

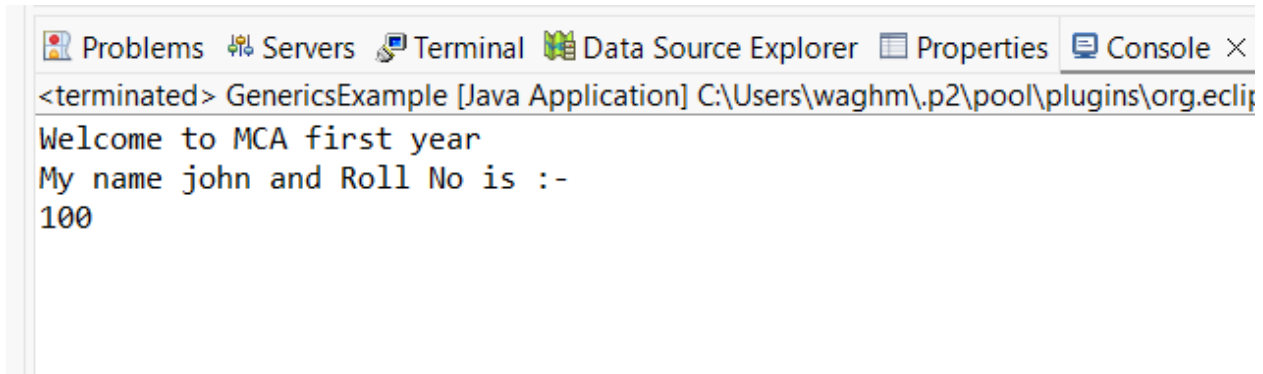
1)Assignment on Java Generics:-

1.1 Write a java program to demonstrate the Generic Class.

CODE:-

```
package mypack;
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 john and Roll No is :-");
        System.out.println(S.get());
        Geg <Integer> i = new Geg<Integer>(100);
        System.out.println(i.get());;
    }
}
```

Output:

A screenshot of an IDE's console window. The window has a title bar with icons for Problems, Servers, Terminal, Data Source Explorer, Properties, and Console. The console text shows the program's output: a terminated Java application, followed by three lines of printed text: "Welcome to MCA first year", "My name john and Roll No is :-", and "100".

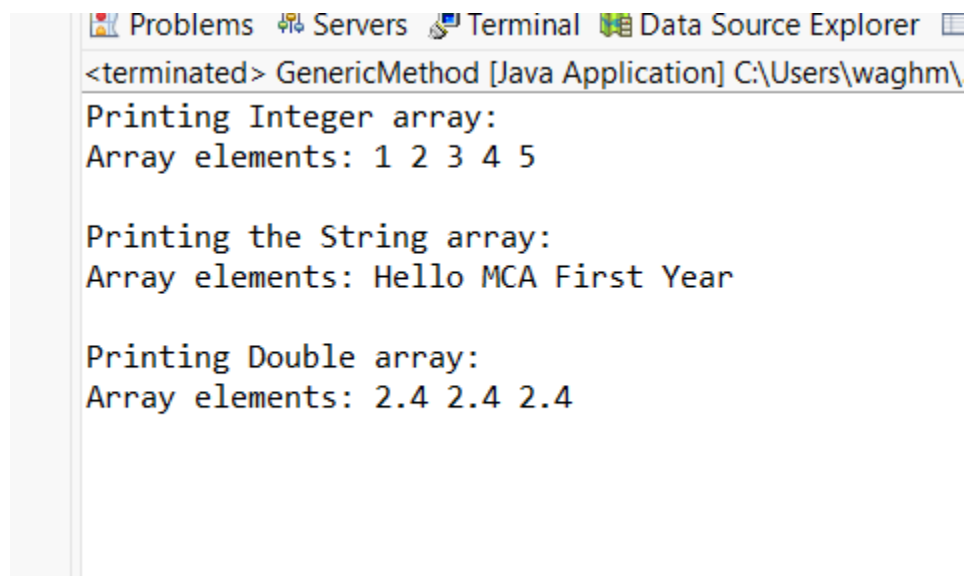
```
<terminated> GenericsExample [Java Application] C:\Users\waghm\.p2\pool\plugins\org.eclips
Welcome to MCA first year
My name john and Roll No is :-
100
```

1.2 Write a java program to demonstrate Generic Methods.

CODE:

```
package mypack;
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:

A screenshot of an IDE terminal window. The title bar shows 'Problems', 'Servers', 'Terminal', and 'Data Source Explorer'. The terminal content shows the output of the Java program: '<terminated> GenericMethod [Java Application] C:\Users\waghm\' followed by three sections of output. The first section is 'Printing Integer array:' followed by 'Array elements: 1 2 3 4 5'. The second section is 'Printing the String array:' followed by 'Array elements: Hello MCA First Year'. The third section is 'Printing Double array:' followed by 'Array elements: 2.4 2.4 2.4'.

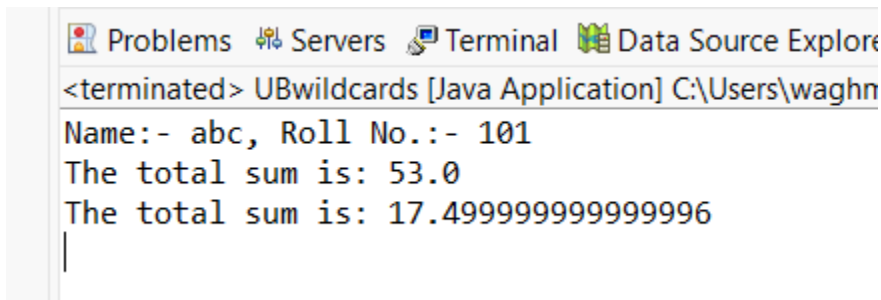
```
<terminated> GenericMethod [Java Application] C:\Users\waghm\  
Printing Integer array:  
Array elements: 1 2 3 4 5  
  
Printing the String array:  
Array elements: Hello MCA First Year  
  
Printing Double array:  
Array elements: 2.4 2.4 2.4
```

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

(a) Upper bounded wildcards:-

```
package mypack;
import java.util.*;
public class UBwildcards {
    public static void main(String[] args) {
        System.out.println("Name:- abc, Roll No.:- 101");
        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:

A screenshot of an IDE terminal window. The title bar shows 'Problems', 'Servers', 'Terminal', and 'Data Source Explorer'. The terminal content shows the output of the Java program: '<terminated> UBwildcards [Java Application] C:\Users\waghn', followed by 'Name:- abc, Roll No.:- 101', 'The total sum is: 53.0', and 'The total sum is: 17.499999999999996'.

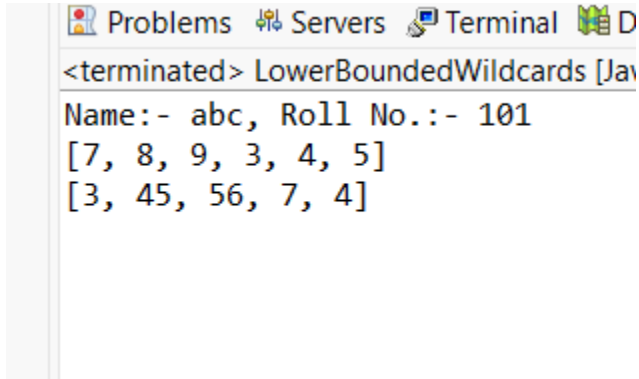
```
<terminated> UBwildcards [Java Application] C:\Users\waghn
Name:- abc, Roll No.:- 101
The total sum is: 53.0
The total sum is: 17.499999999999996
```

(b) Lower bounded wildcards:-

```
package mypack;
import java.util.*;
public class LowerBoundedWildcards {
    public static void main(String[] args) {
        System.out.println("Name:- abc, Roll No.:- 101");
        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:

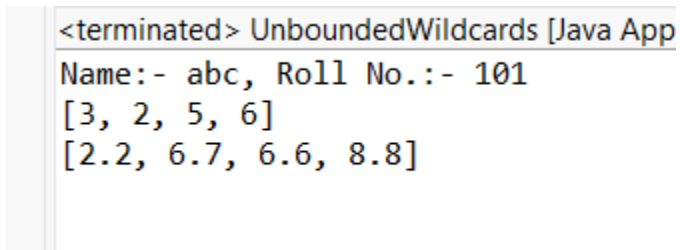


```
Problems Servers Terminal D
<terminated> LowerBoundedWildcards [Java App]
Name:- abc, Roll No.:- 101
[7, 8, 9, 3, 4, 5]
[3, 45, 56, 7, 4]
```

(c) Unbounded wildcards:-

```
package mypack;  
import java.util.*;  
public class UnboundedWildcards {  
    public static void main(String[] args) {  
        System.out.println("Name:- abc, Roll No.:- 101");  
        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:



```
<terminated> UnboundedWildcards [Java App]
Name:- abc, Roll No.:- 101
[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.

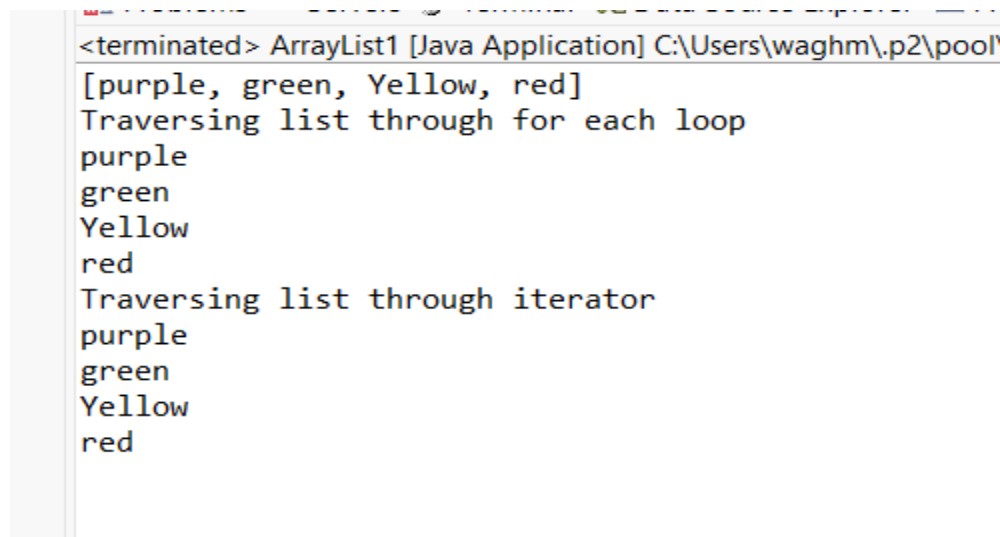
CODE:

```
package mypack;
import java.util.*;
public class ArrayList1 {
    public static void main(String args[]) {

        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<String> itr = list.iterator();
        while (itr.hasNext()) {
            System.out.println(itr.next());
        }
    }
}
```

Output:



The screenshot shows a Java application window titled "<terminated> ArrayList1 [Java Application] C:\Users\waghm\.p2\pool". The output displayed in the window is as follows:

```
[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 mypack;
import java.util.*;
public class ListIterator1{
    public static void main(String args[]) {
        List<String> mylist = new ArrayList<String>();
        mylist.add("Ritesh");
        mylist.add("rahul");
        mylist.add("karan");
        mylist.add("rohit");

        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

```
<terminated> ListIterator1 [Java Applicati
Original list
Ritesh
rahul
karan
rohit
reversed list
rohit
karan
rahul
Ritesh
```

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 mypack;
import java.util.*;
public class SetIterator1 {
    public static void main(String[] args) {
        System.out.println("Name: abc, Div: A, Roll No: 101\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:

```
terminated: C:\Users\pratik\AppData\Local\Programs\Java\jdk-11.0.10\bin\java.exe
Name: abc, Div: A, Roll No: 101

Enter the number of items: 3
Enter the items:
Item 1: pen
Item 2: pencil
Item 3: book

Items in the Set (forward direction):
- pen
- pencil
- book

Items in the Set (backward direction):
- book
- pencil
- 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 mypack;
import java.util.*;
public class SetLterator2{
    public static void main(String[] args) {
        System.out.println("Name: abc, Div: A, Roll No: 101\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.println("\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.println("\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.println("\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> SetLterator2 [Java Application] C:\Users\waghm\p2\pool\plugins\org.e
Name: abc, Div: A, Roll No: 101

Enter number of items to add in first set: 3
Enter items for first set:
Item 1: pen
Item 2: pencil
Item 3: book
First Set: [pen, pencil, book]

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 first set:
First Set (after merge): [pen, pencil, book, bag, cash, pc]

Enter item to remove from first set: pen
pen removed successfully.
First Set after removal: [pencil, book, 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 mypack;
import java.util.*;
public class mapiterator1 {
    public static void main(String args[]) {
        System.out.println("Name: abc, Div: A, Roll No: 101\n");
        Map<String,String> hmap = new HashMap<>();
        hmap.put("India", "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> mapiterator1 [Java Application] C:\Users\wa
Name: abc, Div: A, Roll No: 101

Capital of Japan is Tokyo
Capital of United Kingdom is London
Capital of India is New Delhi
Capital of South korea is Seoul
Capital of Russia is Moscow
-----
Capital of Japan is Tokyo
Capital of India is New Delhi
Capital of South korea is Seoul
Capital of Russia is Moscow

-----
Capital of India is null

-----
Capital of Japan is Tokyo
Capital of Georgia is Tbilisi
Capital of Germany is Berlin
Capital of India is New Delhi
Capital of South korea is Seoul
Capital of Russia is Moscow

```

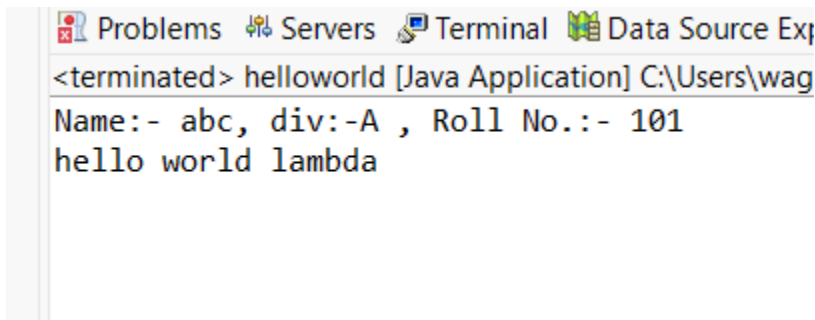
(5) LAMBDA EXPRESSIONS

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

CODE :-

```
package mypack;
interface hello1 {
    void hi();
}
public class helloworld {
    public static void main(String[] args) {
        System.out.println("Name:- abc, div:-A , Roll No.:- 101");
        hello1 h1 = () -> {
            System.out.print("hello world lambda");
        };
        h1.hi();
    }
}
```

Output:

A screenshot of an IDE terminal window. The title bar shows 'Problems', 'Servers', 'Terminal', and 'Data Source Explorer'. The terminal content shows the execution of the 'helloworld' Java application. The first line of output is 'Name:- abc, div:-A , Roll No.:- 101' and the second line is 'hello world lambda'.

```
<terminated> helloworld [Java Application] C:\Users\wag
Name:- abc, div:-A , Roll No.:- 101
hello world lambda
```

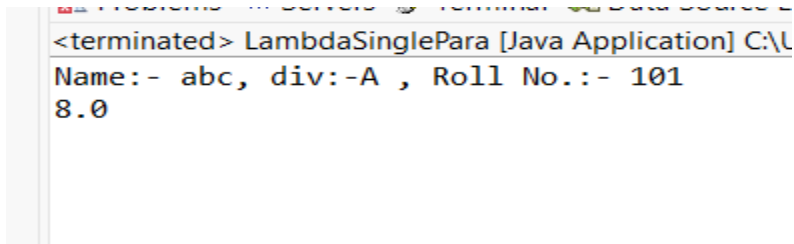
(5.2) Write a Java program using Lambda Expression with single parameters.

CODE:-

```
package mypack;
interface cube {
    double cubevolume(double L);
}
public class LambdaSinglePara{
    public static void main(String[] args) {
        System.out.println("Name:- abc, div:-A , Roll No.:- 101");
        cube c2 = (L) -> {
            return L * L * L;
        };
        System.out.print(c2.cubevolume(2));
    }
}
```

```
}
```

Output:



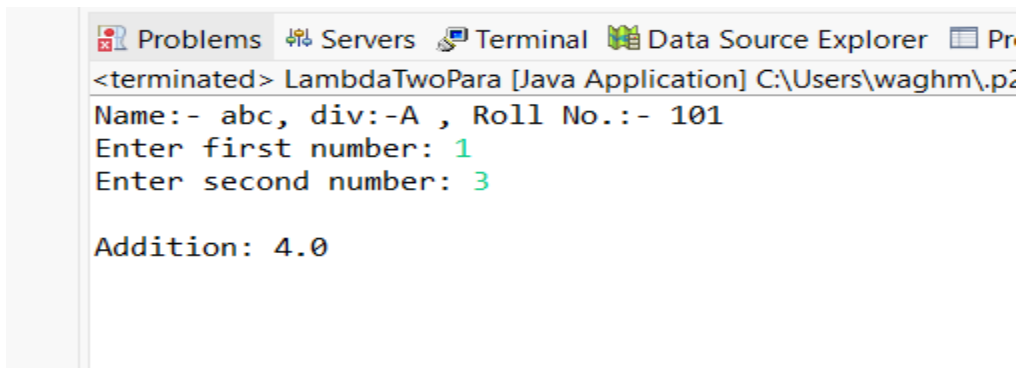
```
<terminated> LambdaSinglePara [Java Application] C:\L
Name:- abc, div:-A , Roll No.:- 101
8.0
```

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

CODE:

```
package mypack;
import java.util.*;
interface Lambda {
    double addNum(double a, double b);
}
public class LambdaTwoPara{
    public static void main(String[] args) {
        System.out.println("Name:- abc, div:-A , Roll No.:- 101");
        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> LambdaTwoPara [Java Application] C:\Users\waghm\p
Name:- abc, div:-A , Roll No.:- 101
Enter first number: 1
Enter second number: 3

Addition: 4.0
```

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

a. Convert Fahrenheit to Celcius

b. Convert Kilometers to Miles.

CODE:-

```
package mypack;
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:- abc, div:-A , Roll No.:- 101");
        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:

```
<terminated> Temp [Java Application] C:\Users\waghr
Name:- abc, div:-A , Roll No.:- 101
Choose Conversion Type:
1. Celsius to Fahrenheit
2. Kilometer to Mile
Enter your choice (1 or 2): 1
Enter temperature in Celsius: 30
Temperature in Fahrenheit: 86.0
```

```
Problems Servers Terminal Data Source Explorer Prop
<terminated> Temp [Java Application] C:\Users\waghr\p2\pool\plugin
Name:- abc, div:-A , Roll No.:- 101
Choose Conversion Type:
1. Celsius to Fahrenheit
2. Kilometer to Mile
Enter your choice (1 or 2): 2
Enter distance in Kilometers: 56
Distance in Miles: 34.888
```

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

CODE :-

```
package mypack;
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:- abc, div:-A , Roll No.:- 101");
        Scanner sc = new Scanner(System.in);
```

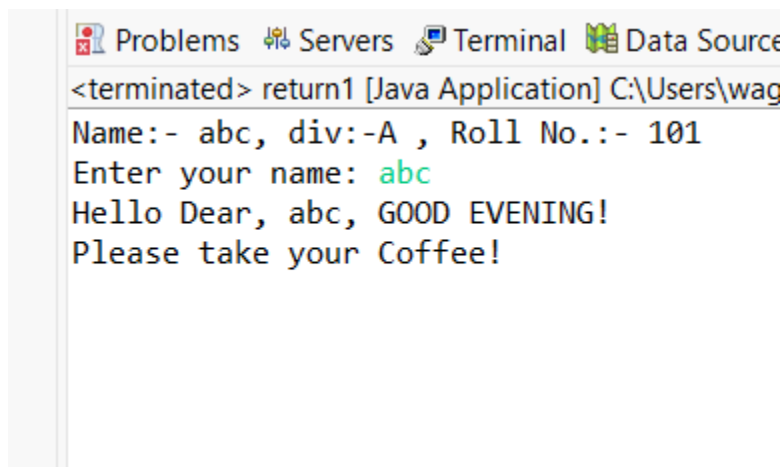


```

    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:



```

<terminated> return1 [Java Application] C:\Users\wag
Name:- abc, div:-A , Roll No.:- 101
Enter your name: abc
Hello Dear, abc, GOOD EVENING!
Please take your Coffee!

```

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

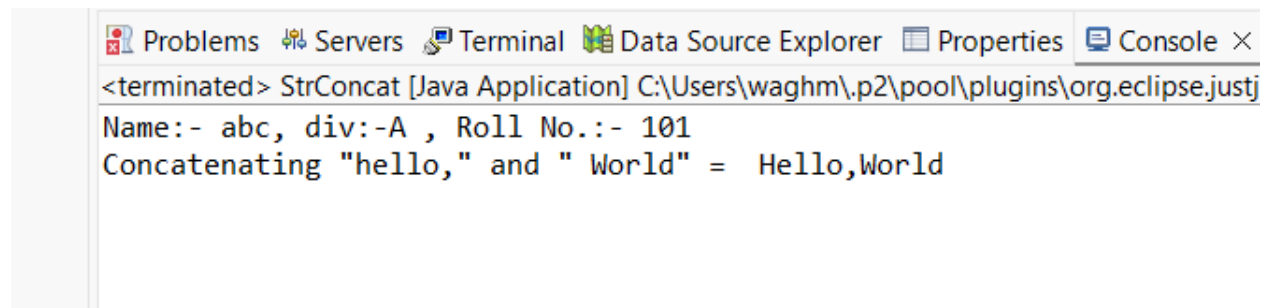
Code:

```

package mypack;
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:- abc, div:-A , Roll No.:- 101");
        ConcatStr cs = (String s1, String s2) -> s1 + s2;
        System.out.println("Concatenating \"hello,\" and \" World\" = " + cs.concat(" Hello," , "World"));
    }
}

```

Output:



The screenshot shows the Eclipse IDE's Console window. The title bar includes tabs for Problems, Servers, Terminal, Data Source Explorer, Properties, and Console. The console text shows the application has terminated, followed by two lines of output: 'Name:- abc, div:-A , Roll No.:- 101' and 'Concatenating "hello," and " World" = Hello,World'.

```
<terminated> StrConcat [Java Application] C:\Users\waghm\.p2\pool\plugins\org.eclipse.justj  
Name:- abc, div:-A , Roll No.:- 101  
Concatenating "hello," and " World" = Hello,World
```