```
----------------TestStack.java-----------
package com.ameya.test;

import java.util.Iterator;
import java.util.Stack;

public class TestStack {

    public static void main(String[] args) {
        Stack<Integer> stack=new Stack<Integer>();
        stack.push(10);//stack.push(new Integer(10));
        stack.push(20);
        stack.push(30);
        stack.push(40);
        stack.push(50);
        System.out.println(stack);
        boolean isFound=stack.contains(30);
        System.out.println("Stack contains 30 :: "+isFound);
        for(int i=0;i<stack.size();i++) {
            System.out.print(stack.elementAt(i)+" , ");
        }
        System.out.println("\n----------------------------");
        for(int x : stack) {
            System.out.print(x+" , ");
        }
        System.out.println("\n----------------------------");
        Iterator<Integer> itr=stack.iterator();
        while(itr.hasNext()) {
            int n=itr.next();
            System.out.print(n+" , ");
        }
        System.out.println();
        stack.pop();
        System.out.println(stack);
        stack.pop();
        System.out.println(stack);
        stack.pop();
        System.out.println(stack);
        stack.pop();
        System.out.println(stack);
        stack.pop();
```

```
42          System.out.println(stack);
43          stack.pop();
44      }
45
46  }
47  ----------------Person.java-----------
48  package com.ameya.domain;
49
50  public class Person {
51      private long id;
52      private String firstName;
53      private String lastName;
54      private int age;
55      public Person() {
56          super();
57          // TODO Auto-generated constructor stub
58      }
59      public Person(long id, String firstName, String lastName, int age) {
60          super();
61          this.id = id;
62          this.firstName = firstName;
63          this.lastName = lastName;
64          this.age = age;
65      }
66      public long getId() {
67          return id;
68      }
69      public void setId(long id) {
70          this.id = id;
71      }
72      public String getFirstName() {
73          return firstName;
74      }
75      public void setFirstName(String firstName) {
76          this.firstName = firstName;
77      }
78      public String getLastName() {
79          return lastName;
80      }
81      public void setLastName(String lastName) {
82          this.lastName = lastName;
```

```
83      }
84      public int getAge() {
85          return age;
86      }
87      public void setAge(int age) {
88          this.age = age;
89      }
90      @Override
91      public String toString() {
92          return "Person [id=" + id + ", firstName=" + firstName + ",
            lastName=" +
93      lastName + ", age=" + age + "]\n";
94      }
95      @Override
96      public boolean equals(Object obj) {
97          return this.id==((Person)obj).getId() ? true : false;
98      }
99      @Override
100     public int hashCode() {
101         final long prime=31;
102         long result=1;
103         result=prime*result+id;
104         return (int)result;
105     }
106 }
107 ----------------TestPersonStack.java-----------
108 package com.ameya.test;
109
110 import java.util.Iterator;
111 import java.util.Stack;
112
113 import com.ameya.domain.Person;
114
115 public class TestPersonStack {
116
117     public static void main(String[] args) {
118         Person p1=new Person(100,"Ameya","Joshi",46);
119         Stack<Person> stack=new Stack<Person>();
120         stack.push(p1);
121         stack.push(new Person(120,"Ramesh","Seth",42));
122         stack.push(new Person(130,"Amol","Pawar",40));
```

```
123         stack.push(new Person(140,"Rakesh","Kumar",38));
124         System.out.println("++++++++++ Stack Of Persons ++++++++++");
125         System.out.println(stack);
126         Iterator<Person> itr=stack.iterator();
127         while(itr.hasNext()) {
128             Person person=itr.next();
129             System.out.println(person);
130         }
131         boolean isPresent=stack.contains(new
            Person(130,"Amol","Pawar",40));
132         System.out.println("isPresent :: "+isPresent);
133     }
134
135 }
136 ---------------PersonListUtil.java-----------
137 package com.ameya.utils;
138
139 import java.util.Iterator;
140 import java.util.LinkedList;
141 import java.util.List;
142
143 import com.ameya.domain.Person;
144
145 public class PersonListUtil {
146
147         List<Person> persons;
148         public PersonListUtil(List<Person> persons) {
149             this.persons=persons;
150         }
151         public void addFirst(Person person) {
152             persons.add(0,person);
153         }
154         public void addLast(Person person) {
155             persons.add(persons.size(),person);
156         }
157         public void addPerson(Person person) {
158             persons.add(person);
159         }
160         public boolean isPresent(Person person) {
161             return persons.contains(person);
162         }
```

```
163         public void printAllPersons() {
164             Iterator<Person> itr=persons.iterator();
165             while(itr.hasNext()) {
166                 Person person=itr.next();
167                 System.out.println(person.getId()+" :
                    "+person.getFirstName().toUpperCase()+" : "
168                 +person.getLastName().toUpperCase()+" :
                    "+person.getAge()+"\n");
169             }
170         }
171 }
172 -------------TestPersonList.java-----------
173 package com.ameya.test;
174
175 import java.util.ArrayList;
176 import java.util.LinkedList;
177
178 import com.ameya.domain.Person;
179 import com.ameya.utils.PersonListUtil;
180
181 public class TestPersonList {
182
183     public static void main(String[] args) {
184         LinkedList<Person> list=new LinkedList<Person>();
185         //ArrayList<Person> list=new ArrayList<Person>();
186         PersonListUtil util=new PersonListUtil(list);
187         util.addPerson(new Person(100,"Ameya","Joshi",46));
188         util.addPerson(new Person(110,"Amit","Pawar",42));
189         util.addPerson(new Person(120,"Amol","Doshi",43));
190         util.addPerson(new Person(130,"Amar","Patil",41));
191         util.printAllPersons();
192         System.out.println("isPresent :: "+util.isPresent(new
            Person(120,"Amol","Doshi",43)));
193     }
194
195 }
196 -----------------Product.java-----------
197 package com.ameya.domain;
198
199 public class Product {
200
```

```
201     private String productName;
202     private String category;
203     private int qty;
204     private double price;
205     private boolean isLive;
206     public Product() {
207         super();
208         // TODO Auto-generated constructor stub
209     }
210     public Product(String productName, String category,int qty, double
        price, boolean isLive) {
211         super();
212         this.productName = productName;
213         this.category=category;
214         this.qty = qty;
215         this.price = price;
216         this.isLive = isLive;
217     }
218     public String getProductName() {
219         return productName;
220     }
221     public void setProductName(String productName) {
222         this.productName = productName;
223     }
224     public int getQty() {
225         return qty;
226     }
227     public void setQty(int qty) {
228         this.qty = qty;
229     }
230     public double getPrice() {
231         return price;
232     }
233     public void setPrice(double price) {
234         this.price = price;
235     }
236     public boolean isLive() {
237         return isLive;
238     }
239     public void setLive(boolean isLive) {
240         this.isLive = isLive;
```

```
241     }
242     public String getCategory() {
243         return category;
244     }
245     public void setCategory(String category) {
246         this.category = category;
247     }
248     @Override
249     public String toString() {
250         return "Product [productName=" + productName + ",
            category="+category+", qty=" + qty + ", price=" + price + ",
            isLive=" + isLive + "]\n";
251     }
252
253 }
254 --------------TestGroupingBy.java-----------
255 package com.ameya.test;
256
257 import java.util.Arrays;
258 import java.util.List;
259 import java.util.Map;
260 import java.util.stream.Collectors;
261
262 public class TestGroupingBy {
263     public static void main(String[] args) {
264         List<Product> products=Arrays.asList(
265                 new Product("usbdrive","accessories",20,500,true),
266                 new Product("laptop","product",20,50000,true),
267                 new Product("hdd","accessories",20,500,true),
268                 new Product("bag","accessories",20,5000,false),
269                 new Product("LED Monitor","product",20,5000,true),
270                 new Product("DSLR Camera","product",20,50000,true),
271                 new Product("memory-card","accessories",20,500,true)
272                 );
273         Map<String, List<Product>> groupByCategory=products.stream()
274
                                                    .collect(Collectors.groupingB
                                                    y(Product::getCategory));
275
276           Map<String, Long> count = products.stream().collect(
277                 Collectors.groupingBy(Product::getCategory,
```

```
                Collectors.counting()));
278           System.out.println(count);
279         System.out.println(groupByCategory);
280         Map<Double, List<Product>>
            groupByPrice=products.stream().sorted((p1,p2)->(int)(p1.getPrice()
            -p2.getPrice()))
281                 .collect(Collectors.groupingBy(Product::getPrice));
282         System.out.println(groupByPrice);
283
284
285     }
286 }
287
```