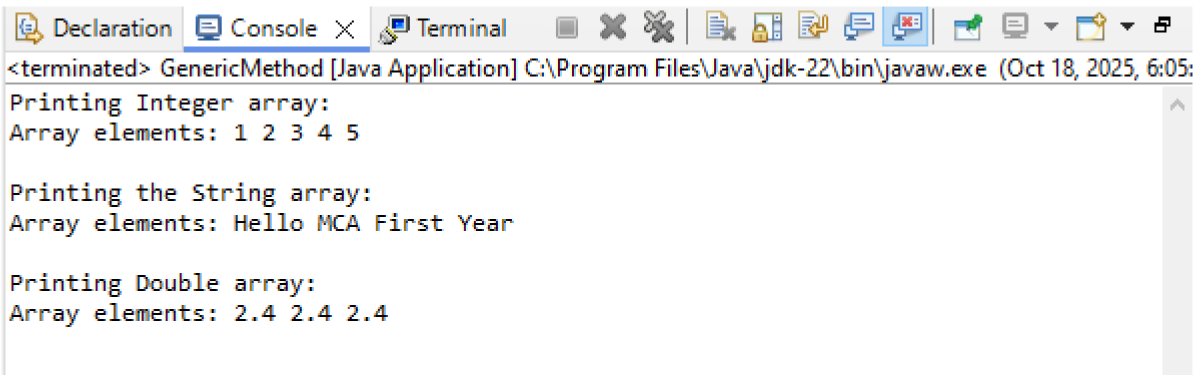
```
<terminated> Generics [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (14-Dec-202
Welcome to MCA first year
My name is Susmit Naik and Roll No is :-
35
```

1.2Write a java program to demonstrate Generic Methods.

CODE:

```
package ASSIGNMENT;

public class GenericMethod {

   // 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:



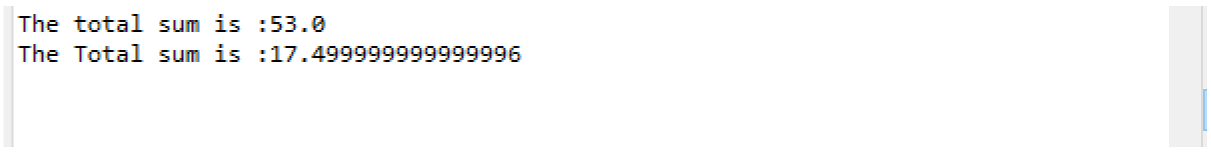
```
<terminated> GenericMethod [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (Oct 18, 2025, 6:05:
Printing Integer array:
Array elements: 1 2 3 4 5

Printing the String array:
Array elements: Hello MCA First Year

Printing Double array:
Array elements: 2.4 2.4 2.4
```

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

(a) Upper bounded wildcards:-

```java
package ASSIGNMENT;

import java.util.*;

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

        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:

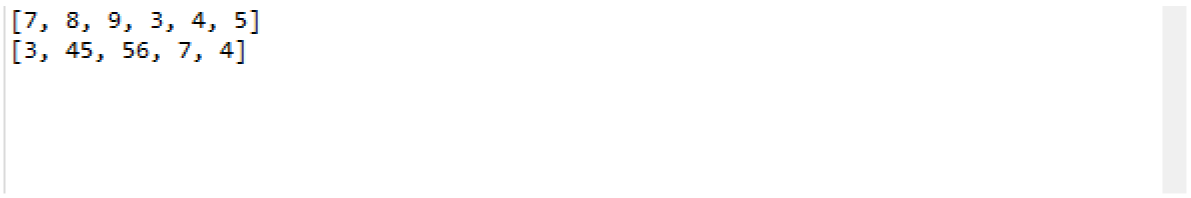


```
The total sum is :53.0
The Total sum is :17.499999999999996
```

(b) Lower bounded wildcards:-

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

public class LowerBoundedWildcards {
   public static void main(String[] args) {
      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:

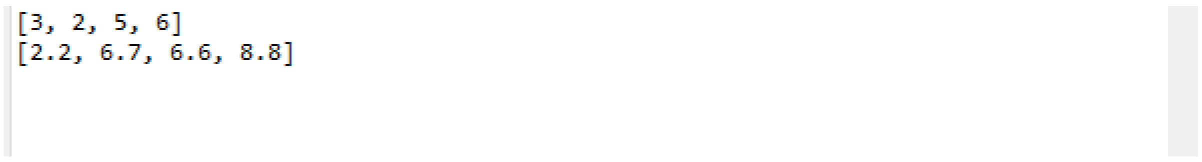


```
[7, 8, 9, 3, 4, 5]
[3, 45, 56, 7, 4]
```

(c) Unbounded wildcards:-

```
package ASSIGNMENT;
import java.util.*;
public class UnboundedWildcards {
    public static void main(String[] args) {
        System.out.println("Name:- Susmit Naik, Roll No.:- 35");
        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:



```
[3, 2, 5, 6]
[2.2, 6.7, 6.6, 8.8]
```

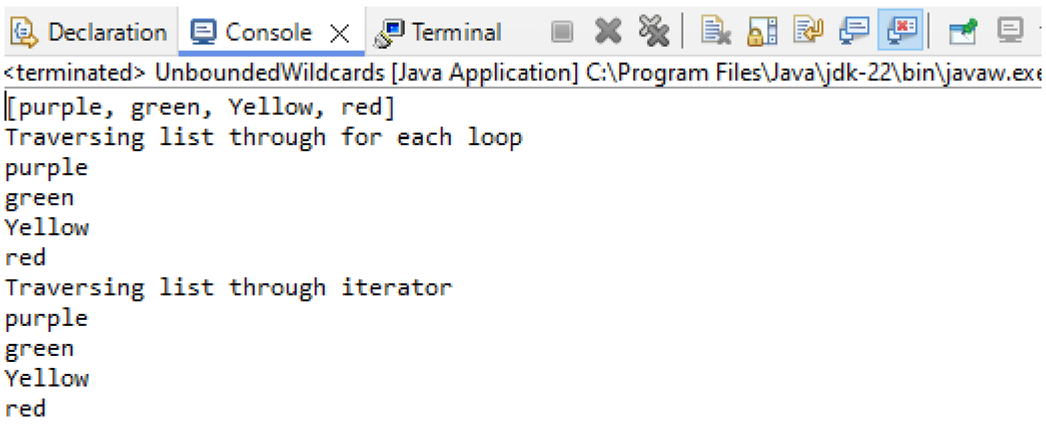
2) Assignment on List Interface:-

(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 ASSIGNMENT;
import java.util.*;
public class ArrayList {
  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("Traversing list through iterator");
    Iterator itr = list.iterator();
    while (itr.hasNext()) {
      System.out.println(itr.next());
    }
  }
}
```

OUTPUT:



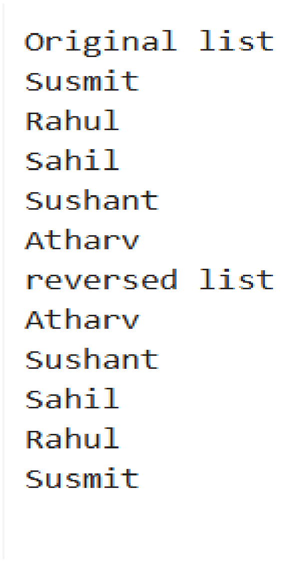
```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe
[purple, green, Yellow, red]
Traversing list through for each loop
purple
green
Yellow
red
Traversing list through iterator
purple
green
Yellow
red
```

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 ASSIGNMENT;
import java.util.*;
public class ListIterator1{
   public static void main(String args[]) {
      List<String> mylist = new ArrayList<String>();
      mylist.add("Susmit");
      mylist.add("Rahul");
      mylist.add("Sahil");
      mylist.add("Sushant");
      mylist.add("Atharv");
      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:



```
Original list
Susmit
Rahul
Sahil
Sushant
Atharv
reversed list
Atharv
Sushant
Sahil
Rahul
Susmit
```

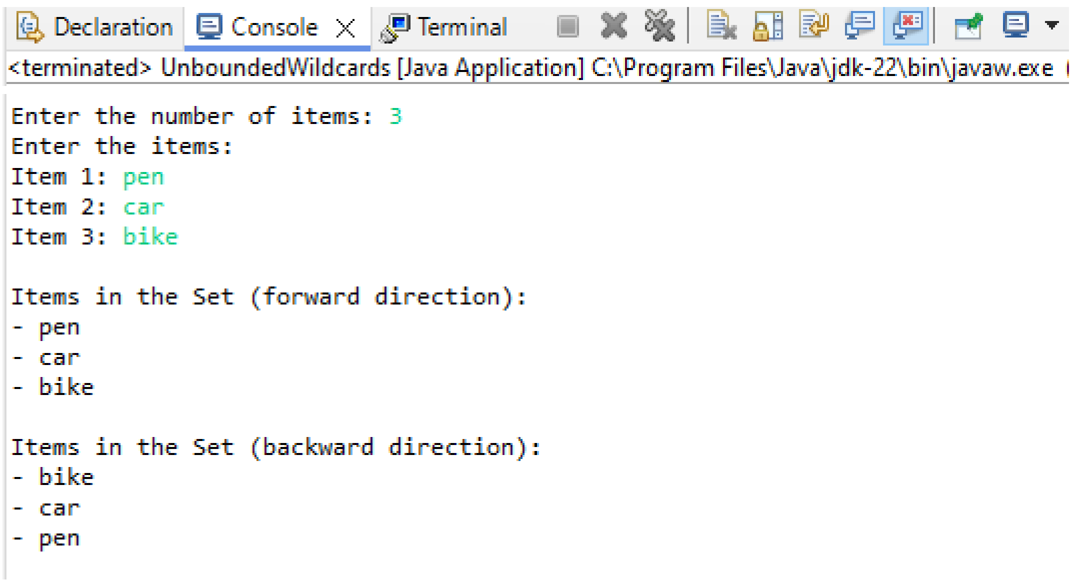
3) Assignment on SET Interface:-

(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 ASSIGNMENT;
import java.util.*;
public class SetIterator1 {
   public static void main(String[] args) {
      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:



```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe
Enter the number of items: 3
Enter the items:
Item 1: pen
Item 2: car
Item 3: bike

Items in the Set (forward direction):
- pen
- car
- bike

Items in the Set (backward direction):
- bike
- car
- pen
```

(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

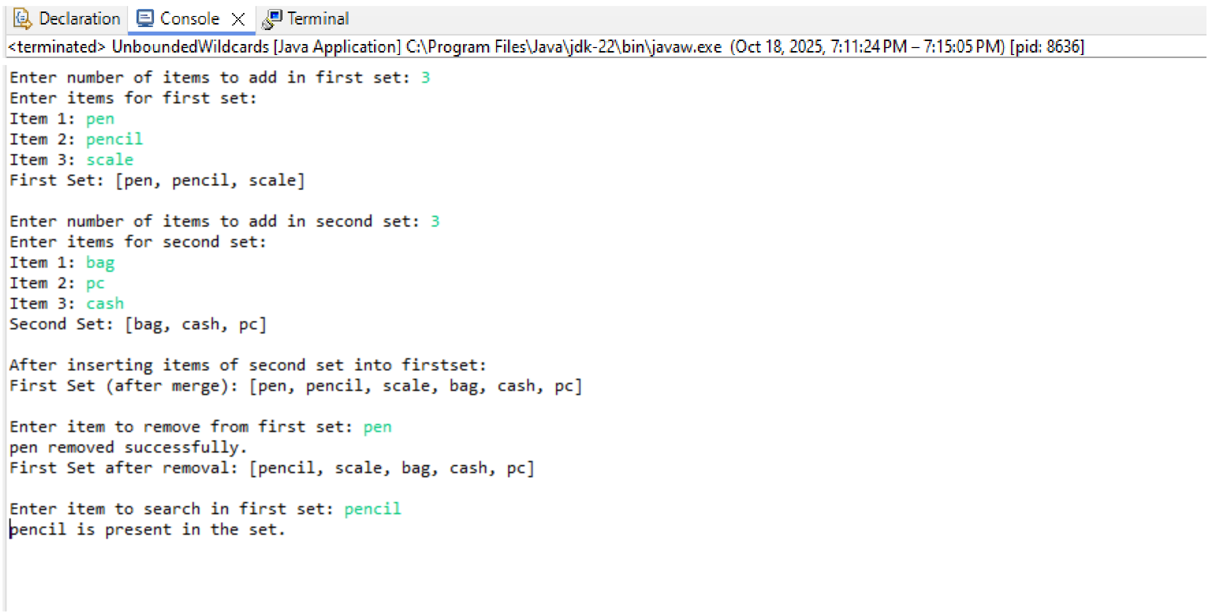
d. Search the specified item in the set

CODE:

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

public class SetLterator2{
   public static void main(String[] args) {
      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());
      }
```

```java
        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:



```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe  (Oct 18, 2025, 7:11:24 PM – 7:15:05 PM) [pid: 8636]
Enter number of items to add in first set: 3
Enter items for first set:
Item 1: pen
Item 2: pencil
Item 3: scale
First Set: [pen, pencil, scale]

Enter number of items to add in second set: 3
Enter items for second set:
Item 1: bag
Item 2: pc
Item 3: cash
Second Set: [bag, cash, pc]

After inserting items of second set into firstset:
First Set (after merge): [pen, pencil, scale, bag, cash, pc]

Enter item to remove from first set: pen
pen removed successfully.
First Set after removal: [pencil, scale, bag, cash, pc]

Enter item to search in first set: pencil
pencil is present in the set.
```

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.

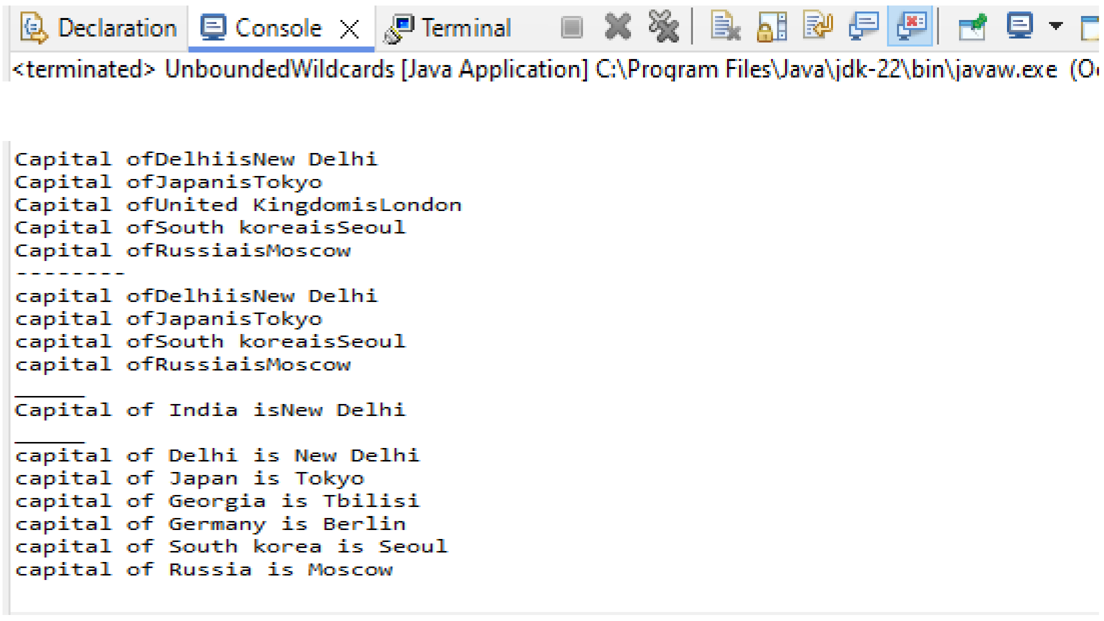
f. Print all keys and values of the map.

CODE:

```
package ASSIGNMENT;
import java.util.*;
public class mapiterator1 {
   public static void main(String args[]) {
      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:



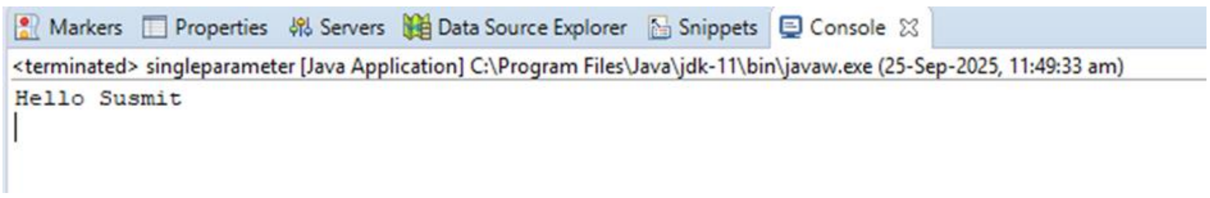
```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw.exe (O

Capital ofDelhiisNew Delhi
Capital ofJapanisTokyo
Capital ofUnited KingdomisLondon
Capital ofSouth koreaisSeoul
Capital ofRussiaisMoscow
--------
capital ofDelhiisNew Delhi
capital ofJapanisTokyo
capital ofSouth koreaisSeoul
capital ofRussiaisMoscow
_____
Capital of India isNew Delhi
_____
capital of Delhi is New Delhi
capital of Japan is Tokyo
capital of Georgia is Tbilisi
capital of Germany is Berlin
capital of South korea is Seoul
capital of Russia is Moscow
```

(5) LAMBDA EXPRESSIONS

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

CODE :-

```
package Lambdaexpression;
interface Say {
public String say(String name);
}
public class singleparameter {
        public static void main(String[] args) {
        Say s1=(name)->{
                return "Hello "+name;
        };
System.out.println(s1.say("Susmit"));
}
}
```



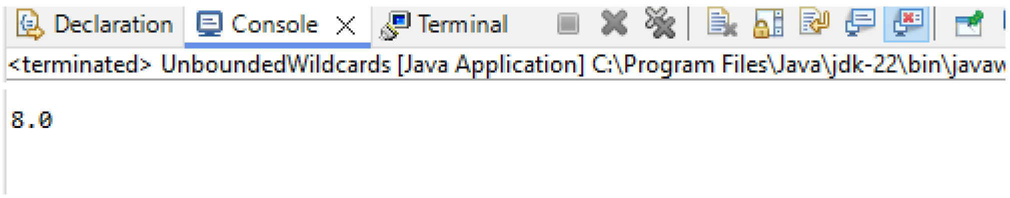
```
<terminated> singleparameter [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (25-Sep-2025, 11:49:33 am)
Hello Susmit
```

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

CODE:-

```
package ASSIGNMENT;
interface cube {
    double cubevolume(double L);
}

public class LambdaSinglePara{
    public static void main(String[] args) {
        cube c2 = (L) -> {
            return L * L * L;
        };
        System.out.print(c2.cubevolume(2));
    }
}
```

OUTPUT:



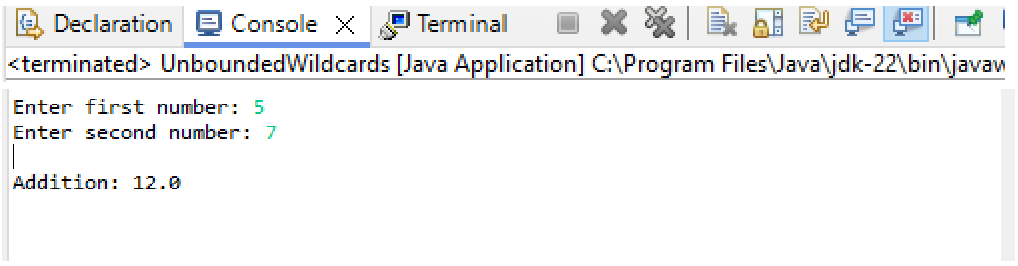
```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw
8.0
```

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

CODE:

```
package ASSIGNMENT;
import java.util.*;
interface Lambda {
    double addNum(double a, double b);
}

public class LambdaTwoPara{
    public static void main(String[] args) {
        System.out.println("Name: Susmit naik, Div A, Roll No: 35\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:



```
<terminated> UnboundedWildcards [Java Application] C:\Program Files\Java\jdk-22\bin\javaw
Enter first number: 5
Enter second number: 7

Addition: 12.0
```

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

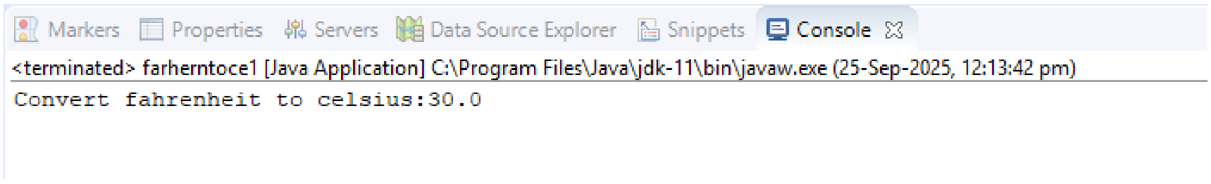
a. Convert Fahrenheit to Celcius

b. Convert Kilometers to Miles.

CODE:-

a.Fahrenheit to celcius

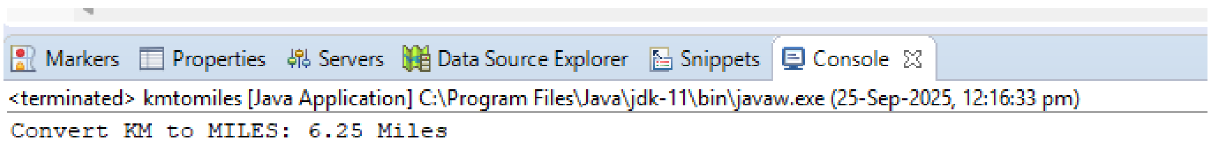
```
package Lambdaexpression;


interface temp {
        public double convert(double temp);
}

public class farherntoce1 {
	public static void main(String[] args) {
	temp t1=(double a)->{
	return((a-32) * 5/9);
	};
	System.out.print("Convert fahrenheit to celsius:" + t1.convert(86));
	}
}
```



```
<terminated> farherntoce1 [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (25-Sep-2025, 12:13:42 pm)
Convert fahrenheit to celsius:30.0
```

b. Kilometer to miles

```
package Lambdaexpression;
interface templ
{
	public double convert(double temp);
}
public class kmtomiles {
	public static void main(String[] args) {
		temp t1=(double a)->{
			return(a/1.6);
};
System.out.print("Convert KM to MILES: "+ t1.convert(10)+" Miles");
}
}
```

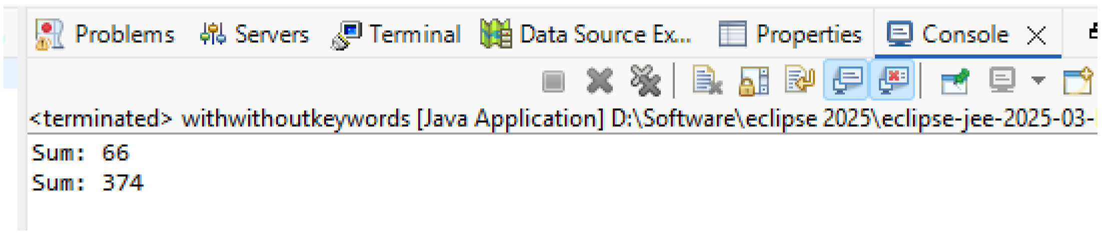


```
<terminated> kmtomiles [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (25-Sep-2025, 12:16:33 pm)
Convert KM to MILES: 6.25 Miles
```

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

CODE :-

```
package Lambdaexpression;
interface Add2 {
            int add(int a, int b);
        }
        public class withwithoutkeywords {
          public static void main(String[] args) {
              // without return keyword
              Add2 ad1=(a,b)->(a+b);
              System.out.println("Sum: " + ad1.add(43,23));
              // with return keyword
              Add2 ad2=(int a,int b)-> { return (a+b); };
              System.out.println("Sum: " + ad2.add(54,320));
          }
        }
```



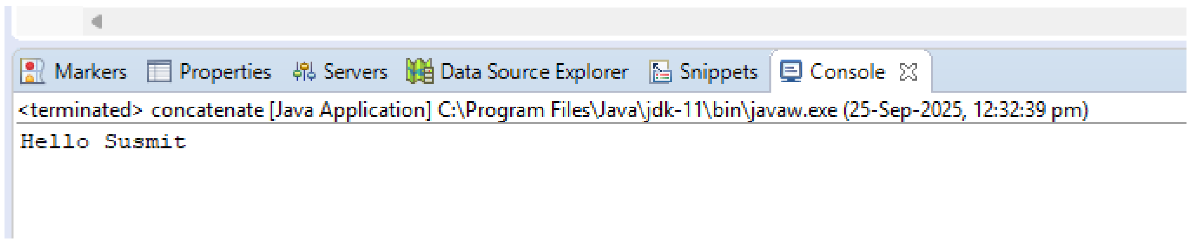
```
<terminated> withwithoutkeywords [Java Application] D:\Software\eclipse 2025\eclipse-jee-2025-03-
Sum: 66
Sum: 374
```

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

CODE :-

```
 interface concl {

        public String concat(String a, String b);
}
public class concatenate {
        public static void main(String[] args) {
                concl s1 = (String a,String b)->{
                        return (a+b);
                };
                System.out.println(s1.concat("Hello"," Susmit"));
        }
}
```

Output:-



```
<terminated> concatenate [Java Application] C:\Program Files\Java\jdk-11\bin\javaw.exe (25-Sep-2025, 12:32:39 pm)
Hello Susmit
```