Assignment 1

Java Generics

1. Write a Java Program to demonstrate a Generic Class.

2. Write a Java Program to demonstrate Generic Methods.

3. Write a Java Program to demonstrate Wildcards in Java Generics.

Problem Statement 1 : Write a Java Program to demonstrate a Generic Class.

Code :

```
package bvimit;  public class geg53<T>{  T obj;  geg53(T

obj){this.obj = obj;} public T get() {return this.obj;}

}

class G1

{

public static void main (String[] args)

{

geg53<Integer>i=new geg53<Integer>(35); System.out.println(i.get());

geg53<String> s = new geg53<String>("Mayur");

System.out.println(s.get());

}

}
```

Output :-

Problem Statement 2 : Write a Java Program to demonstrate Generic

Methods.Code : package bvimit; public class Genericmethod53 { void

display()

{

System.out.println("generic method exmaple");

}

<T> void gdisplay (T e)

{

System.out.println(e.getClass().getName() + " = " + e);

} public static void main(String[]

args)

{

Genericmethod53 g1=new Genericmethod53();

g1.display(); g1.gdisplay(1); g1.gdisplay("Mayur");

g1.gdisplay(11.0);

}

}

Output :

Problem Statement 3 : Write a Java Program to demonstrate Wildcards in Java

Generics.Code : import java.util.*; public class Wildcard53 { // Upper bounded

private static double sum(List<? extends Number> list)

{double sum = 0.0; for

(Number i : list) { sum =

sum + i.doubleValue();

} return

sum;

}

// Lower Bounded private static void

show(List<? super Integer> list)

{list.forEach((x) -> {

System.out.print(x + " ");

});

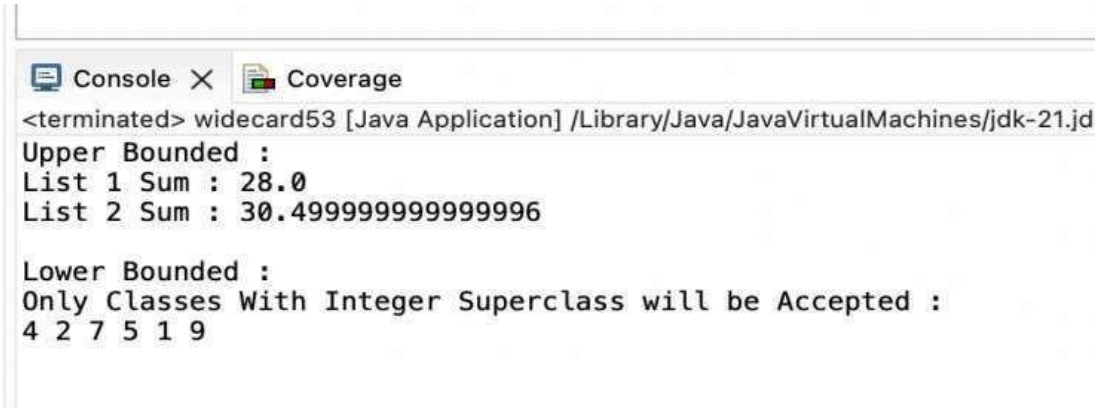
}

public static void main(String[] args)

{ System.out.println("Upper Bounded : ");

```
List<Integer> list1 = Arrays.asList(4, 2, 7, 5, 1,

9);System.out.println("List 1 Sum : " +

sum(list1));

List<Double> list2 = Arrays.asList(4.7, 2.4, 7.3, 5.4, 1.5,

9.2);System.out.println("List 2 Sum : " + sum(list2));

System.out.println("\nLower Bounded : ");

List<Integer> list3 = Arrays.asList(4, 2, 7, 5, 1, 9);

System.out.println("Only Classes With Integer Superclass will be Accepted :

");show(list3); }

}
```

Output :



Assignment 2
List Interface

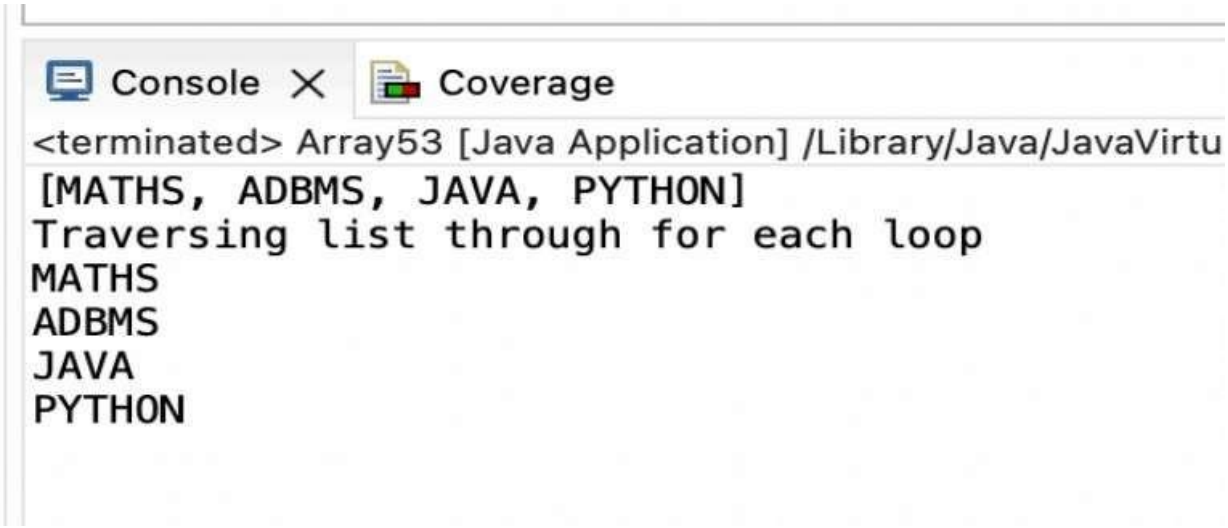
1. Write a Java program to create List containing list of items of type String and use for- --each loop to
print the items of the list.
2. Write a Java program to create List containing list of items and use ListIterator interface to print
items present in the list. Also print the list in reverse/ backward direction.

Problem Statement 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 bvimit;
import java.util.*;
public class Array53 {
public static void main(String[] args)
{  ArrayList<String>list=newArrayList<String>();
list.add("MATHS");
list.add("ADBMS");
list.add("JAVA");
list.add("PYTHON");
System.out.println(list);
System.out.println("Traversing list through for each loop");
for(String subject:list)
System.out.println(subject);
}
}
```

Output : -



Problem Statement 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 bvimit;

import java.util.*;
public class Reverse {
public static void main(String[] args) {
List<String> mylist = new ArrayList<String>();
mylist.add("Mayur");
mylist.add("aniket");
mylist.add("yash");
mylist.add("Sahil");
mylist.add("omkar");
System.out.println("Traversing through iterator");
System.out.println("Original List:");
Iterator itr=mylist.iterator();
while(itr.hasNext()) {
System.out.println(itr.next());
}
Collections.reverse(mylist);
System.out.println(); //space between two lines
System.out.println("Reversed List:");
Iterator itr1=mylist.iterator();
while(itr1.hasNext()) {
System.out.println(itr1.next());
}
}

}
```

Output :-


Problems
Javadoc
Declaration
Console
<terminated> Reverse [Java Application] C:\Users\LENOVO\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_17.0.8.v20230831-1047\jre\bin\javaw.exe (02-Dec-2024, 8:43:24 pm – 8:43:26 pm)
Reversed List:
omkar
Sahil
yash
aniket
Mayur

Assignment 3
Set Interface

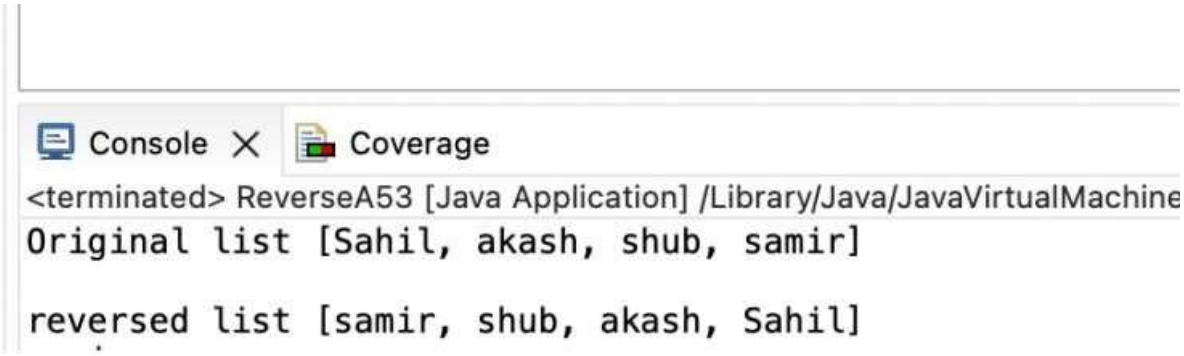
1. Write a Java program to create a Set containing list of items of type String and print the items in thelist
using Iterator interface. Also print the list in reverse/ backword direction.
2. Write a Java program using Set interface containing list of items and perform the following operations:
a. Add 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.

Problem Statement 1 : Write a Java program to create a Set containing list of items of type String andprint
the items in the list using Iterator interface. Also print the list in reverse/ backword direction.
Solution :

```
Package bvimit;
import java.util.*;
public class ReverseA53
{
public static void main(String[] args) {
// Let us create a list of strings
List<String> mylist = new ArrayList<String>();
mylist.add("Sahil");
mylist.add("akash");
mylist.add("shub");
mylist.add("samir");
System.out.println("Original list ");
Iterator<String> itr=mylist.iterator();//getting the Iterator
while(itr.hasNext())
{//check if iterator has the elements
System.out.println(itr.next());
}
Collections.reverse(mylist);
System.out.println(" ");
System.out.println("reversed list ");
Iterator<String> itr1=mylist.iterator();//getting the Iterator
while(itr1.hasNext()){//check if iterator has the elements
```

```
System.out.println(itr1.next());
}
}
}
```

Output :



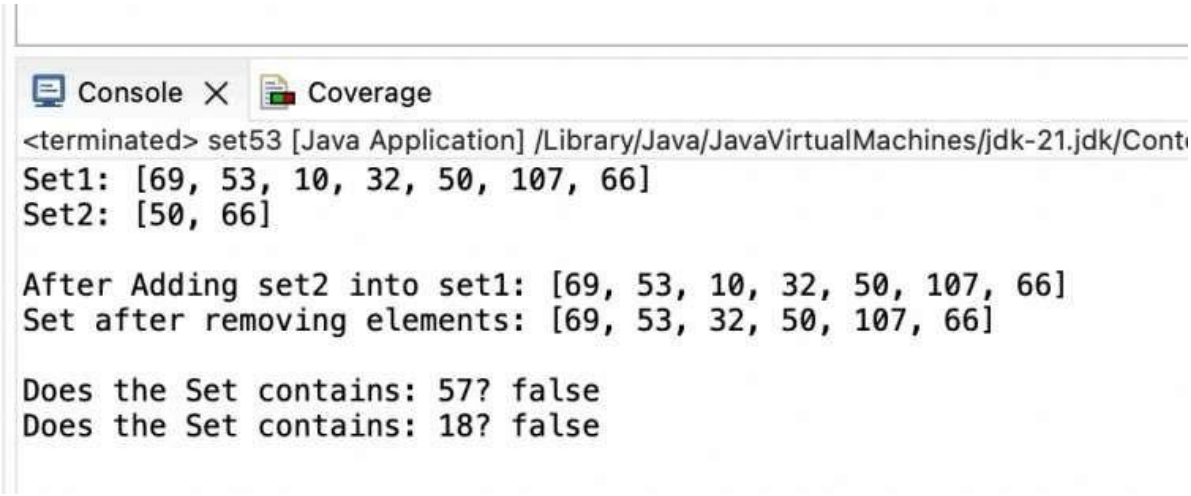
Problem Statement2 : Write a Java program using Set interface containing list of items and perform the
following operations:
a. Add items in the set.
b. Insert items of one set in to other set.
c. Remove items from the set
d. Search the specified item in the set

Solution :

```
package bvimit;
import java.util.*;
public class set53{
public static void main(String[] args) {
// TODO Auto-generated method stub Set<Integer> s
= new LinkedHashSet<Integer>();s.add(69);
s.add(57);
s.add(10);
s.add(18);
s.add(90);
s.add(151);
Set<Integer> s1 = new LinkedHashSet<Integer>();
s1.add(70);
s1.add(35);
s.addAll(s1);
System.out.println("Set1: " + s);
System.out.println("Set2: " + s1);
System.out.println();
```

```
System.out.println("After Adding set2 into set1: " + s);
s.remove(10);
s.remove(18);
System.out.println("Set after removing elements: " + s);
System.out.println();
System.out.println("Does the Set contains: 57? "
+ s.contains(57));
System.out.println("Does the Set contains: 18? "
+ s.contains(18));
}
}
```

Output :-



Assignment 4
Map Interface

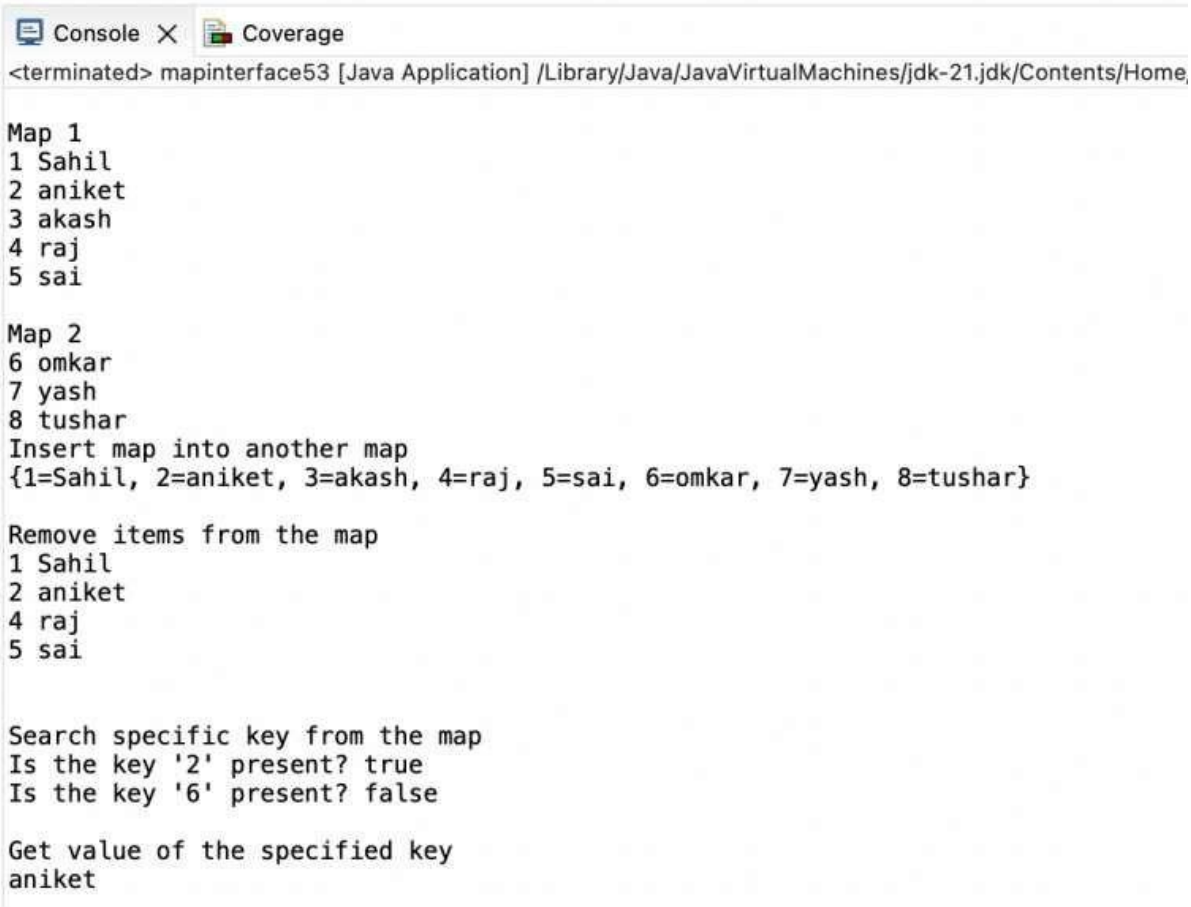
1. Write a Java program using Map interface containing list of items having keys and associated valuesand
perform the following operations:
a. Add items in the map.
b. Remove items from the map
c. Search specific key from the map
d. Get value of the specified key
e. Insert map elements of one map in to other map.
f. Print all keys and values of the map.

Solution :

```
Package bvimit ;
import java.util.*;
public class mapinterface53 {
public static void main(String[] args) {
// TODO Auto-generated method stub
Map<Integer, String> map = new HashMap<>();
map.put(1 ,"Sahil");
map.put(2,"aniket");
map.put(3,"akash");
map.put(4,"raj");
map.put(5,"sai");
System.out.println();
Map<Integer, String> map1 = new HashMap<>();
map1.put(6 ,"omkar");
map1.put(7,"yash");
map1.put(8,"tushar");
System.out.println("Map 1");
for (Map.Entry<Integer, String> e : map.entrySet())
System.out.println(e.getKey() + " " + e.getValue());
System.out.println();
System.out.println("Map 2");
for (Map.Entry<Integer, String> e : map1.entrySet())
System.out.println(e.getKey() + " " + e.getValue());
System.out.println("Insert map into another map");
Map<Integer, String> map2 = new HashMap<>();
map2.putAll(map); map2.putAll(map1);
System.out.println(map2);
```

```
System.out.println();
System.out.println("Remove items from the map");
map.remove((3));
for (Map.Entry<Integer, String> e : map.entrySet())
System.out.println(e.getKey() + " "+ e.getValue());
System.out.println();
System.out.println();
System.out.println("Search specific key from the map");
System.out.println("Is the key '2' present? " +
map.containsKey(2));
System.out.println("Is the key '6' present? " +
map.containsKey(6));
System.out.println();
System.out.println("Get value of the specified key");
String val = (String)map.get(2); System.out.println(val);
System.out.println();
}
}
```

Output :



Assignment 5
Lambda Expressions

1. Write a Java program using Lambda Expression to print “Hello World!”.
2. Write a Java program using Lambda Expression with single parameter.

Problem Statement 1 : Write a Java program using Lambda Expression to print “Hello World!”.

Solution :

```
package bvimit ;
interface HelloWorld53 {
String sayHello(String name);
}
public class helloworld53 {
public static void main(String args[]){
HelloWorld53 helloWorld = (String name) -> { return "Hello " + name; };
System.out.println(helloWorld.sayHello("World"));
}
}
```

Output :

Problem Statement 2 : Write a Java program using Lambda Expression with single parameter.

Solution :

```
package bvimit;
interface Say{
public String say(String name);
}
public class singleparameter53{
public static void main(String[] args) {Say
s1=(name)->{
return "Hello "+name;
```

```
};
System.out.println(s1.say("Sahil"));
}
}
```

Output :