Example 1

**package** hello;

**import** java.util.\*;

**public** **class** Vectorexample1 {

**public** **static** **void** main(String args[]) {

//Create a vector

Vector<String> vec = **new** Vector<String>();

//Adding elements using add() method of List

vec.addElement("Tiger");

vec.addElement("Lion");

vec.addElement("Dog");

vec.addElement("Elephant");

//Adding elements using addElement() method of Vector

vec.addElement("Rat");

vec.addElement("Cat");

vec.addElement("Deer");

System.***out***.println("Elements are: "+vec);

}

}

**Output:**

Elements are: [Tiger, Lion, Dog, Elephant, Rat, Cat, Deer]

Graphical user interface, text, application

Description automatically generated

**Example 2**

**package** hello;

**import** java.util.\*;

**public** **class** Vectorexample2 {

**public** **static** **void** main(String args[]) {

//Create an empty vector with initial capacity 4

Vector<String> vec = **new** Vector<String>(6);

//Adding elements to a vector

vec.add("Tiger");

vec.add("Lion");

vec.add("Dog");

vec.add("Elephant");

//Check size and capacity

System.***out***.println("Size is: "+vec.size());

System.***out***.println("Default capacity is: "+vec.capacity());

//Display Vector elements

System.***out***.println("Vector element is: "+vec);

vec.addElement("Rat");

vec.addElement("Cat");

vec.addElement("Deer");

//Again check size and capacity after two insertions

System.***out***.println("Size after addition: "+vec.size());

System.***out***.println("Capacity after addition is: "+vec.capacity());

//Display Vector elements again

System.***out***.println("Elements are: "+vec); // [tiger,lion,dog,elephant,rat,cat,deer]

//Checking if Tiger is present or not in this vector

**if**(vec.contains("Tiger"))

{

System.***out***.println("Tiger is present at the index " +vec.indexOf("Tiger"));

}

**else**

{

System.***out***.println("Tiger is not present in the list.");

}

//Get the first element

System.***out***.println("The first animal of the vector is = "+vec.firstElement());

//Get the last element

System.***out***.println("The last animal of the vector is = "+vec.lastElement());

}

}

**Output:**

Size is: 4

Default capacity is: 6

Vector element is: [Tiger, Lion, Dog, Elephant]

Size after addition: 7

Capacity after addition is: 12

Elements are: [Tiger, Lion, Dog, Elephant, Rat, Cat, Deer]

Tiger is present at the index 0

The first animal of the vector is = Tiger

The last animal of the vector is = Deer

Graphical user interface, text, application

Description automatically generatedGraphical user interface, text, application

Description automatically generated

Example 3

**package** hello;

**import** java.util.\*;

**public** **class** Vectorexample4 {

**public** **static** **void** main(String args[]) {

//Create an empty Vector

Vector<Integer> in = **new** Vector<>();

//Add elements in the vector

in.add(100);

in.add(200);

in.add(300);

in.add(200);

in.add(400);

in.add(500);

in.add(600);

in.add(700);

//Display the vector elements

System.***out***.println("Values in vector: " +in);

System.***out***.println("Remove first occourence of element 200: "+in.remove((Integer)200));

System.***out***.println("Values in vector: " +in);

System.***out***.println("Remove element at index 4: " +in.remove(4));

System.***out***.println("New Value list in vector: " +in);

in.removeElementAt(5);

//System.out.println("Remove element at index 5: " +in.remove(5));

System.***out***.println("Vector element after removal: " +in);

System.***out***.println("Hash code of this vector = "+in.hashCode());

System.***out***.println("Element at index 1 is = "+in.get(1));

}

}

**Output:**

Values in vector: [100, 200, 300, 200, 400, 500, 600, 700]

Remove first occourence of element 200: true

Values in vector: [100, 300, 200, 400, 500, 600, 700]

Remove element at index 4: 500

New Value list in vector: [100, 300, 200, 400, 600, 700]

Vector element after removal: [100, 300, 200, 400, 600]

Hash code of this vector = 130123751

Element at index 1 is = 300

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated

Graphical user interface, text, application

Description automatically generated