

```

1  =====TestStack.java=====
2  package com.ameya.test;
3
4  import java.util.Iterator;
5  import java.util.Stack;
6
7  public class TestStack {
8
9      public static void main(String[] args) {
10         Stack<Integer> stck=new Stack<Integer>();
11         stck.push(10);
12         stck.push(20);
13         stck.push(30);
14         stck.push(40);
15         stck.push(50);
16         System.out.println(stck);
17         boolean isPresent=stck.contains(30);
18         System.out.println("stck contains 30 :: "+isPresent);
19         for(int i=0;i<stck.size();i++) {
20             System.out.println(stck.elementAt(i));
21         }
22         System.out.println("=====");
23         Iterator<Integer> itr=stck.iterator();
24         while(itr.hasNext()) {
25             int x = itr.next();
26             System.out.println(x);
27         }
28         /*stck.pop();
29         System.out.println(stck);
30         stck.pop();
31         System.out.println(stck);
32         stck.pop();
33         System.out.println(stck);
34         stck.pop();
35         System.out.println(stck);
36         stck.pop();
37         System.out.println(stck);*/
38     }
39 }
40
41 }
42 =====Person.java=====
43 package com.ameya.models;
44
45 package com.ameya.models;
46
47 public class Person {
48     private int id;
49     private String firstName;
50     private String lastName;
51     private int age;
52     public Person() {
53         firstName="N/A";
54         lastName="N/A";
55         id=-1;
56         age=-1;
57     }
58     public Person(int id, String firstName, String lastName, int age) {
59         super();
60         this.id = id;
61         this.firstName = firstName;
62         this.lastName = lastName;
63         this.age = age;
64     }
65     public int getId() {
66         return id;
67     }
68     public void setId(int id) {
69         this.id = id;
70     }
71     public String getFirstName() {
72         return firstName;
73     }

```

```

74     public void setFirstName(String firstName) {
75         this.firstName = firstName;
76     }
77     public String getLastName() {
78         return lastName;
79     }
80     public void setLastName(String lastName) {
81         this.lastName = lastName;
82     }
83     public int getAge() {
84         return age;
85     }
86     public void setAge(int age) {
87         this.age = age;
88     }
89     @Override
90     public String toString() {
91         return "Person [id=" + id + ", firstName=" + firstName + ", lastName=" +
92             lastName + ", age=" + age + "]\n";
93     }
94     @Override
95     public int hashCode() {
96         final int prime = 31;
97         int result = 1;
98         result = prime * result + id;
99         return result;
100    }
101    @Override
102    public boolean equals(Object obj) {
103        Person p=(Person)obj;
104        return this.id==p.getId()?true:false;
105    }
106
107
108
109 }
110
111 =====TestPersonStack.java=====
112 package com.ameya.test;
113
114 import java.util.Iterator;
115 import java.util.Stack;
116
117 import com.ameya.models.Person;
118
119 public class TestPersonStack {
120
121     public static void main(String[] args) {
122         Person p1=new Person();
123         Person p2=new Person(007, "Ameya", "Joshi", 45);
124         System.out.println("+++++++");
125         System.out.println("p1 FirstName :: "+p1.getFirstName());
126         System.out.println("p1 LastName  :: "+p1.getLastName());
127         System.out.println("p1 Id       :: "+p1.getId());
128         System.out.println("p1 Age      :: "+p1.getAge());
129         System.out.println("+++++++");
130         System.out.println("p2 FirstName :: "+p2.getFirstName());
131         System.out.println("p2 LastName  :: "+p2.getLastName());
132         System.out.println("p2 Id       :: "+p2.getId());
133         System.out.println("p2 Age      :: "+p2.getAge());
134
135         p1.setId(006);
136         p1.setFirstName("Kshiti");
137         p1.setLastName("Joshi");
138         p1.setAge(10);
139         System.out.println("+++++++");
140         System.out.println("p1 FirstName :: "+p1.getFirstName());
141         System.out.println("p1 LastName  :: "+p1.getLastName());
142         System.out.println("p1 Id       :: "+p1.getId());
143         System.out.println("p1 Age      :: "+p1.getAge());
144
145         System.out.println("+++++++");

```

```

146     System.out.println("p1 :: "+p1);
147     System.out.println("p2 :: "+p2);
148
149     Stack<Person> stck=new Stack<>();
150     stck.push(p1);
151     stck.push(p2);
152     stck.push(new Person(100,"Avani","Joshi",40));
153     stck.push(new Person(101,"Sanjay","Kadam",44));
154     stck.push(new Person(102,"Amol","Deshpande",38));
155     System.out.println("+++++++ STACK FOR PERSONS ++++++");
156     System.out.println(stck);
157
158     System.out.println("+++++");
159     for(int indx=0;indx<stck.size();indx++) {
160         Person p=stck.elementAt(indx);
161         System.out.println(p);
162     }
163     System.out.println("+++++");
164
165     Iterator<Person> itr=stck.iterator();
166     while(itr.hasNext()) {
167         Person p=itr.next();
168         System.out.println(p);
169     }
170
171     boolean isPresent=stck.contains(new Person(100,"Avani","Joshi",40));
172     System.out.println("Object is Found :: "+isPresent);
173 }
174
175 }
176

```