LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE
OF COMPUTER APPLICATION, MUMBAI

Advance Java Practical Journal

PROF. **VIKRAM PATALBANSI,**
DR. **RASHMITA PRADHAN,**
AQUILA SHAIKH

Faculty, Master of Computer Application (M.C.A.)
Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application

16LEAVES

ADVANCE JAVA PRACTICAL JOURNAL

ADVANCE JAVA PRACTICAL JOURNAL

LATE BHAUSAHEB HIRAY S.S. TRUST'S INSTITUTE OF COMPUTER APPLICATION BANDRA, EAST MUMBAI-51

Prof. Vikram Patalbansi
Co Author : Dr. Rashmita Pradhan
Co Author : Aquila Shaikh (MCA)

Faculty, Master of Computer Application (M.C.A.)
Late Bhausaheb Hiray S.S. Trust's Institute of Computer Application

16LEAVES

First Edition, 2023

Paperback ISBN: 978-81-19221-17-2
eBook ISBN: 978-81-19221-22-6
WebPDF ISBN: 978-81-19221-05-9

Note: Due care and diligence has been taken while editing and printing the book; neither the author nor the publishers of the book hold any responsibility for any mistake that may have inadvertently crept in.

The publishers shall not be liable for any direct, consequential, or incidental damages arising out of the use of the book. In case of binding mistakes, misprints, missing pages, etc., the publishers' entire liability, and your exclusive remedy, is replacement of the book within one month of purchase by similar edition/reprint of the book.

Printed and bound in India by
16Leaves
2/579, Singaravelan Street
Chinna Neelankarai
Chennai – 600 041, India

info@16leaves.com
www.16Leaves.com
Call: 91-9940638999

Contents

1 **Practical on Java Collection Framework** 1

Write a Java Program to demonstrate a Generic Class.

Write a Java Program to demonstrate Generic Methods.

Write a Java Program to demonstrate Generic Method –part 2

Write Program in java to understand the use of unbound wildcards

Write Program in java to understand the use of upper bound wildcards

2 **Practical on List Interface** 7

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.

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.

3 **Practical on Set Interface** 11

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

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

4 **Practical on Map Interface** 21

Write a Java program using Map interface containing list of items having keys andassociated 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.

5 Practical on Lambda Expression **25**

Write a Java program using Lambda Expression to print “Hello World”.

Write a Java program using Lambda Expression with single parameters.

Write a Java program using Lambda Expression with multiple parameters to add twonumbers.

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

a. Convert Fahrenheit to Celcius

b. Convert Kilometers to Miles.

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

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

6 Practical based on web application development using JSP **29**

Create a Telephone directory using JSP and store all the information within adatabase, so that later could be retrieved as per the requirement. Make your own assumptions.

Write a JSP page to display the Registration form (Make your own assumptions)

Write a JSP program to add, delete and display the records from StudentMaster (RollNo, Name, Semester, Course) table

Design loan calculator using JSP which accepts Period of Time (in years) and Principal Loan Amount. Display the payment amount for each loan and then list the loan balance and interest paid for each payment over the term of the loan for the following time period and interest rate:

a. 1 to 7 year at 5.35%

b. 8 to 15 year at 5.5%

c. 16 to 30 year at 5.75%

Write a program using JSP that displays a webpage consisting Application form for change of Study Center which can be filled by any student who wants to change his/her study center. Make necessary assumptions

Write a JSP program to add, delete and display the records from StudentMaste r(RollNo, Name, Semester, Course) table.

Write a JSP program that demonstrates the use of JSP declaration, scriptlet,directives, expression, header and footer.

7 **Practical program on Core Java Spring Framework** 63

Write a program to print "Hello World" using spring framework.

Write a program to demonstrate dependency injection via setter method.

Write a program to demonstrate dependency injection via setter method.

Write a program to demonstrate dependency injection via Constructor.

8 **Practical based on Spring Aspect Oriented Programming (AOP)** 71

Write a program to demonstrate Spring AOP –All

Write a program to demonstrate Spring AOP – before advice.

Write a program to demonstrate Spring AOP – after advice

Write a program to demonstrate Spring AOP – around advice.

Write a program to demonstrate Spring AOP – after returning advice.

Write a program to demonstrate Spring AOP – after throwing advice.

Write a program to demonstrate Spring AOP – pointcuts.

9 **Practical based on SPRING JDBC** 81

Write a program to insert, update and delete records from the given table.

Write a program to demonstrate PreparedStatement in Spring JdbcTemplate

Write a program in Spring JDBC to demonstrate ResultSetExtractor Interface

Write a program to demonstrate RowMapper interface to fetch the records from thedatabase.

10 **Practical based on SPRING boot and RESTFUL Web Services** 103

Write a program to create a simple Spring Boot application that prints a message.

Write a program to demonstrate RESTful Web Services with spring boot.

11 **Annexure - I: How to run Spring Program using Netbeans** 109

12 **Annexure - II: How to run Java Spring program using Eclipse** 133

13 **Annexure - III: Steps for Executing Spring AOP program using Eclipse** 145

14 **Annexure - IV: Steps to run JDBC Spring Programs** 179

15 **Annexure - IV: Spring Important** 209

Chapter 1 Practical on Java Collection Framework

Practical 1.1

```
        /* Understand the Generic class
* Write a Java Program to demonstrate a Generic Class
* Author: Vikram
* rollno:
* div:
        * */ Solution
public class GenericClassExample <T>{
private T dvariable;

public T getDvariable() {
return dvariable;
}

public void setDvariable(T dvariable) {
this.dvariable = dvariable;
}

public static void main(String []args)

{

GenericClassExample<Integer> intvar=new GenericClassExample<>();
intvar.setDvariable(10);

System.out.println("Variable value="+intvar.getDvariable());

GenericClassExample<String> strvar=new GenericClassExample<>();
strvar.setDvariable("Sukhiram");

System.out.println("Variable value="+strvar.getDvariable());

}

}
```

Output

```
Variable value = 10

Variable value = Sukhiram
```

Practical 1.2

```
/*

* Practical b1
* Write a program in java that will use Generic methods
* There are following kind of Generic Types
* T Type
* E Element
* k Key
* v Value
* N Number
*/
public class GenericMethod1 {
//print array is generic method
public static <E> void printArray(E[] inputArray)
{
//display array element
for(E element:inputArray)
{
System.out.printf("%s,",element);
}
System.out.println();
}
public static void main(String[] args) {
// TODO Auto-generated method stub Integer [] intArray= {1,2,3,4,5,6};
Double [] doubleArray= {1.45,2.35,4.63,8.56}; Character [] charArray= {'H','E','L','L','O'}; printArray(intArray);
printArray(doubleArray); printArray(charArray);
}
}
```

Output

```
1, 2, 3, 4, 5, 6,
1.45, 2.35, 4.63, 8.56,
H, E, L, L, O,
```

Practical 1.2a

```
/*
* Write a program in java that will use Generic methods
* There are following kind of Generic Types
* T Type
* E Element
* k Key
* v Value
* N Number
* understand the use of k,v types
* S,V
*/
class Pair<K,V>
{
private K key; private V value;
public Pair(K key, V value)
{
this.key = key;
this.value = value;
}
public K getKey()
{return key;}
public void setKey(K key)
{this.key = key;}
public V getValue()
{return value;}
public void setValue(V value)
{this.value = value;}
}
public class GenericMethod2 {
public static <K,V> boolean compare(Pair<K,V>p1,Pair<K,V>p2) {return p1.getKey().equals(p2.getKey()) &&
p1.getValue().equals(p2.getValue());
};
public static void main(String[] args) {
// TODO Auto-generated method stub Pair<Integer,String>p1=new Pair<>(1,”aaa”); Pair<Integer,String>p2=new Pair<>(2,”bbb”); boolean result=GenericMethod2.compare(p1, p2); System.out.println(result);
}
}
```

Output

```
false
```

Practical 1.3

Write a program to demonstrate unbound Wildcard in List Interface

```
import java.util.ArrayList; import java.util.Iterator; import java.util.List;
/*
* Write a program in java to understand the use
* of unbound wildcards ?
*/
public class unboundWildcardExample {
public static void printArray(List<?> mylist)
{
Iterator itr=mylist.iterator(); while(itr.hasNext())
{
System.out.println(itr.next());
}
System.out.println(“ “);
}
public static void main(String[] args) {

// TODO Auto-generated method stub

ArrayList <Integer>mynumberlist=new ArrayList<Integer>(); mynumberlist.add(1);

mynumberlist.add(2); mynumberlist.add(3); printArray(mynumberlist);

ArrayList <String>mynamelist=new ArrayList<String>(); mynamelist.add("Suresh");

mynamelist.add("Ramesh"); mynamelist.add("Jayesh"); mynamelist.add("Mahesh"); printArray(mynamelist);

}

}
```

Output

```
1

2

3

--------

Suresh

Ramesh

Jayesh

Mahesh

--------
```

Practical 1.3a

Write a program to demonstrate upper Wildcard in List Interface

```
import java.util.ArrayList; import java.util.Iterator; import java.util.List;

/*

* Write a program in java to understand the use

* of upperbound wildcards ? extends

*/

public class upperboundWildcardExample {

public static void sumofElements(List<? extends Number> numberlist)

{
```

```
double sum=0.0;

for (Number n : numberlist) sum += n.doubleValue(); System.out.println("Sum of all elements is="+sum);

}

public static void main(String []args) {

ArrayList <Integer>intlist=new ArrayList<Integer>(); intlist.add(1);

intlist.add(2); intlist.add(3); sumofElements(intlist);

ArrayList <Double>dblist=new ArrayList<Double>(); dblist.add(24.56);

dblist.add(21.24); dblist.add(124.16); dblist.add(121.14); sumofElements(dblist);

}

}
```

Output

```
Sum of all elements is=6.0

Sum of all elements is=291.09999999999997
```

Chapter 2 Practical on List Interface

Practical 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

```
import java.util.ArrayList;
import java.util.List;
/* Practical 2-1
* Author: Vikram
*
* 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.
*/
public class ListInterfaceExample1 {
public static void main(String[] args) {
// TODO Auto-generated method stub List<String> mylist=new ArrayList<String>(); mylist.add("Prakash");
mylist.add("Suresh"); mylist.add("Ramesh"); mylist.add("Jayesh");
for(String item: mylist) {System.out.println(item);
}
}
}
```

Output

```
Prakash Suresh Ramesh Jayesh
```

Practical 2.2

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

```
import java.util.ArrayList; import java.util.Collections; import java.util.Iterator; import java.util.List;
```

```
/*
* Write a Java program to create List containing list of items
* and use List Iterator interface to print items present in the
* list. Also print the list in reverse/ backward direction.
*/
public class ListInterfaceRevFor {
public static void main(String[] args) {
// TODO Auto-generated method stub
List<Integer> numberlist=new ArrayList<Integer>();
//Generating list of 10 numbers
for(int i=1;i<=10;i++) numberlist.add(i);
// printing the list using iterator Iterator<Integer> itr=numberlist.iterator(); while(itr.hasNext())
{
System.out.println(itr.next());
}
// printing the list using iterator in reverse order System.out.println("Reverse order"); Collections.reverse(numberlist);
Iterator<Integer> itr2=numberlist.iterator();
while(itr2.hasNext())
{
System.out.println(itr2.next());
}
}
}
```

Output

```
1
2
3
4
```

```
5
6
7
8
9
10
Reverse order 10
9
8
7
6
5
4
3
2
1
```

Chapter 3 Practical on Set Interface

Practical 3.1

Problem Statement

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.

Student.java

```
package com.hiraymca; public class Student {

int rollno; String name;

Double percentage;

public Student(int rollno, String name, Double percentage) {this.rollno = rollno;

this.name = name; this.percentage = percentage;

}

public Student()

{

}

public int getRollno() {return rollno;

}

public void setRollno(int rollno) {this.rollno = rollno;

}

public String getName() {return name;

}

public void setName(String name) {this.name = name;

}

public Double getPercentage() {return percentage;

}

public void setPercentage(Double percentage) {this.percentage = percentage;

}
```

```
}
setExample.java package com.hiraymca;
import java.util.ArrayList; import java.util.Collection;
import java.util.Collections;
import static java.util.Collections.list;
import java.util.Comparator;
import java.util.HashSet;
import java.util.Iterator;
import java.util.List;
import java.util.ListIterator;
import java.util.NavigableSet;
import java.util.Scanner;
import java.util.TreeSet;
/**
*
* @author VIKRAM
*/
public class setExample {
public static void main(String []args)
{
String ans=null;
int rn;
String sn;
double per;
Scanner sc=new Scanner(System.in);
Student s=new Student();
int choice;
//Creating set of students HashSet<Student>studentSet=new HashSet<Student>(); Iterator i;
```

```
//Creating Students
do
{
System.out.println(“Menu”); System.out.println(“1.Adding student”); System.out.println(“2.List all students”); System.out.println(“3.List in reverse direction”); System.out.println(“4.Exit”); System.out.print(“Enter your choice(1..4)”); choice=sc.nextInt();
int flag=0; switch(choice)
{
case 1:
System.out.print(“Enter rollno”); rn=sc.nextInt(); System.out.print(“Enter name”); sn=sc.next();
System.out.print(“Enter percentage”); per=sc.nextDouble();
s=new Student(rn,sn,per); studentSet.add(s);
break;
case 2:
System.out.println(“Rollno”+”\t”+”Name”+”\t”+”Percentage”); studentSet.forEach((s1) -> {System.out.println(s1.getRollno()+”\t”+s1.getName()+”\t”+s1.getPercentage());
});
break; case 3:
Comparator<Student> c=
Comparator.comparing(Student::getRollno,Comparator.reverseOrder()).thenComparing(Student::getName,Comparator.reverseOrder());
List<Student>list=new ArrayList<>(studentSet);
list.sort(c);
System.out.println(“Rollno”+”\t”+”Name”+”\t”+”Percentage”); for(Student s2:list)
{
System.out.println(s2.getRollno()+”\t”+s2.getName()+”\t”+s2.getPercentage());
}
break;
case 4:
System.exit(0); break;
}
//adding student in HashSet
System.out.print(“Do you wish to continue(y/n)”); ans=sc.next();
}while(ans.equals(“Y”)||ans.equals(“y”));
//reteriving the student from set
}
}
```

Practical 3.2

Problem Statement

1. Write a Java program using Set interface containing list of items and perform the following operations:

1. Add items in the set.

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

3. Remove items from the set

4. Search the specified item in the set

Student.java

```
package com.hiraymca;
public class Student {
int rollno; String name;
Double percentage;
public Student(int rollno, String name, Double percentage) {this.rollno = rollno;
this.name = name; this.percentage = percentage;
}
public Student()
{
}
public int getRollno() {return rollno;

}

public void setRollno(int rollno) {this.rollno = rollno;

}

public String getName() {return name;

}

public void setName(String name) {this.name = name;

}

public Double getPercentage() {return percentage;

}

public void setPercentage(Double percentage) {this.percentage = percentage;

}

}
```

setExample.java

```
package com.hiraymca;

import java.util.HashSet; import java.util.Iterator; import java.util.Scanner;

import java.util.function.Predicate;

/**

*

* @author VIKRAM

*/

public class setExample {

public static void main(String []args)

{

String ans=null; int rn;

String sn; double per;

Scanner sc=new Scanner(System.in); Student s=new Student();
```

```
int choice;

//Creating set of students HashSet<Student>studentSet=new HashSet<Student>(); Iterator i;

//Creating Students

do

{

System.out.println(“Menu”); System.out.println(“1.Adding student”); System.out.println(“2.Removing student”);
System.out.println(“3.Search student”); System.out.println(“4.List all students”); System.out.println(“5.Exit”);
System.out.print(“Enter your choice(1..4)”); choice=sc.nextInt();

int flag=0; switch(choice)

{

case 1:

System.out.print(“Enter rollno”); rn=sc.nextInt(); System.out.print(“Enter name”); sn=sc.next();

System.out.print(“Enter percentage”); per=sc.nextDouble();

s=new Student(rn,sn,per); studentSet.add(s);

break; case 2:

System.out.print(“Enter roll no to remove element”); rn=sc.nextInt();

i=studentSet.iterator(); while(i.hasNext())

{

s=(Student)i.next(); if(s.getRollno()==rn)

{

i. remove();

System.out.println(“Element is successfully removed”); break;

}

else

{

System.out.println(“Rollno=“+rn);

}

}

break; case 3:
```

```
System.out.println(“Enter the name of student”); sn=sc.next();
i=studentSet.iterator(); if(!i.hasNext())
{
System.out.println(“List is empty”);
}
while(i.hasNext())
{
s=(Student)i.next(); if(s.getName().equals(sn))
{
System.out.println(“Record found”);
System.out.println(s.getRollno()+” “+s.getName()+” “+s.getPercentage()); flag=1;
break;
}
if(flag!=1)
{
System.out.println(“Record not found search again”);
}
}
break;
case 4:
System.out.println(“Rollno”+”\t”+”Name”+”\t”+”Percentage”); studentSet.forEach((s1) -> {
System.out.println(s1.getRollno()+” “+s1.getName()+” “+s1.getPercentage());
});
break; case 5:
System.exit(0); break;
}
//adding student in HashSet
System.out.print(“Do you wish to continue(y/n)”); ans=sc.next();
```

```
}while(ans.equals(“Y”)||ans.equals(“y”));

//reteriving the student from set

}

}
```

Output

```
Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)1 Enter rollno 1

Enter name AAA

Enter percentage 99.98

Do you wish to continue(y/n)y Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)1

Enter rollno 2 Enter name BBB

Enter percentage 89.45

Do you wish to continue(y/n)y Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)1 Enter rollno 3

Enter name CCC

Enter percentage55.24

Do you wish to continue(y/n)y Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)3 Enter the name of student CCC

Record found 3 CCC 55.24

Do you wish to continue(y/n)y Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)4 Rollno Name Percentage 3 CCC 55.24

2 BBB 89.45
```

```
1 AAA 99.98

Do you wish to continue(y/n)y Menu

1.Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)2

Enter roll no to remove element2 Rollno=2

Element is successfully removed Do you wish to continue(y/n)y Menu

1. Adding student 2.Removing student 3.Search student 4.List all students 5.Exit

Enter your choice(1..4)4 Rollno Name Percentage 3 CCC 55.24

1 AAA 99.98

Do you wish to continue(y/n)n
```

Chapter 4 Practical on Map Interface

Practical 4

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

1. Add items in the map.
2. Remove items from the map
3. Search specific key from the map
4. Get value of the specified key
5. Insert map elements of one map in to other map.
6. Print all keys and values of the map.

Book.java

```
package com.hiraymca;
/**
*
* @author VIKRAM
*/
public class Book {private int id; private String name; private String author;
public Book()
{
}
public Book(int id, String name, String author) {this.id = id;
this.name = name;
this.author = author;
}
public int getId() {
return id;
}
public void setId(int id) {this.id = id;

}

public String getName() {return name;

}

public void setName(String name) {this.name = name;

}

public String getAuthor() {return author;

}

public void setAuthor(String author) {this.author = author;

}

}
```

mapExample.java

```
package com.hiraymca;

import java.util.HashMap; import java.util.Map;

import java.util.Scanner;

/**

*

* @author VIKRAM

*/

public class mapExample {

public static void main(String []args)

{

//Creating map of books

Map<Integer,Book>map=new HashMap<Integer,Book>();

//Creating books

Book b1=new Book(101,”Let us C”,”Yashwant Kanetkar”);

Book b2=new Book(102,”Data communication & Networking”,”Forouzan”); Book b3=new Book(103,”Operating
System”,”Achuyut Godbole”);
Book b=new Book();
//Adding books to map map.put(1,b1);
map.put(2,b2);
map.put(3,b3);
//Traversing the map
for(Map.Entry<Integer, Book> entry:map.entrySet()){int key=entry.getKey();
b=entry.getValue();
System.out.println(b.getId()+” “+b.getName()+” “+b.getAuthor());
}
//Removing element from map map.remove(2);
//Traversing the map after removing System.out.println();
System.out.println(“Traversing map after removing 2nd element”); for(Map.Entry<Integer, Book> entry:map.entrySet()){
int key=entry.getKey(); b=entry.getValue();
System.out.println(b.getId()+” “+b.getName()+” “+b.getAuthor());
}
//searching for specific entry int mykey;
Scanner sc=new Scanner(System.in); System.out.print(“Enter the key “); mykey=sc.nextInt();
for(Map.Entry<Integer, Book> entry:map.entrySet())
{int key=entry.getKey();
b=entry.getValue();
System.out.println(b.getId()+” “+b.getName()+” “+b.getAuthor());
}
}
}
```

Output

```
run:
101 Let us C Yashwant Kanetkar
102 Data communication & Networking Forouzan 103 Operating System Achuyut Godbole
```

```
Traversing map after removing 2nd element
101 Let us C Yashwant Kanetkar
103 Operating System Achuyut Godbole Enter the key 101
101 Let us C Yashwant Kanetkar
```

Chapter 5 Practical on Lambda Expression

Practical 5.1

```
/* @author VIKRAM
* Write a Java program using Lambda Expression with single parameters.
*/
interface HelloWorld {
void sayHello();
}
public class LambdaExpressionExample {public static void main(String args[]){
HelloWorld helloworld=()->{System.out.println(“Hello World”);}; helloworld.sayHello();
}
}
```

Output

```
Hello World
```

Practical 5.2

```
* Write a Java program using Lambda Expression with single parameters
interface findCube
{
int cube(int num);
}
public class Practical52 {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here findCube cubeobj=(int x)->{return x*x*x;};
int result=cubeobj.cube(2);
System.out.println(“Cube of Enter number is=“+result);
}
}
```

Output

```
Cube of Enter number is=8
```

Practical 5.3

```
* Write a Java program using Lambda Expression
* with multiple parameters to add two numbers.
*/
interface Calculator
{
int add(int x,int y);
}
public class Practical53 {
/**
* @param args the command line arguments
*/
public static void main(String[] args) {
// TODO code application logic here int num1=20,num2=10;
Calculator calc=(int n1,int n2)->{return num1+num2;};
int result=calc.add(num1, num2);
System.out.println(“Addition of two number is =“+result);
}
}
```

Output

```
Addition of two number is =30
```

Practical 5.6

```
interface Constr{

String addstr(String fname,String lname);

}

public class Practical56 {

/**

* @param args the command line arguments

*/

public static void main(String[] args) {

// TODO code application logic here String fname=“Hiray”;

String lname=“College”;

Constr ctr=(String f,String l)->{return f+” “+l;}; String fullstring=ctr.addstr(fname,lname); System.out.println(“My Full name is “+fullstring);

}

}
```

Output

```
My Full name is Hiray College
```