PART-A

1. Write a program to convert numbers into words using Enumerations with constructors, methods and instance variables.(INPUT RANGE-0 TO 99999) EX: 36 THIRTY SIX

=

New Project->Java->Java Application

name

```
import java.util.Scanner;

public class lab1 {
    private static final String[] units = {
        "", "ONE", "TWO", "THREE", "FOUR", "FIVE", "SIX", "SEVEN", "EIGHT",
"NINE"
    };
    private static final String[] teens = {
        "TEN", "ELEVEN", "TWELVE", "THIRTEEN", "FOURTEEN",
        "FIFTEEN", "SIXTEEN", "SEVENTEEN", "EIGHTEEN", "NINETEEN"
    };
    private static final String[] tens = {
        "", "", "TWENTY", "THIRTY", "FORTY", "FIFTY", "SIXTY", "SEVENTY",
"EIGHTY", "NINETY"
    };

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.println("Enter the number:");
        int number = sc.nextInt();
        System.out.println(convertToWords(number));
    }

    public static String convertToWords(int number) {
        if (number == 0) {
            return "ZERO";
        }
        if (number < 0 || number > 99999) {
            return "Number out of range";
        }

        String word = "";
        if (number >= 1000) {
```

```
            word += convertToWords(number / 1000) + " THOUSAND ";
            number %= 1000;
        }
        if (number >= 100) {
            word += units[number / 100] + " HUNDRED ";
            number %= 100;
        }
        if (number >= 20) {
            word += tens[number / 10] + " ";
            number %= 10;
        }
        if (number >= 10) {
            word += teens[number - 10] + " ";
        } else if (number > 0) {
            word += units[number] + " ";
        }

        return word.trim();
    }
}
```

OUTPUT :

run:

Enter the number:

2004

TWO THOUSAND FOUR

BUILD SUCCESSFUL (total time: 4 seconds)

2. Find the second maximum and second minimum in a set of numbers using auto boxing and unboxing.

=

New Project->Java->Java Application

name

```java
import java.util.*;
class PartA2
{
    public static void main(String args[])
    {
    System.out.println("Enter number of element:");
    Scanner sc=new Scanner(System.in);
    int n =sc.nextInt();
    Integer[] numbers=new Integer[n];
    System.out.println("Enter elements:");
    for(int i=0;i<n;i++)
    numbers[i]=sc.nextInt();
    Arrays.sort(numbers);
    System.out.println("second largest number is :"+ numbers[n-2]);
    System.out.println("second smallest number is :"+ numbers[1]);

    }
}
```

OUTPUT:

run:

Enter number of element:

5

Enter elements:

2

98

56

45

1

second largest number is :56

second smallest number is :2

BUILD SUCCESSFUL (total time: 13 seconds)

3. Write a menu driven program to create an Arraylist and perform the following operations

i) Adding elements

ii) Sorting elements

iii) Replace an element with another

iv) Removing an element

v) Displaying all the elements

vi) Adding an element between two elements

=

New Project->Java->Java Application

name

```
import java.util.*;

public class PartA3 {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        ArrayList<Integer> list = new ArrayList<>();

        while (true) {
            System.out.println("\nMENU");
            System.out.println("1. Add elements");
            System.out.println("2. Sort elements");
            System.out.println("3. Replace an element");
            System.out.println("4. Remove an element");
            System.out.println("5. Display elements");
            System.out.println("6. Add element between two elements");
            System.out.println("7. Exit");
            System.out.print("Enter your choice: ");
            int choice = sc.nextInt();

            switch (choice) {
                case 1:
                    System.out.print("Enter the number of elements to add: ");
                    int n = sc.nextInt();
```

```
                    System.out.println("Enter the elements:");
                    for (int i = 0; i < n; i++) {
                        list.add(sc.nextInt());
                    }
                    break;
                case 2:
                    Collections.sort(list);
                    System.out.println("Elements sorted.");
                    break;
                case 3:
                    System.out.print("Enter the element to replace: ");
                    int oldElement = sc.nextInt();
                    System.out.print("Enter the new element: ");
                    int newElement = sc.nextInt();
                    Collections.replaceAll(list, oldElement, newElement);
                    System.out.println("Element replaced.");
                    break;
                case 4:
                    System.out.print("Enter the element to remove: ");
                    int removeElement = sc.nextInt();
                    list.remove(Integer.valueOf(removeElement));
                    System.out.println("Element removed.");
                    break;
                case 5:
                    System.out.println("Elements: " + list);
                    break;
                case 6:
                    System.out.print("Enter the index to add the element: ");
                    int index = sc.nextInt();
                    System.out.print("Enter the element to add: ");
                    int addElement = sc.nextInt();
                    list.add(index, addElement);
                    System.out.println("Element added.");
                    break;
                case 7:
                    System.out.println("Exiting program.");
                    System.exit(0);
                default:
                    System.out.println("Invalid choice.");
            }
        }
    }
}
```

OUTPUT:

run:

MENU

1. Add elements
2. Sort elements
3. Replace an element
4. Remove an element
5. Display elements
6. Add element between two elements
7. Exit

Enter your choice: 1

Enter the number of elements to add: 5

Enter the elements:

7

3

9

2

4

MENU

1. Add elements
2. Sort elements
3. Replace an element
4. Remove an element
5. Display elements
6. Add element between two elements

7. Exit

Enter your choice: 2

Elements sorted.

MENU

1. Add elements

2. Sort elements

3. Replace an element

4. Remove an element

5. Display elements

6. Add element between two elements

7. Exit

Enter your choice: 3

Enter the element to replace: 2

Enter the new element: 8

Element replaced.

MENU

1. Add elements

2. Sort elements

3. Replace an element

4. Remove an element

5. Display elements

6. Add element between two elements

7. Exit

Enter your choice: 4

Enter the element to remove: 4

Element removed.

MENU

1. Add elements

2. Sort elements

3. Replace an element

4. Remove an element

5. Display elements

6. Add element between two elements

7. Exit

Enter your choice: 5

Elements: [8, 3, 7, 9]

MENU

1. Add elements

2. Sort elements

3. Replace an element

4. Remove an element

5. Display elements

6. Add element between two elements

7. Exit

Enter your choice: 6

Enter the index to add the element: 3

Enter the element to add: 10

Element added.

MENU

1. Add elements
2. Sort elements
3. Replace an element
4. Remove an element
5. Display elements
6. Add element between two elements
7. Exit

Enter your choice: 5

Elements: [8, 3, 7, 10, 9]

MENU

1. Add elements
2. Sort elements
3. Replace an element
4. Remove an element
5. Display elements
6. Add element between two elements
7. Exit

Enter your choice:7

Exiting program.

4. Write a java program to find words with even number of characters in a string, then swap the pair of characters in those words and also toggle the characters in a given string

EX: Good Morning everyone

Output: oGdo vereoyen

gOOD mORNING EVERYONE

=

New Project->Java->Java Application

name

```
 import java.util.Scanner;

public class PartA4 {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a string:");
        String inputString = scanner.nextLine();

        String modifiedString = modifyString(inputString);
        System.out.println("Modified string:");
        System.out.println(modifiedString);

        String originalWithToggledCase = toggleCase(inputString);
        System.out.println(originalWithToggledCase);
    }

    private static String modifyString(String input) {
        StringBuilder modifiedString = new StringBuilder();

        for (String word : input.split("\\s+")) {
            if (word.length() % 2 == 0) {
                char[] chars = word.toCharArray();
                for (int i = 0; i < chars.length - 1; i += 2) {
                    char temp = chars[i];
                    chars[i] = chars[i + 1];
                    chars[i + 1] = temp;
```

```
                }
                modifiedString.append(new String(chars)).append(" ");
            }
        }

        return toggleCase(modifiedString.toString());
    }

    private static String toggleCase(String input) {
        char[] chars = input.toCharArray();
        for (int i = 0; i < chars.length; i++) {
            char c = chars[i];
            if (Character.isUpperCase(c)) {
                chars[i] = Character.toLowerCase(c);
            } else if (Character.isLowerCase(c)) {
                chars[i] = Character.toUpperCase(c);
            }
        }
        return new String(chars);
    }
}
```

OUTPUT:

run:

Enter a string:

Good Moring everyone

Modified string:

OgDO OmIRGN VEREOYEN

gOOD mORING EVERYONE

BUILD SUCCESSFUL (total time: 16 seconds)

5. Write a Servlet program that accepts the age and name and displays if the user is eligible for voting or not

Output:

=

New project ->java web-> web application->name->index.html

Source Package -> right click->sevlet->CheckVoter->package:com-> check add box ->finish

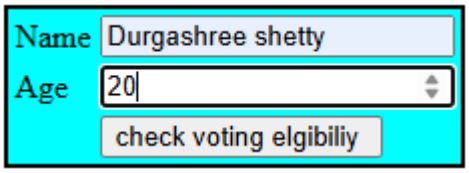
```
<html>
    <head>
        <title>Voting Eligibility Test</title>
        <meta charset="UTF-8">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <style>
            table{
                background-color: aqua;
                width: 200px;
                margin-top: 100px;
                margin-left: auto;
                margin-right: auto;
                border: solid 2px;

            }
        </style>
    </head>
    <body>
        <form method="POST" action="CheckVoter">
            <table>
                <tr>
                    <td>Name</td>
                    <td><input type="text" name="uname"></td>
                </tr>
                 <tr>
                    <td>Age</td>
```

```
                    <td><input type="number" name="age"></td>
                </tr>
                 <tr>
                    <td></td>
                    <td><input type="submit" name="uname" value="check voting
elgibiliy "></td >
                </tr>
            </table>
        </form>
    </body>
</html>
```

CheckVoter

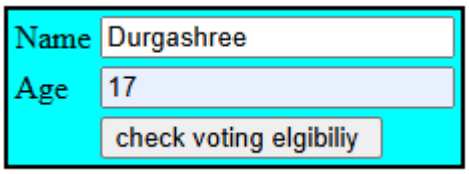
```
package com;

import java.io.IOException;
import java.io.PrintWriter;
import javax.servlet.ServletException;
import javax.servlet.http.HttpServlet;
import javax.servlet.http.HttpServletRequest;
import javax.servlet.http.HttpServletResponse;

public class CheckVoter extends HttpServlet {
    protected void doPost(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {
        response.setContentType("text/html");
        PrintWriter out = response.getWriter();

        String name = request.getParameter("uname");
        int age = Integer.parseInt(request.getParameter("age"));

        out.println("<html><body>");
        if (age >= 18) {
            out.println("<h4 style='color: green'>" + name + ", you are
eligible to vote.</h4>");
        } else {
            out.println("<h4 style='color: red'>" + name + ", you are not
eligible to vote.</h4>");
        }
        out.println("<a href='index.html'>Home</a>");
        out.println("</body></html>");
    }
}
```

OUTPUT:



Durgashree shetty, you are eligible to vote.

Home



Durgashree , you are not eligible to vote.

Home

6. Write a JSP program to print first 10 Fibonacci and 10 prime numbers.

=

New project ->java web-> web application->name->index.html

Web pages->right click->new->jsp->new or any name. ->finish.

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<!DOCTYPE html>
<html>
    <head>
        <meta http-equiv="Content-Type" content="text/html; charset=UTF-8">
        <title>Fibonacci and Prime</title>
    </head>
    <body>

        <h4>Fibonacci Series:</h4>
        <%
        int a=0, b=1, c, i;
        out.println(a +"   " +b +"   ");
        for(i=1; i<=10; i++){
```

```
            c=a+b;
            out.println(c+"    ");
            a=b;
            b=c;
        }

        %>


        <h4>Prime Numbers :</h4>
        <%
         int pn=2,count=1;
         boolean isprime;
         while(count<=10){
             isprime=true;
             for(i=2;i<pn/2;i++){
                 if(pn%i==0){
                     isprime=false;
                     break;
                 }
             }

             if(isprime){
                 out.println(pn+"   ");
                 count++;
          }
           pn++;

         }
        %>

    </body>
</html>
```

OUTPUT:

Fibonacci Series:

0 1 1 2 3 5 8 13 21 34 55 89

Prime Numbers :

2 3 4 5 7 11 13 17 19 23

7. Write a JSP Program to design a shopping cart to add items, remove item and to display items from the cart using Sessions

=

New project ->java web-> web application ->index.html

Web page->new->jsp->index.jsp->addItem.jsp->removeItem.jsp

Index.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>

<%
    java.util.HashMap<String, Integer> cart = (java.util.HashMap<String,
Integer>) session.getAttribute("cart");

    if(cart == null) {
        cart = new java.util.HashMap<String, Integer>();
        session.setAttribute("cart", cart);
    }
%>

<!DOCTYPE html>
<html>
<head>
    <title>Shopping Cart</title>
    <style>
        table {
            width: 50%;
            border-collapse: collapse;
            margin-top: 20px;
        }
        th, td {
            border: 1px solid black;
            padding: 8px;
            text-align: left;
        }
        th {
            background-color: #f2f2f2;
        }
    </style>
</head>
<body>

<h2>Add Item to Cart</h2>
```

```
<form action="addItem.jsp" method="post">
    <label>Item Name:</label>
    <input type="text" name="item" required>
    <input type="submit" value="Add to Cart">
</form>

<h2>Shopping Cart</h2>
<table>
    <tr>
        <th>Item</th>
        <th>Quantity</th>
        <th>Action</th>
    </tr>
    <%
        for(String item : cart.keySet()) {
    %>
    <tr>
        <td><%= item %></td>
        <td><%= cart.get(item) %></td>
        <td><a href="removeItem.jsp?item=<%= item %>">Remove</a></td>
    </tr>
    <%
        }
    %>
</table>

</body>
</html>
```

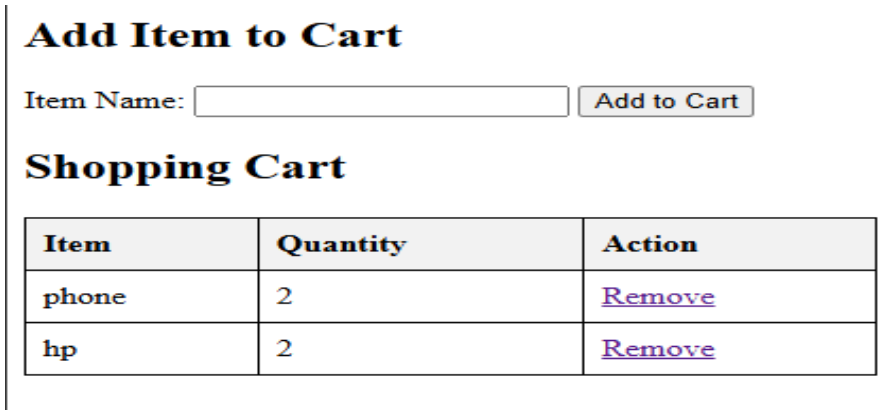
addItem.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    String item = request.getParameter("item");
    java.util.HashMap<String, Integer> cart = (java.util.HashMap<String,
Integer>) session.getAttribute("cart");

    if(cart.containsKey(item)) {
        cart.put(item, cart.get(item) + 1);
    } else {
        cart.put(item, 1);
    }

    response.sendRedirect("index.jsp");
%>
```

removeItem.jsp

```
<%@page contentType="text/html" pageEncoding="UTF-8"%>
<%
    String item = request.getParameter("item");
    java.util.HashMap<String, Integer> cart = (java.util.HashMap<String,
Integer>) session.getAttribute("cart");

    if(cart.containsKey(item)) {
        if(cart.get(item) > 1) {
            cart.put(item, cart.get(item) - 1);
        } else {
            cart.remove(item);
        }
    }

    response.sendRedirect("index.jsp");
%>
```

OUTPUT:



8. Write a java Servlet program to Download a file and display it on the screen(A link has to be provided in HTML, when the link is clicked corresponding file has to be displayed on screen).

=

New Project ->java web->web application -> index.html

Source Package ->right click->new->servlet->download.java

Index.html

```
<!DOCTYPE html>
<html>
<head>
    <meta charset="ISO-8859-1">
    <title>Welcome</title>
</head>
<body>
    <form action="download" method="get">
<h2>File Display</h2>
<h3>Download and Display File.</h3>
<input type="submit" value="Download" />
    </form>
</body>
</html>
```

Download.java

```
package file;
import java.io.*;
import javax.servlet.ServletException;
import javax.servlet.annotation.WebServlet;
import javax.servlet.http.*;

@WebServlet("/download")
public class Download extends HttpServlet {

    public void doGet(HttpServletRequest request, HttpServletResponse
response)
            throws ServletException, IOException {

        PrintWriter out = response.getWriter();
        String textFileName = "newfile.txt";
        String textFilePath = "C:\\Users\\durga\\Desktop\\Hello.txt";

        response.setContentType("text/plain");
        response.setHeader("Content-Disposition", "attachment; filename=\"" +
textFileName + "\"");

        FileInputStream inputStream = new FileInputStream(textFilePath);

        int in;
```

```
        while ((in = inputStream.read()) != -1) {
            out.write(in);
        }
        inputStream.close();
        out.close();
    }
}
```

OUTPUT: