JAVA AND J2EE LAB
(15CSL59A)
MANUAL

FOR 5th SEM
Computer Science and Engineering

(2016-2017)

Prepared By

Mr. Santosh and Mr.Prashanth Kumar A
Asst.Professor

COMPUTER SCIENCE AND ENGINEERING
DEPARTMENT
CANARA ENGINEERING COLLEGE
BEJANAPADAVU-574219

TABLE OF CONTENT

Program 1: Program on Simple example of java enum

```
 class EnumExample1

        {

        public enum Season { WINTER, SPRING, SUMMER, FALL }

         public static void main(String[] args)

                 {

                                  for (Season s : Season.values())

                                 System.out.println(s);

                 }

        }
```

OUTPUT:

```
 WINTER
 SPRING
 SUMMER
 FALL
```

Program 2: Program on specifying initial value to the enum constants.

```
class EnumExample4

{

        enum Season

                {

                WINTER(5), SPRING(10), SUMMER(15), FALL(20);

                private int value;

                private Season(int value)

                        {

                        this.value=value;

                        }

                }

        public static void main(String args[])
```

```
                        {

                                for (Season s : Season.values())

                                System.out.println(s+" "+s.value);

                        }

}
Output:
WINTER 5
SPRING 10
SUMMER 15
FALL 20
```

Program 3: Simple example on Annotation

```
import java.lang.annotation.*;

import java.lang.reflect.*;

import java.lang.annotation.*;

@Retention(RetentionPolicy.RUNTIME)

@interface MyINF

{

        //String str() ;

        int value();

}

class annu

{

                        @MyINF(100)

                        public static void myMeth()

                        {

                        annu ob = new annu();

                        try {

                                Class<?> c = ob.getClass();

                                Method m = c.getMethod("myMeth");
```

```
            MyINF anno = m.getAnnotation(MyINF.class);
            System.out.println(anno.value());
        }
        catch (NoSuchMethodException exc) {
        System.out.println("Method Not Found.");
        }
      }
    public static void main(String args[]) {
    myMeth();
    }
}
```

Output:

```
100
```

Program 4: Program on all Wrapper class

```
class Wrap
{
    public static void main(String args[])
    {
        Character c=new Character('@'); // character type
        char c1=c.charValue();
        System.out.println("Character wrapper class"+c1);
        Boolean b=new Boolean(true);
        boolean b1=b.booleanValue();
        System.out.println("Boolean wrapper class"+b1);
        Integer i1 = new Integer(100);  // integre type
        int i = i1.intValue();
        System.out.println("Integer wrapper class"+i); // displays 100 100
        Float f1 = new Float(12.5);  // Float type
        float f = f1.floatValue();
        System.out.println("Float wrapper class"+f);
    }
}
```

Output:

```
Character wrapper class@
Boolean wrapper classtrue
Integer wrapper class100
Float wrapper class12.5
```

Program 5: Simple program for autoboxing and autoUnboxing

```
class auto
{
    public static void main(String[] args)
    {
        Integer iob = 100; //Auto-boxing of int i.e converting primitive data type
         int to a Wrapper class Integer
        int i = iob;   //Auto-unboxing of Integer i.e converting Wrapper class
        Integer to a primitve type int
        System.out.println("integer type="+i+" "+iob);

        Character cob = 'a';  //Auto-boxing of char i.e converting primitive data
        type char to a Wrapper class Character
        char ch = cob;  //Auto-unboxing of Character i.e converting Wrapper class
        Character to a primitive type char
```

```
                        System.out.println("character type="+cob+" "+ch);
            }
}
```

Output:

```
integer type=100 100
character type=a a
```

Program 6: Simple program on ArrayList collection class.

```
public class A
{
                        public static void main(String args[])
                        {
                                    ArrayList<String> al = new ArrayList<String> ();
                                    System.out.println("Initial size of al: " + al.size());
                                     al.add("C");
                                     al.add("A");
                                    al.add("E");
                                    al.add("B");
                                    al.add("D");
                                    al.add("F");
                                    al.add(1, "A2");
                                    System.out.println("Size of al after additions: " + al.size());
                                    System.out.println("Contents of al: " + al);
                                    al.remove("F");
                                     al.remove(2);
                                    System.out.println("Size of al after deletions: " + al.size());
                                    System.out.println("Contents of al: " + al);
```

```
                }

}
```

Output:

```
Initial size of al: 0

Size of al after additions: 7

Contents of al: [C, A2, A, E, B, D, F]

Size of al after deletions: 5

Contents of al: [C, A2, E, B, D]
```

Program 7: Simple program on LinkedList collection class.

```
public class A {

  public static void main(String args[]) {

    LinkedList<String> l = new LinkedList<String>();

    l.add("F");

    l.add("B");

    l.add("D");

    l.add("E");

    l.add("C");

    l.addLast("Z");

    l.addFirst("A");

    l.add(1, "A2");

    System.out.println("Original contents of l: " + l);

    l.remove("F");

    l.remove(2);

    System.out.println("Contents of l after deletion: " + l);

    l.removeFirst();

    l.removeLast();
    System.out.println("l after deleting first and last: " + l);
    Object val = l.get(2);
    l.set(2, (String) val);
    System.out.println("l after change: " + l);
  }
}
```

Output:

```
Original contents of list: A, A2, F, B, D, E, C, Z
Contents of list after deletion: A, A2, D, E, C, Z
list after deleting first and last: A2, D, E, C
list after change: A2, D, E  C
```

Program 8: Simple program on HashSet collection class.

```
public class A {
  public static void main(String args[]) {
    HashSet<String>  hs = new HashSet<String> ();
    hs.add("B");
    hs.add("A");
    hs.add("D");
    hs.add("E");
    hs.add("C");
    hs.add("F");
    System.out.println(“Elements in hashset “+hs);
  }
}
```

output:

```
Elements in hashset : A, B, C, D, E, F
```

Program 9: Program on Storing User Defined Classes in Collections

```
class A
{
        String name;
        String usn;
        String Branch;
        int p_No;
        A(String name,String usn,String Branch,int p_no)
        {
                this.name=name;
                this.usn=usn;
                this.Branch=Branch;
                p_No=p_no;
        }
}
class LinkedListClass
{
        public static void main(String ar[])
        {
                LinkedList<A> l=new LinkedList<A>();
                l.add(new A(“Amar”,”123”,”CSE”,99999999));
                l.add(new A(“Annu”,”456”,”CSE”,9900000));
                l.add(new A(“Raj”,”789”,”CSE”,99999900));
                System.out.println(l);
        }
}
```

Output:

```
Amar 123 CSE 99999999

Annu 456 CSE 9900000

Raj    789 CSE 99999900
```

Program 10: Program on different types of string class constructor.

```
public class StrCon {
        public static void main(String[] args) {
        String a=new String();
        System.out.println("Empty String"+a);
        char ch[]={'a','b','c','d'};
        String b=new String(ch);
        System.out.println("String with one argument as Char="+b);
        String c=new String(ch,1,3);
        System.out.println("String with Three argument as Char="+c);
        String d=new String(b);
        System.out.println("String with String object="+d);
        byte e[]={65,66,67,68,69};
        String f=new String(e);
        System.out.println("byte to String="+e);
        String g=new String(e,1,3);
        System.out.println("byte to string for subbyte="+g);
        StringBuffer h=new StringBuffer("hello");
        String i=new String(h);
        System.out.println("StringBuffer to String="+i);
        StringBuilder j=new StringBuilder("welcome");
        String k=new String(j);
        System.out.println("StringBuilder to Stirng="+k);
```

```
                int l[]={66,67,68,69,70};

                String m=new String(l,1,3);

                System.out.println("codepoint to String="+m);

                }

}
```

Output:

```
Empty String

String with one argument as Char=abcd

String with Three argument as Char=bcd

String with String object=abcd

byte to String=[B@19821f

byte to string for subbyte=BCD

StringBuffer to String=hello

StringBuilder to Stirng=welcome

codepoint to String=CDE
```

Program 11: Program on different type of String methods.

```
public class CO {

public static void main(String[] args) {

                String a="hello";

                int b=10;

                char c='a';

                System.out.println("STring to String as a object="+String.valueOf(a));

                System.out.println("Int to String as a object="+String.valueOf(b));

                System.out.println("char to String as a object="+String.valueOf(c));

                Integer a1=10;

                System.out.println("Integer to string"+a1.toString());
```

```
            String a1="hello";
            char c1=a1.charAt(1);
            System.out.println("charAt="+c1);
            char ch[]=new char[2];
            a1.getChars(1, 3, ch, 0);
            System.out.println(ch);
            byte b1[]=a.getBytes();
            System.out.println(b1);
            char ch1[]=a1.toCharArray();
            System.out.println(ch1);
        }
}
```

Output:

```
String to String as a object=hello
Int to String as a object=10
char to String as a object=a
Integer to string=10
charAt=e
el
[B@19821f
hello
```

Program 12: Program on different type of StringBuffer class methods.

```
class CO
{
  public static void main(String args[])
  {
    StringBuffer sb=new StringBuffer("Canara Enhioneering college    ");
```

```
        System.out.println("Unicode = " + sb.codePointAt(5));

        System.out.println("Length " + sb.codePointAt(5));

        System.out.println("Substring Index = " + sb.indexOf("Can"));

        System.out.println("Substring Index = " + sb.lastIndexOf("can"));

        System.out.println("Reverse = " + sb.reverse());

    }

}
```

Output:

```
Unicode = 97

Length 97

Substring Index = 0

Substring Index = -1

Reverse =     egelloc gnireenoihnE aranaC
```

Program 13: To display greeting message on the browser Hello UserName How Are You accept username from the client.

```
import java.io.*;

import javax.servlet.ServletException;

import javax.servlet.http*;

public class A extends GenericServlet

{

            public void service(ServletRequest req ,ServletResponse res)throws

            ServletException,IOException

            {

                        res.setContentType(“text/html”) ;

                        PrintWriter out=res.getWriter();

                        String msg=req.getParameter("t1");
```

```
                    out.println("hello"+msg+"how are you");
          }
}
```

HTML code

```
<html>
<body>
<form action=http://localhost:8080/A >
<input type="text box" name="t1" value=" ">
<input type="submit" name="submit">
</form>
</body>
</html>
```

OUTPUT:

```
hello CEC how are you
```

program 14: To create and read the cookie for the given cookie name as "EMPID" and its value as"AN2356".

```
public class A extends GenericServlet
{
          public void service(ServletRequest req ,ServletResponse res)throws
          ServletException,IOException
          {
                    res.setContentType("text/html") ;
                    PrintWriter out=res.getWriter();
                    /* creating cookie object */
                     Cookie c=new Cookie("EMPID","AN2356");
                    res.addCookie(c);//adding cookie in the response
```

```
        /*reading cookies */
        Cookie c[]=req.getCookies();
        for(int i=0;i<c.length;i++)
        {
            String Name=c[i].getName();
            String value= c[i].getValue();
            out.println(“name=”+Name);
            out.println(“Value=”+Value);
        }
    }
}
```

Output:

```
name= EMPID
Value=AN2356
```

Program 15:program to retrieve the data from the database

```
import java.sql.*;
class A
{
    A()
    {
        try
        {
            Class.forName(“sun.jdbc.odbc.JdbcOdbcDriver”);
            Connection c=DriverManager.getConnection(“JDBC:ODBC:CSB”);
            Statement s=c.createStatement();
            ResultSet r=s.executeQuery(“Select *from emp”);
            System.out.println(“Name /t Usn”);
            while(r.next())
            {
                String name=r.getString(1);
                String usn=r.getString(2);
                System.out.println(name);
                System.out.println(usn);
            }
            c.close();
        }
        catch(Exception e)
        {
            S.o.p(e);
        }
    }
    public stataic void main(String ar[])
    {
        A a1=new A();
    }
}
```

OUTPUT:

```
Name            Usn
abc                     123
efg                     456
cba                     789
```