

Exercise-14  
Packages and Access control

Q1. Define package?

Ans: A package is nothing but a physical folder structure that contains a group of related classes, interfaces and sub packages according to their functionality. It provides a convenient way to organize your word java language has various built in packages.

Q2. Write the keywords to create and import a package?

Ans: The package, import keyword is used to create and import a package in java.

Q3. Write the advantages of using packages?

Ans: Advantages of using packages in Java

Programmers can define their own packages to bundle a group of classes/interfaces, etc.

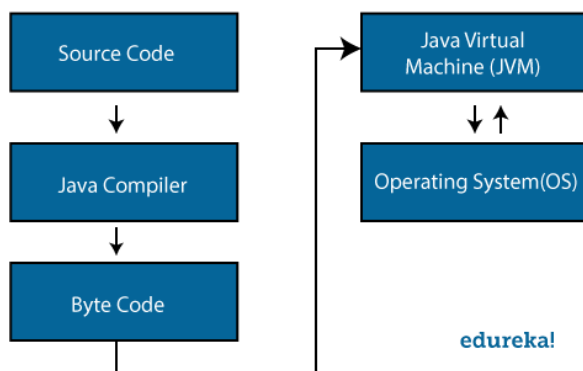
It is a good practice to group related classes implemented by you so that a programmer can easily determine that the classes, interfaces, enumerations, and annotations are related.

Q4. Write the significance of classpath?

Ans: Classpath is an environment variable (i.e., global variables of the operating system available to all the processes) needed for the Java compiler and runtime to locate the Java packages/classes used in a Java program.

Q5. What are the different places java compiler looks out for classes. Mention them in order?

Ans:



Q6. In `import java.awt.event.*`; statement classes and interfaces are imported from which Package?

Ans: In `import java.awt.event.*`; statement classes and interfaces are imported from which java.awt package.

Q7. What is the affect of importing all the classes and interfaces from a package than importing a single required class?

Ans: The affect of importing all the classes and interfaces from a package than importing a single Required class is it loads all the classes and interfaces by that all load gets increased on the program. So that the time complexity and space complexity increases.

Q8. Write the command to compile a java program with a package statement?

Ans: The javac command reads source files that contain module, package and type declarations written in the Java programming language, and compiles them into class files that run on the Java Virtual Machine.

Q9. List the different access control keywords?

Ans: Java provides access control through three keywords - private, protected, public and default.

Q11. Name the access control for which there is no keyword?

Ans: Default is the access control for which there is no keyword.

### Programs

Q1. Write a java program with a class A in a package p1. Write the commands and procedure to compile and execute the java program.

#### PROCEDURE:

Save the program

Compile it by `javac -d . filename.java`

Execute it by `packagename.filename`

#### PROGRAM:

```
package p1; class A {  
    public static void main(String s[])  
    {  
        System.out.println("IAM CLASS A FROM PACKAGE P1");  
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");  
    }  
}
```

#### OUTPUT:

```
A:\Java Records>javac -d . p1.java
```

```
A:\Java Records>java p1.A  
IAM CLASS A FROM PACKAGE P1  
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q2. Write a java program with a class A in a package p1. Access the class A in package p1 in class B of the package p2. Write the commands and procedure to compile and execute the java program.

PROGRAM 1:

```
package p1;
public class A1
{
void show()
{
System.out.println("IAM CLASS A1 FROM PACKAGE P1");
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

PROGRAM 2:

```
package p1;
class B1
{
public static void main(String s[])
{
A1 ref=new A1();
ref.show();
}
}
```

OUTPUT:

```
A:\Java Records>javac -d . B1.java
```

```
A:\Java Records>javac -d . A1.java
```

```
A:\Java Records>java p1.B1
```

```
IAM CLASS A1 FROM PACKAGE P1
```

```
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q3. Write a java program with a class A in a package p1. Access the class A in package p1 in class A of the package p2. Write the commands and procedure to compile and execute the java program.

PROGRAM 1:

```
package p1;
public class S1
{
    public void display7()
    {
        System.out.println("IAM CLASS A2 FROM PACKAGE P1");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

PROGRAM 2:

```
package p2; import p1.*;
public class S2
{
    public static void main(String s[])
    {
        S1 ref=new S1();
        ref.display7();
    }
}
```

OUTPUT:

```
A:\Java Records>javac -d . B1.java
```

```
A:\Java Records>javac -d . A1.java
```

```
A:\Java Records>java p1.B1
```

```
IAM CLASS A1 FROM PACKAGE P1
```

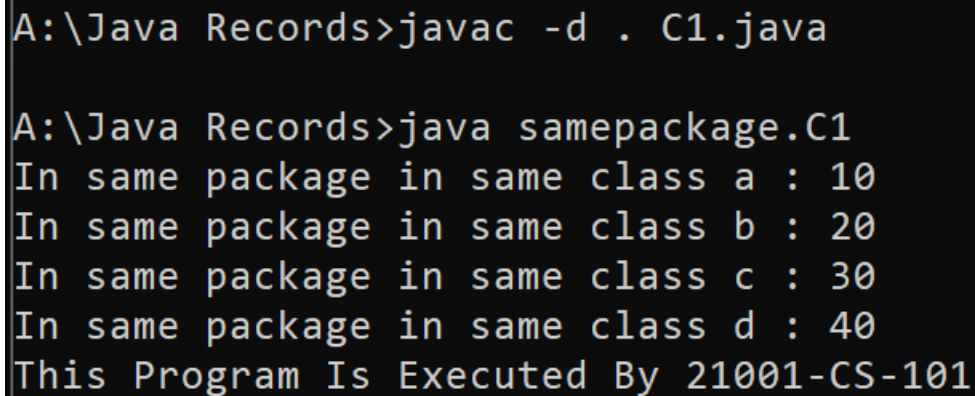
```
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q4. Write java programs to illustrate all access control specifiers.

PROGRAM 1:

```
package samepackage;
public class C1
{
    public int a = 10;
    int b = 20;
    protected int c = 30;
    private int d = 40;
    public void display()
    {
        System.out.println("In same package in same class a : "+a);
        System.out.println("In same package in same class b : "+b);
        System.out.println("In same package in same class c : "+c);
        System.out.println("In same package in same class d : "+d);
    }
    public static void main(String s[])
    {
        C1 obj = new C1();
        obj.display();
        System.out.println("This Program Is Executed By 21001-CS-101");
    }
}
```

OUTPUT:

A screenshot of a terminal window with a black background and white text. It shows the compilation and execution of a Java program. The first command is 'A:\Java Records>javac -d . C1.java'. The second command is 'A:\Java Records>java samepackage.C1'. The output consists of five lines: 'In same package in same class a : 10', 'In same package in same class b : 20', 'In same package in same class c : 30', 'In same package in same class d : 40', and 'This Program Is Executed By 21001-CS-101'.

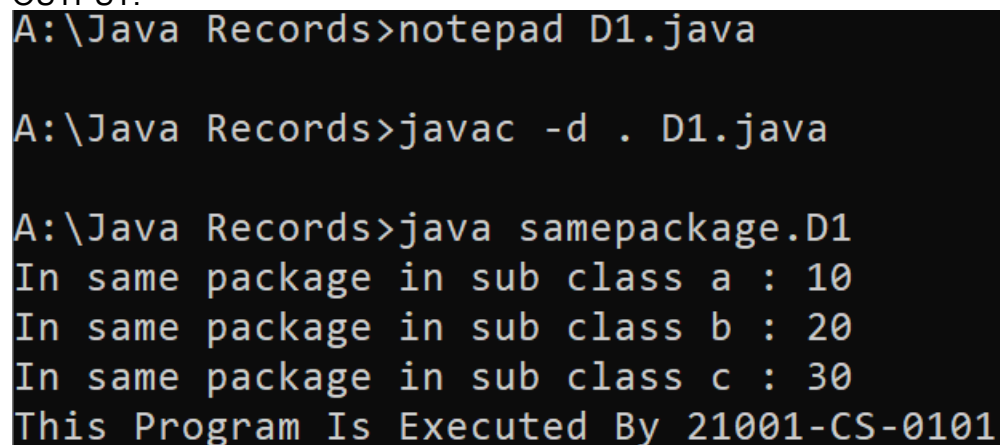
```
A:\Java Records>javac -d . C1.java

A:\Java Records>java samepackage.C1
In same package in same class a : 10
In same package in same class b : 20
In same package in same class c : 30
In same package in same class d : 40
This Program Is Executed By 21001-CS-101
```

## PROGRAM 2:

```
package samepackage;
public class D1 extends C1
{
    public void displayD1()
    {
        System.out.println("In same package in sub class a : "+a);
        System.out.println("In same package in sub class b : "+b);
        System.out.println("In same package in sub class c : "+c);
        //System.out.println("In same package in sub class d : "+d);
        // Private members cannot be accessed in sub classes of same package.
    }
    public static void main(String s[])
    {
        D1 obj1 = new D1();    obj1.displayD1();
        System.out.println("This Program Is Executed By 21001-CS-0101");
    }
}
```

## OUTPUT:



```
A:\Java Records>notepad D1.java

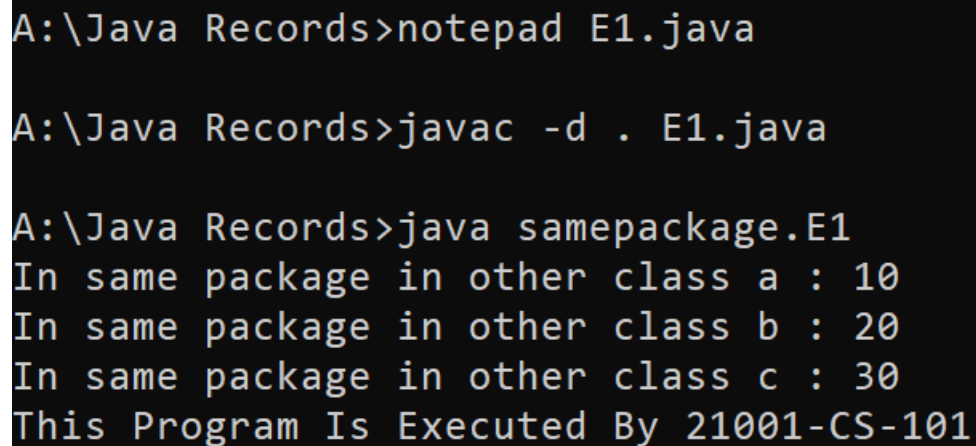
A:\Java Records>javac -d . D1.java

A:\Java Records>java samepackage.D1
In same package in sub class a : 10
In same package in sub class b : 20
In same package in sub class c : 30
This Program Is Executed By 21001-CS-0101
```

## PROGRAM 3:

```
package samepackage;
class E1
{
    public void displayE1()
    {
        C1 obj3 = new C1();
        System.out.println("In same package in other class a : "+obj3.a);
        System.out.println("In same package in other class b : "+obj3.b);
        System.out.println("In same package in other class c : "+obj3.c);
        //System.out.println("In same package in other class d : "+obj3.d);
        // Private members cannot be accessed in other classes of same package.
    }
    public static void main(String s[])
    {
        E1 a = new E1();
        a.displayE1();
        System.out.println("This Program Is Executed By 21001-CS-101");
    }
}
```

## OUTPUT:



```
A:\Java Records>notepad E1.java

A:\Java Records>javac -d . E1.java

A:\Java Records>java samepackage.E1
In same package in other class a : 10
In same package in other class b : 20
In same package in other class c : 30
This Program Is Executed By 21001-CS-101
```

## PROGRAM 4:

```
package otherpackage;
import samepackage.*;
public class F1 extends C1
{
    public void displayF1()
    {
        System.out.println("In same package in same class a : "+a);
        // System.out.println("In same package in same class b : "+b);
        // default members cannot be accessed in sub classes of other package.
        System.out.println("In same package in same class c : "+c);
        // System.out.println("In same package in same class d : "+d);
        // Private members cannot be accessed in sub classes of other package.
    }
    public static void main(String s[])
    {
        F1 obj4 = new F1();
        obj4.displayF1();
        System.out.println("This Program Is Executed By 21001-CS-101");
    }
}
```

## OUTPUT:

```
A:\Java Records>javac -d . F1.java

A:\Java Records>java otherpackage.F1
In same package in same class a : 10
In same package in same class c : 30
This Program Is Executed By 21001-CS-101
```



## PROGRAM 5:

```
package otherpackage;
import samepackage.*;
public class E1
{
    public void displayE1()
    {
        A1 obj5 = new A1();
        System.out.println("In same package in same class a : "+obj5.a);
        // System.out.println("In same package in same class b : "+obj5.b);
        // default members cannot be accessed in other classes of other package.
        // System.out.println("In same package in same class c : "+obj5.c);
        // Protected members cannot be accessed in other classes of other package.
        // System.out.println("In same package in same class d : "+obj5.d);
        // Private members cannot be accessed in other classes of other package.
    }
    public static void main(String s[])
    {
        E1 obj6 = new E1();
        obj6.displayE1();
        System.out.println("This Program Is Executed By 21001-CS-101");
    }
}
```

## OUTPUT:

```
A:\Java Records>notepad E1.java

A:\Java Records>javac -d . E1.java

A:\Java Records>java samepackage.E1
In same package in other class a : 10
This Program Is Executed By 21001-CS-101
```

## EXERCISE-15

### Utility Classes

Q1. What is the name of java utility package?

Ans: The java. util package is the name of Java utility package

Q2. What is Collections Framework in Java?

Ans: The Collections Framework in Java is a standard library feature that provides the implementation for a set

of commonly used data structures

Q3. List five classes from java.util package?

AbstractCollection

AbstractList

Arrays

Abstractset

Calendar

LinkedList

Q4. What is the use ArrayList class?

Ans: The ArrayList class extends AbstractList and implements the List interface. ArrayList supports dynamic arrays that can grow as needed. Standard Java arrays are of a fixed length. After arrays are created, they cannot grow or shrink, which means that you must know in advance how many elements an array will hold.

Q5. What is the difference between Arrays and ArrayList? Ans:

Array	ArrayList
Array is a fixed length data structure that is you can not change length of Array once created in Java	ArrayList is a variable length Collection class that is ArrayList re-size itself when gets full depending upon capacity and load factor
We can not use Generics along with Array, as Array instance knows about what kind of type it can hold and throws ArrayStoreException, if you try to store type which is not convertible into type of Array	ArrayList allows you to use Generics to ensure type-safety
Array can contain both primitive data types as well as objects of a class depending on the definition of the array	ArrayList only supports object entries, not the primitive data types
Java provides add() method to insert element into ArrayList	We use assignment operator to store element into Array
Its mandatory to provide size of Array	We can create ArrayList with default size of 10
Each array object has the length variable which returns the length of the array	Length of the ArrayList is provided by the size() method
Array can be multi dimensional	ArrayList is always single dimensional
Use Array, when you know that the size is fixed	Use ArrayList, when you don't know about the size of elements

Q6. List ten Methods of ArrayList class.?

Ans:

Java ArrayList add()

Java ArrayList addAll()

Java ArrayList clear()

Java ArrayList clone()

Java ArrayList contains()

Java ArrayList get()

Java ArrayList indexOf()  
Java ArrayList removeAll()  
Java ArrayList subList()  
Java ArrayList set()

Q7. Define LinkedList.

Ans: A linear data structure used to store the elements in contiguous locations is called a Linked List

Q8. Define queue?

Ans: A queue is an object that represents a data structure designed to have the element inserted at the end of the queue, and the element removed from the beginning of the queue. Java. Util. Queue contains multiple elements before the process. The order of elements of the queue in Java is FIFO (first-in-firstout).

Q9. Define double ended queue?

Ans: A double-ended queue is a special type of data in the field of computer programming. In this abstract data type, elements can be added from both the front and the back of the queue. Due to this property, it is also known as a head-tail linked list.

Q10. Define stack?

Ans: Stack in Java usually means the class from Collection Framework that implements the List interface. It works on the principle of the Stack data structure, which is used to organize one of the types of memory

Q11. Write the applications of stack.?

Ans: Application of the Stack

A Stack can be used for evaluating expressions consisting of operands and operators.

Stacks can be used for Backtracking, i.e., to check parenthesis matching in an expression.

It can also be used to convert one form of expression to another form.

It can be used for systematic Memory Management.

Q12. Write the applications of queue.?

Ans: Application of Queue

Managing requests on a single shared resource such as CPU scheduling and disk scheduling.

Handling hardware or real-time systems interrupts.

Handling website traffic.

Routers and switches in networking.

Maintaining the playlist in media players.

Q13. Write the applications of double ended queue.?

Ans Applications of deque

The A-steal algorithm implements task scheduling for multiple processors (multiprocessor scheduling).

The processor gets the first element from the double ended queue.

When one of the processors completes execution of its own thread, it can steal a thread from other processors

Q14. What is the use of seed in random class?

Ans: Seed function is used to save the state of a random function, so that it can generate same random numbers on multiple executions of the code on the same machine or on different machines (for a specific seed value). The seed value is the previous value number generated by the generator.

Q15. What is the use of StringTokenizer class?

Ans: The string tokenizer class allows an application to break a string into tokens. The tokenization method is much simpler than the one used by the StreamTokenizer class. The StringTokenizer methods do not distinguish among identifiers, numbers, and quoted strings, nor do they recognize and skip comments.

Q16. What are the default separators used in StringTokenizer class.?

Ans: The tokenizer uses the default delimiter set, which is " \t\n\r\f" : the space character, the tab character, the newline character, the carriage-return character, and the form-feed character.

Q17. List two methods of StringTokenizer class.?

Ans: Methods of StringTokenizer class are as follows:

hasMoreToken.

nextToken.

countTokens.

nextElement.

hasMoreElements.

## PROGRAMS

Q1. Write a java program using 10 methods of ArrayList class.

PROGRAM:

```
import java.util.ArrayList;
import java.util.*;
class ArrayListex
{
public static void main(String s[])
{
ArrayList<Integer> a=new ArrayList<Integer>(); a.add(1);
a.add(2);
a.add(3);
ArrayList<Integer> b=new ArrayList<Integer>(); b.add(1);
System.out.println("ELEMENTS of A =" +a);
System.out.println("IS EMPTY="+a.isEmpty());
System.out.println("REMOVED ELEMENT="+a.remove(1));
System.out.println("SIZE="+a.size());
System.out.println("HASHCODE="+a.hashCode());
System.out.println("INDEX OF 1="+a.indexOf(1)); b.addAll(a);
System.out.println("ELEMENTS of b="+b); b.removeAll(a);
System.out.println("REMAINING ELEMENTS of b="+b);
System.out.println("POSITION OF 1="+a.get(1)); a.clear();
System.out.println("REMANINING IN A="+a);
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

OUTPUT:

```
A:\Java Records>javac ArrayListex.java

A:\Java Records>java ArrayListex
ELEMENTS of A =[1, 2, 3]
IS EMPTY=false
REMOVED ELEMENT=2
SIZE=2
HASHCODE=995
INDEX OF 1=0
ELEMENTS of b=[1, 1, 3]
REMAINING ELEMENTS of b=[]
POSITION OF 1=3
REMANINING IN A=[]
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q2. Write a java program to implement a queue using a linked list.

PROGRAM:

```
import java.util.LinkedList;
import java.util.Scanner;
class Queues
{
    public static void main(String s[])
    {
        int in=0;
        LinkedList<Integer> a=new LinkedList<Integer>();
        Scanner sc=new Scanner(System.in);

        while(true){
            System.out.println("1 TO ADD=");
            System.out.println("2 TO DELETE=");
            System.out.println("3 TO PRINT=");
            System.out.println("4 TO EXIT=");
            int choice,s1;
            System.out.println("Enter your choice=");
            choice=sc.nextInt();
            String k=sc.nextLine();
            switch(choice)
            {
                case 1:
                    System.out.println("enter the value : ");
                    s1=sc.nextInt();
                    a.add(s1);
                    break;
                case 2:
                    a.remove(0);
                case 3:
                    System.out.println(a);
                    break;
                case 4:
                    System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
                    System.exit(1);
            }
        }
    }
}
```

## OUTPUT:

```
D:\sujith>javac Queues.java
```

```
D:\sujith>java Queues
```

```
1 TO ADD=
```

```
2 TO DELETE=
```

```
3 TO PRINT=
```

```
4 TO EXIT=
```

```
Enter your choice=
```

```
1
```

```
enter the value :
```

```
1
```

```
1 TO ADD=
```

```
2 TO DELETE=
```

```
3 TO PRINT=
```

```
4 TO EXIT=
```

```
Enter your choice=
```

```
1
```

```
enter the value :
```

```
2
```

```
1 TO ADD=
```

```
2 TO DELETE=
```

```
3 TO PRINT=
```

```
4 TO EXIT=
```

```
Enter your choice=
```

```
3
```

```
[1, 2]
```

```
1 TO ADD=
```

```
2 TO DELETE=
```

```
3 TO PRINT=
```

```
4 TO EXIT=
```

```
Enter your choice=
```

```
2
```

```
[2]
```

```
1 TO ADD=
```

```
2 TO DELETE=
```

```
3 TO PRINT=
```

```
4 TO EXIT=
```

```
Enter your choice=
```

```
3
```

```
Enter your choice=
3
[2]
1 TO ADD=
2 TO DELETE=
3 TO PRINT=
4 TO EXIT=
Enter your choice=
4
This Program Is Executed By 21001-CS-101
```



Q3. Write a java program to implement a Double Ended queue using a linked list.

PROGRAM:

```
import java.util.LinkedList;
import java.util.Scanner;
class doubleque
{
    public static void main(String s[])
    {
        int in=0;
        LinkedList<Integer> a=new LinkedList<Integer>();
        Scanner sc=new Scanner(System.in);

        while(true)
        {
            System.out.println("1 TO Add from starting =");
            System.out.println("2 TO Add from ending =");
            System.out.println("3 TO remove from starting=");
            System.out.println("4 TO remove from ending=");
            System.out.println("5 TO display from starting=");
            System.out.println("6 TO display from ending =");
            System.out.println("7 TO exit =");

            int choice,s1,s2;
            System.out.println("Enter your choice=");
            choice=sc.nextInt();
            k=sc.nextLine();
            switch(choice)
            {
                case 1:
                    System.out.println("enter the value : ");
                    s1=sc.nextInt();
                    a.addFirst(s1);
                    break;
                case 2:
                    System.out.println("enter the value : ");
                    s2=sc.nextInt();
                    a.addLast(s2);
                    break;
                case 3:
                    a.removeFirst();
                    break;
                case 4:
                    a.removeLast();
                    break;
                case 5:
                    System.out.println(a.getFirst());
                    break;
                case 6:
                    System.out.println(a.getLast());
                    break;
                case 7:
```

```

        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
        System.exit(1);
    }
}
}
}

```

OUTPUT:

```
D:\sujith>javac DoubleQueue.java
```

```
D:\sujith>java DoubleQueue
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
1
```

```
enter the value :
```

```
1
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
2
```

```
enter the value :
```

```
2
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
3
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=

```

```
Enter your choice=
```

```
3
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
6
```

```
2
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
4
```

```

1 TO Add from starting =
2 TO Add from ending =
3 TO remove from starting=
4 TO remove from ending=
5 TO display from starting=
6 TO display from ending =
7 TO exit =

```

```
Enter your choice=
```

```
7
```

```
This Program Is Executed By 21001-CS-101
```

Q4. Write a java program to implement a stack.

PROGRAM:

```
import java.util.Stack;
import java.util.Scanner;
class Stack1
{
    public static void main(String s[])
    {
        Stack<String> h=new Stack<String>();
        Scanner sc=new Scanner(System.in);
        String ele;
        while(true)
        {
            System.out.println("PRESS 1 TO ENTER THE ELEMENT=");
            System.out.println("PRESS 2 TO DELETE THE ELEMENT=");
            System.out.println("PRESS 3 TO SEE THE FIRST ELEMENT=");
            System.out.println("PRESS 4 TO EXIT=");
            int ch=sc.nextInt();
            String s3=sc.nextLine();
            switch(ch)
            {
                case 1:
                    System.out.println("ENTER THE ELEMENT=");
                    ele=sc.nextLine();
                    h.push(ele);
                    break;
                case 2:
                    System.out.println("DELETING THE ELEMENT=");
                    h.pop();
                    break;
                case 3:
                    System.out.println("DISPLAYING THE FIRST ELEMENT=");
                    System.out.println(h.peek());
                    break;
                case 4:
                    System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
                    System.exit(1);
                default:
                    System.out.println("INVALID NUMBER");
                    break;
            }
        }
    }
}
```

## OUTPUT:

```
A:\Java Records>javac Stack1.java
```

```
A:\Java Records>java Stack1
PRESS 1 TO ENTER THE ELEMENT=
PRESS 2 TO DELETE THE ELEMENT=
PRESS 3 TO SEE THE FIRST ELEMENT=
PRESS 4 TO EXIT=
1
ENTER THE ELEMENT=
3
PRESS 1 TO ENTER THE ELEMENT=
PRESS 2 TO DELETE THE ELEMENT=
PRESS 3 TO SEE THE FIRST ELEMENT=
PRESS 4 TO EXIT=
1
ENTER THE ELEMENT=
4
PRESS 1 TO ENTER THE ELEMENT=
PRESS 2 TO DELETE THE ELEMENT=
PRESS 3 TO SEE THE FIRST ELEMENT=
PRESS 4 TO EXIT=
3
DISPLAYING THE FIRST ELEMENT=
4
```

```
PRESS 1 TO ENTER THE ELEMENT=
PRESS 2 TO DELETE THE ELEMENT=
PRESS 3 TO SEE THE FIRST ELEMENT=
PRESS 4 TO EXIT=
1
ENTER THE ELEMENT=
6
PRESS 1 TO ENTER THE ELEMENT=
PRESS 2 TO DELETE THE ELEMENT=
PRESS 3 TO SEE THE FIRST ELEMENT=
PRESS 4 TO EXIT=
4
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q5. Write a java program to display current date and time.

PROGRAM:

```
import java.util.Date;
class CurrentDate
{
public static void main(String s[])
{
Date d=new Date();
System.out.println("THE DATE IS="+d);
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

OUTPUT:

```
A:\Java Records>javac CurrentDate.java
```

```
A:\Java Records>java CurrentDate
THE DATE IS=Mon Mar 06 13:51:50 IST 2023
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q6. Write a java program to find on which day next birthday will happen.

PROGRAM:

```
import java.util.Calendar;
import java.util.Scanner;
class NextBirthday
{
    public static void main(String s[])
    {
        Calendar ca=Calendar.getInstance();
        Scanner sc=new Scanner(System.in);
        System.out.println("ENTER THE YEAR:");
        int a=sc.nextInt();
        System.out.println("ENTER THE MONTH:");
        int b=sc.nextInt()-1;
        System.out.println("ENTER THE DATE:");
        int c=sc.nextInt();
        ca.set(a,b,c);
        int d=ca.get(Calendar.DAY_OF_WEEK);
        switch(d-1)
        {
            case 0:
                System.out.println("MY NEXT BIRTHDAY IS ON SUNDAY");
                break;
            case 1:
                System.out.println("MY NEXT BIRTHDAY IS ON MONDAY");
                break;
            case 2:
                System.out.println("MY NEXT BIRTHDAY IS ON TUESDAY");
                break;
            case 3:
                System.out.println("MY NEXT BIRTHDAY IS ON WEDNESDAY");
                break;
            case 4:
                System.out.println("MY NEXT BIRTHDAY IS ON THURSDAY");
                break;
            case 5:
                System.out.println("MY NEXT BIRTHDAY IS ON FRIDAY");
                break;
            case 6:
                System.out.println("MY NEXT BIRTHDAY IS ON SATURDAY");
                break;
        }
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

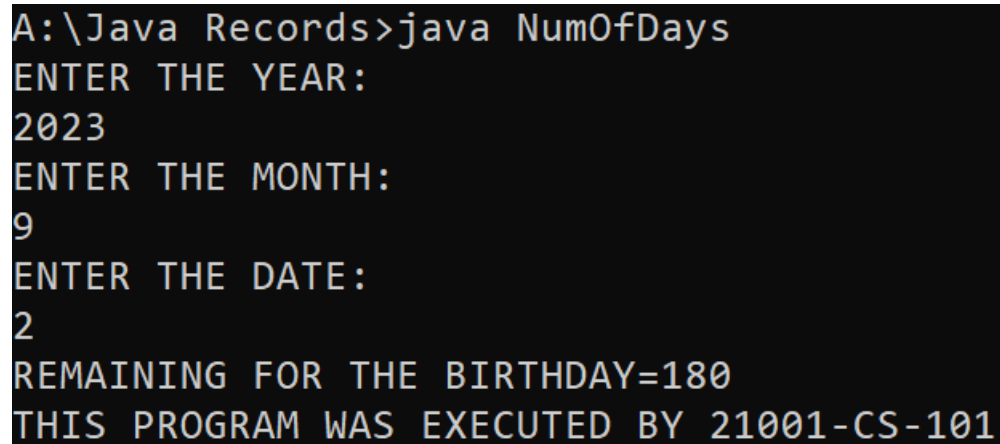
```
A:\Java Records>javac NextBirthday.java  
  
A:\Java Records>java NextBirthday  
ENTER THE YEAR:  
2022  
ENTER THE MONTH:  
9  
ENTER THE DATE:  
2  
MY NEXT BIRTHDAY IS ON FRIDAY  
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q7. Write a java program to calculate the number days to next birthday.

PROGRAM:

```
import java.util.Calendar;
import java.util.Scanner;
class NumOfDays
{
    public static void main(String s[])
    {
        Calendar ca=Calendar.getInstance();
        Scanner sc=new Scanner(System.in);
        System.out.println("ENTER THE YEAR:");
        int a=sc.nextInt();
        System.out.println("ENTER THE MONTH:");
        int b=sc.nextInt()-1;
        System.out.println("ENTER THE DATE:");
        int c=sc.nextInt();
        Calendar cb=Calendar.getInstance();
        cb.set(a,b,c);
        long day=((cb.getTimeInMillis()-ca.getTimeInMillis())/(1000*3600*24));
        System.out.println("REMAINING FOR THE BIRTHDAY="+day);
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:



```
A:\Java Records>java NumOfDays
ENTER THE YEAR:
2023
ENTER THE MONTH:
9
ENTER THE DATE:
2
REMAINING FOR THE BIRTHDAY=180
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```



Q8. Write a java program to calculate the number of seconds you lived.

PROGRAM:

```
import java.util.Calendar;
import java.util.Scanner;
class NumOfDays
{
    public static void main(String s[])
    {
        Calendar ca=Calendar.getInstance();
        Scanner sc=new Scanner(System.in);
        System.out.println("ENTER THE YEAR:");
        int a=sc.nextInt();
        System.out.println("ENTER THE MONTH:");
        int b=sc.nextInt()-1;
        System.out.println("ENTER THE DATE:");
        int c=sc.nextInt();
        Calendar cb=Calendar.getInstance();
        cb.set(a,b,c);
        System.out.println("SECONDS I LIVED ARE="+ca.getTimeInMillis()-
cb.getTimeInMillis())/1000);
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java Records>javac NumOfDays.java

A:\Java Records>java NumOfDays
ENTER THE YEAR:
2005
ENTER THE MONTH:
9
ENTER THE DATE:
2
SECONDS I LIVED ARE=552441594
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q9. Write a java program to generate random numbers from 1 to 100.

PROGRAM:

```
import java.util.Random;
class RandomNum
{
    public static void main(String s[])
    {
        Random ra=new Random(); for(int i=1;i<100;i++)
        {
            System.out.print(ra.nextInt(100)+" ");
        }
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java Records>java RandomNum
34 4 58 53 71 53 41 62 88 97 39 53 76 18 14 93 92 57 80 83 92 24 82 13 36 10 78 4 94 59 63 9 3
5 85 26 45 7 36 17 43 41 95 55 5 9 34 51 20 17 77 88 70 41 8 41 31 48 7 40 35 55 91 40 55 90
8 53 24 40 2 0 19 30 7 3 39 79 55 2 6 15 71 41 14 82 80 60 29 97 83 82 35 44 86 93 12 10 58
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q10. Write a java program to break the string into token with the default separators.

PROGRAM:

```
import java.util.Scanner;
import java.util.StringTokenizer;
class Token
{
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        String s;
        System.out.println("Enter a String : ");
        s=sc.nextLine();
        StringTokenizer st=new StringTokenizer(s);
        System.out.println("Tokens :- ");
        while(st.hasMoreTokens())
        {
            System.out.println(st.nextToken());
        }
        System.out.println("This Program Is Exececuted By 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java Records>javac Token.java
```

```
A:\Java Records>java Token
```

```
Enter a String :
```

```
my name is abhinay
```

```
Tokens :-
```

```
my
```

```
name
```

```
is
```

```
abhinay
```

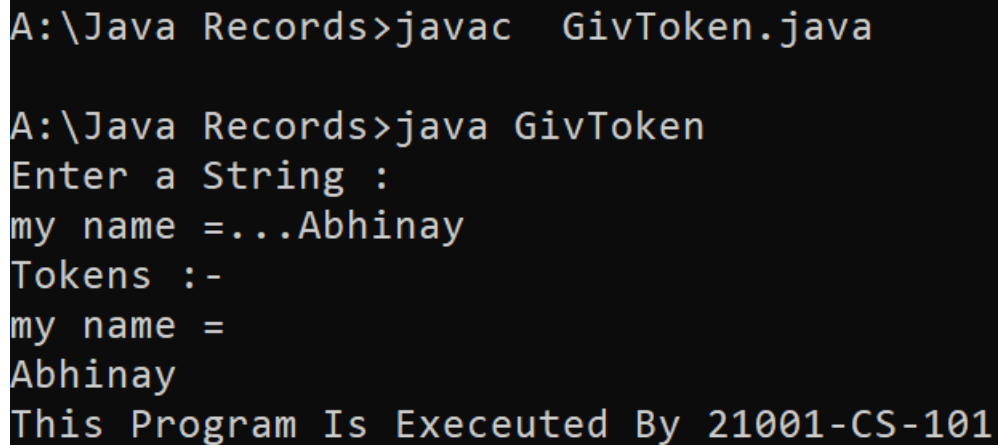
```
This Program Is Exececuted By 21001-CS-101
```

Q11. Write a java program to break the string into token with the given separators.

PROGRAM:

```
import java.util.Scanner;
import java.util.StringTokenizer;
class GivToken
{
    public static void main(String args[])
    {
        Scanner sc= new Scanner(System.in);
        String s;
        System.out.println("Enter a String : ");
        s=sc.nextLine();
        StringTokenizer st=new StringTokenizer(s,",.!",false);
        System.out.println("Tokens :- ");
        while(st.hasMoreTokens())
        {
            System.out.println(st.nextToken());
        }
        System.out.println("This Program Is Exececuted By 21001-CS-101");
    }
}
```

OUTPUT:



```
A:\Java Records>javac GivToken.java

A:\Java Records>java GivToken
Enter a String :
my name =...Abhinay
Tokens :-
my name =
Abhinay
This Program Is Exececuted By 21001-CS-101
```

## Exercise-16

Q1. Define Stream.?

Ans: A stream is a sequence of objects that supports various methods which can be pipelined to produce the desired result.

Q2. List the types of stream supported by java?

Ans: There are two basic types of stream defined by Java, called byte stream and character stream. The byte stream classes provide a convenient means for handling input and output of bytes and character streams provide a convenient means for handling input and output of characters, respectively

Q3. Name the package which is help to process the files? Ans:java.io package helps to process the files.

Q4. List 5 classes in the byte oriented streams?

Ans

ByteArrayInputStream

DataInputStream

FileInputStream

ObjectInputStream

PipedInputStream

Q5. List 5 classes in the character oriented streams?

1. FileReader

2.     FilterReader  
       CharArrayReader

3.InputStreamReader

StringReader

Q6. List two constructor of any byte Oriented Stream.?

Ans:

Q7. List two constructor of any character Oriented Stream.

Ans:

Q8. What is the advantage of using Buffered streams?

Ans: A stream buffer helps reduce erosion while protecting water quality.

## PROGRAM

Q1. Write a java program that checks file permissions.

PROGRAM:

```
import java.io.File;
class Check
{
    public static void main(String s[])
    {
        File f=new File("D1.java");
        System.out.println("IT CAN READ="+f.canRead());
        System.out.println("IT CAN WRITE="+f.canWrite());
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

BEFORE CHANGING:

```
A:\Java Records>javac Check.java

A:\Java Records>java Check
IT CAN READ=true
IT CAN WRITE=true
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

AFTER CHANGING:

```
A:\Java Records>javac Check.java

A:\Java Records>java Check
IT CAN READ=true
IT CAN WRITE=true
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q2. Write a java program to delete a file.

PROGRAM:

```
import java.io.File;
class Remove
{
public static void main(String s[])
{
File f=new File("D1.java");
System.out.println("IT IS EXISTING="+f.exists()); f.delete();
System.out.println("IT IS EXISTING NOW TOO="+f.exists());
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

OUTPUT:

```
A:\Java Records>javac Remove.java
```

```
A:\Java Records>java Remove
```

```
IT IS EXISTING=true
```

```
IT IS EXISTING NOW TOO=false
```

```
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q3. Write a java program to copy one file into another.

PROGRAM:

```
import java.io.*;
class Copy
{
    public static void main(String s[]) throws Exception
    {
        int a;
        FileOutputStream f1=new FileOutputStream("D1.java");
        FileInputStream f2=new FileInputStream("C1.java");
        while((a=f2.read())!=-1)
        {
            f1.write((char)a);
        }
        System.out.println("Succesfully copied");
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java Records>javac Copy.java
```

```
A:\Java Records>java Copy
```

```
Succesfully copied
```

```
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```



Q4. Write a java program that displays the content of the file on the console.

PROGRAM:

```
import java.io.*;
class Copy
{
    public static void main(String s[]) throws Exception
    {
        int a;
        FileInputStream f2=new FileInputStream("C1.java");
        while((a=f2.read())!=-1)
        {
            System.out.print((char)a);
        }
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java Records>notepad Copy.java

A:\Java Records>javac Copy.java

A:\Java Records>java Copy
package samepackage;

        public class C1
{
    public int a = 10;
        int b = 20;
    protected int c = 30;
    private int d = 40;
    public void display()
    {
        System.out.println("In same package in same class a : "+a);
        System.out.println("In same package in same class b : "+b);
        System.out.println("In same package in same class c : "+c);
        System.out.println("In same package in same class d : "+d);
    }
    public static void main(String s[])
    {
        C1 obj = new C1();
        obj.display();
        System.out.println("This Program Is Executed By 21001-CS-106");
    }
}

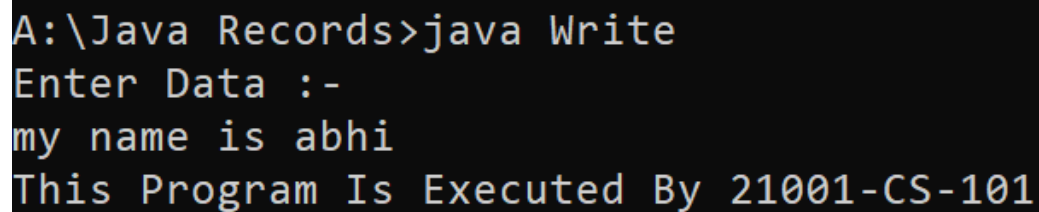
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q5. Write a java program that saves the data entered by the user into the file.

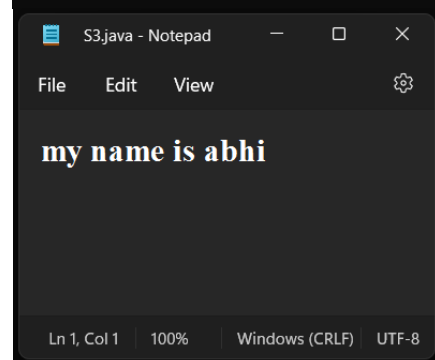
PROGRAM:

```
import java.io.FileOutputStream;
import java.util.Scanner;
class Write
{
    public static void main(String args[])throws Exception
    {
        FileOutputStream f=new FileOutputStream("S3.java");
        Scanner sc=new Scanner(System.in);
        String s;
        int i;
        System.out.println("Enter Data :- ");
        s=sc.nextLine();
        for(i=0;i<s.length();i++)
        {
            f.write(s.charAt(i));
        }
        f.close();
        System.out.println("This Program Is Executed By 21001-CS-101");
    }
}
```

OUTPUT:



```
A:\Java Records>java Write
Enter Data :-
my name is abhi
This Program Is Executed By 21001-CS-101
```



Q6. Write a java program that displays the contents of the file along with the line number.

PROGRAM:

```
import java.io.BufferedReader;
import java.io.FileReader;
class index
{ public static void main(String s[]) throws Exception
{
BufferedReader b=new BufferedReader(new FileReader("Check.java"));
int i=1;
String a;
System.out.println(i+"."+b.readLine());
while((a=b.readLine())!=null)
{ i++;
System.out.println(i+"."+a); }
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

OUTPUT:

```
A:\Java Records>javac Index.java

A:\Java Records>java index
1 .import java.io.File;
2 .class Check
3 .{
4 . public static void main(String s[])
5 .{
6 .File f=new File("D1.java");
7 .System.out.println("IT CAN READ="+f.canRead());
8 .System.out.println("IT CAN WRITE="+f.canWrite());
9 .System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
10 .}
11 .}
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```

Q7. Write a java program that displays the lines of the file from the end.

PROGRAM:

```
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Stack;
class Reverse
{
    public static void main(String s2[]) throws Exception
    {
        BufferedReader b=new BufferedReader(new FileReader("Index.java"));
        Stack<String> s=new Stack<String>();
        String a; int c=1;
        while((a=b.readLine())!=null)
        {
            s.push(a); c++;
        }
        while(c!=1)
        {
            System.out.println(s.peek()); c--;
            s.pop();
        }
        System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
D:\sujith>javac Reverse.java

D:\sujith>java Reverse
}
}
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-078");
System.out.println("IT CAN WRITE="+f.canWrite());
System.out.println("IT CAN READ="+f.canRead());
File f=new File("D1.java");
{
public static void main(String s[])
{
class Check
import java.io.File;
THIS PROGRAM WAS EXECUTED BY 21001-CS-078
```

Q8. Write a java program to count the number of words, line and characters in the file.

PROGRAM:

```
import java.io.RandomAccessFile; class Count
{
    public static void main(String args[]) throws Exception
    {
        RandomAccessFile r=new RandomAccessFile("Reverse.java","r");
        int lc=0,wc=1,cc=0,ch;        char a;
```

```
while((ch=r.read())!=-1)
{
    a=(char)ch;          if(a=='\n')
    lc++;
    if(a==' '||a=='\n')
    wc++;                cc++;
}
System.out.println("Number of Lines = "+lc);
System.out.println("Number of Words = "+wc);
System.out.println("Number of Characters = "+cc);
System.out.println("This Program is Executed By 21001-CS-101");
}
}
```

OUTPUT:

```
Number of Lines = 21
Number of Words = 69
Number of Characters = 498
This Program is Executed By 21001-CS-101
```

Q9. Write a java that shows buffered streams improves performance over other streams.

PROGRAM:

```
import java.io.*; import java.util.Date; class Performance
{
    public static void main(String[] args)throws FileNotFoundException,IOException
    {
        Date dt=new Date();          int i=0;
        long a1,a2,a3,a4;             a1=dt.getTime();
        FileInputStream fis=new FileInputStream("large.java");
        while(fis.read()!=-1)
        {}
        Date dt1=new Date();
        a2=dt1.getTime();
        System.out.println("Time taken for FileInputStream:"+a2-a1+" in milliseconds");
        BufferedReader br=new BufferedReader(new FileReader("large.java"));
        Date d=new Date();             long se=d.getTime();
        while(br.readLine()!=null)
        {}
        Date d1=new Date();             long se1=d1.getTime();
        System.out.println("Time taken for BufferedReaderStream:"+se1-se+" in milliseconds");
        System.out.println("This program is executed by pin 21001-CS-101");    }
}
```

OUTPUT:

```
D:\learnjava>javac Performance.java

D:\learnjava>java Performance
Time taken for FileInputStream:1208 in milliseconds
Time taken for BufferedReaderStream:33 in milliseconds
This program is executed by pin 21001-CS-078
```

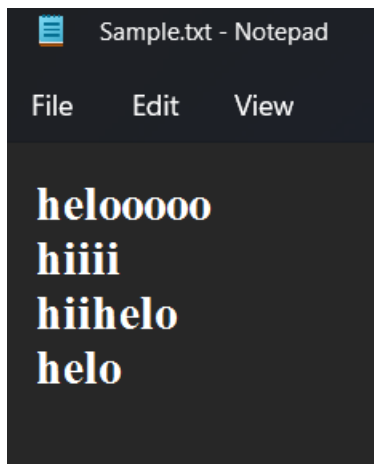
Q10. Write a java program to read the data from the file randomly.

PROGRAM:

```
import java.io.RandomAccessFile;
class RandomRead
{ public static void main(String s[]) throws Exception
{
RandomAccessFile ra=new RandomAccessFile("Sample.txt","r"); ra.seek(20);
System.out.println(ra.readLine());
System.out.println("THIS PROGRAM WAS EXECUTED BY 21001-CS-101");
}
}
```

OUTPUT:

```
helo
THIS PROGRAM WAS EXECUTED BY 21001-CS-101
```



Q12. Write a Java Program to sort a list of names in lexicographical order.

PROGRAM :

```
import java.util.Scanner; class Sort {
    public static void main(String args[])
    {
        Scanner sc=new Scanner(System.in);
        int i,n,z=0;
        System.out.println("HOW MANY NAMES YOU WANT TO ENTER : ");      n=sc.nextInt();
        String temp;
        String ch[]=new String[n];
        System.out.println("ENTER THE NAMES : ");      for( i=0;i<n;i++)
        {
            ch[i]=sc.next();
        }
        for(i=0;i<n;i++)
        {
            label:for(int j=0;j<n-1;)
            {
                if(ch[j].equals(ch[j+1]))
                {
                    j++;
                    continue label;
                }
                if(ch[j].charAt(z)==ch[j+1].charAt(z))
                {
                    z++;
                    continue label;
                }
                if(ch[j].charAt(z)>ch[j+1].charAt(z))
                {
                    temp=ch[j];
                    ch[j]=ch[j+1];
                    ch[j+1]=temp;
                    z=0;
                }
                j++;
            }
            z=0;
        }
        System.out.println(" AFTER SORTING : ");
        for( i=0;i<n;i++)
        {
            System.out.println(" " +ch[i]);
        }
        System.out.println("THIS PROGRAM IS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT :



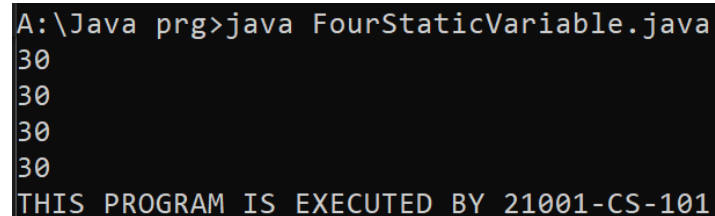
```
A:\Java prg>java Sort
HOW MANY NAMES YOU WANT TO ENTER :
3
ENTER THE NAMES :
john
abhi
bobi
  AFTER SORTING :
  abhi
  bobi
  john
THIS PROGRAM IS EXECUTED BY 21001-CS-101
```

Q4. Write a Java Program using a static variable. Change the value of the one static variable using four instances one after the other and access them using same four instances after all modifications. After Execution of the program add you observation.

PROGRAM :

```
class FourStaticVariable
{
    static int x=10;
    public static void main(String[] args)
    {
        FourStatic a=new FourStatic();
        FourStatic b=new FourStatic();          FourStatic c=new FourStatic();
        FourStatic d=new FourStatic();
        a.x=15;
        b.x= 20;
        c.x=25;
        d.x=30;
        System.out.println(a.x);
        System.out.println(b.x);          System.out.println(c.x);
        System.out.println(d.x);
        System.out.println("THIS PROGRAM IS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT :



```
A:\Java prg>java FourStaticVariable.java
30
30
30
30
THIS PROGRAM IS EXECUTED BY 21001-CS-101
```

Q5. Write the Errors when non static method is tried to access from a static method.

PROGRAM :

```
import java.util.Scanner; class NonStaticMethod
{
    public void method1()
    {
        System.out.println("THIS IS NON STATIC METHOD.");
    }
    public static void method2()
    {
        System.out.println("THIS IS STATIC METHOD.");
        method1();
    }
    public static void main(String args[])
    {
        NonStatic obj=new NonStatic();
        method2();
        System.out.println("THIS PROGRAM IS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT :

```
D:\learnjava>javac NonStaticMethod.java
NonStaticMethod.java:11: non-static method method1() cannot be referenced from a static context
    method1();
    ^
.\NonStatic.java:11: non-static method method1() cannot be referenced from a static context
    method1();
    ^
2 errors
```

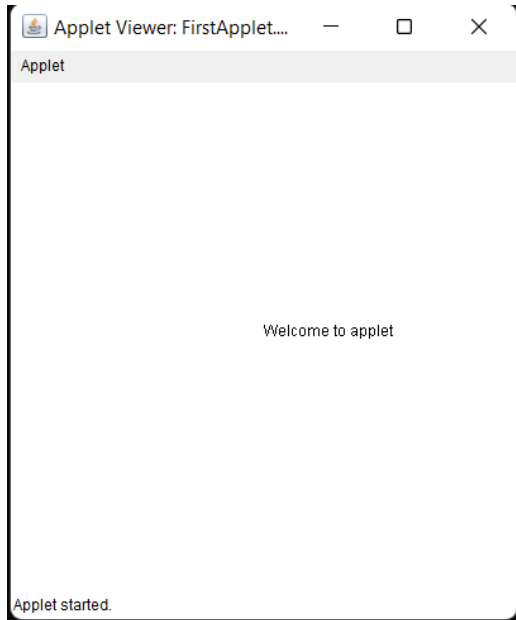
## PROGRAMS

Q1. Write the java program which overrides the following methods, After executing the program ; write when each method is called init(), paint(), stop(), destroy()

PROGRAM:

```
/*<html>
<head>
<title>My First applet</title>
</head>
<body>
<applet code="FirstApplet.class" height="400" width="400">
</applet>
</body>
</html>*/
import java.applet.*; //applet package
import java.awt.*; //graphics package
public class FirstApplet extends Applet
{
    public void init()
    {
        System.out.println("Called only once");
    }
    public void start()
    {
        System.out.println("i will be called every time i get focused");
    }
    public void paint(Graphics g)
    {
        g.drawString("Welcome to applet ", 200, 200);
        System.out.println("Called after Paint");
    }
    public void stop()
    {
        System.out.println("i lost focus");
    }
    public void destroy()
    {
        System.out.println("its gone");
        System.out.println("THIS PROGRAM IS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:



```
A:\Java prg>javac FirstApplet.java
```

```
A:\Java prg>Appletviewer FirstApplet.html
```

```
Called only once
```

```
i will be called every time i get focused
```

```
Called after Paint
```

```
Called after Paint
```

```
Called after Paint
```

```
Called after Paint
```

```
i lost focus
```

```
i will be called every time i get focused
```

```
Called after Paint
```

```
i lost focus
```

```
its gone
```

```
THIS PROGRAM IS EXECUTED BY 21001-CS-101
```

Q2. Write a java program to create an applet that displays Welcome message

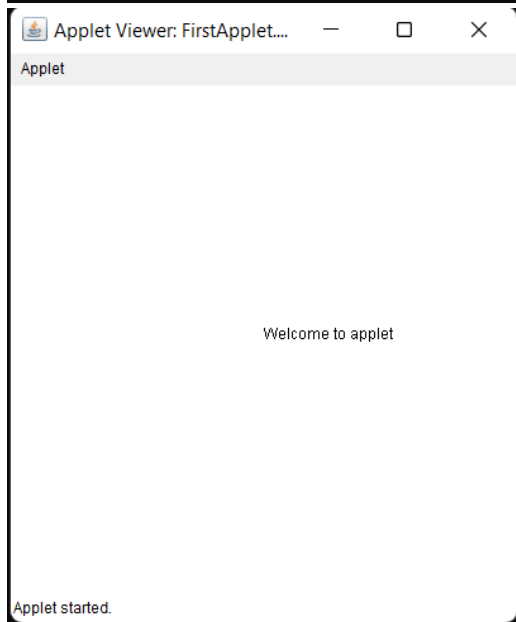
PROGRAM:

```
/*<html>
<head>
<title>My First applet</title>
</head>
<body>
<applet code="FirstApplet.class" height="400" width="400">
</applet>
</body>
</html>*/
import java.applet.*;//applet package
import java.awt.*;//graphics package
public class FirstApplet extends Applet
{
    public void init()
    {
        System.out.println("Called only once");
    }
    public void start()
    {
        System.out.println("i will be called every time i get focused");
    }
    public void paint(Graphics g)
    {
        g.drawString("Welcome to applet ",200,200);
        System.out.println("Called after Paint");
    }
    public void stop()
    {
        System.out.println("i lost focus");
    }
    public void destroy()
    {
        System.out.println("its gone");
        System.out.println("THIS PROGRAM IS EXECUTED BY 21001-CS-101");
    }
}
```

OUTPUT:

```
A:\Java prg>javac WelcomeApplet.java
```

```
A:\Java prg>Appletviewer WelcomeApplet.java  
THIS PROGRAM IS EXECUTED BY 21001-CS-101
```



Q3. Write a java program to create an applet that Displays Square, Rectangle, Hexagon, pentagon  
PROGRAM: