```java
import java.util.*; // Importing the package containing the Scanner class

/* Driver class that is responsible for the main functionality */

public class Supermarket
{
public static void main(String[] args) // Main method
{

    Scanner sc = new Scanner(System.in);
    Products obj = new Products();
    Customers std = new Customers();
    int choice;
    int s;
    try // Try block of exception handling
    {
    do // Start of 'do' block
    {
        System.out.println(" _____________________________________________ ");
        System.out.println("|                                             |");
        System.out.println("|                                             |");
        System.out.println("|      Welcome to All In One Supermarket      |");
        System.out.println("|                                             |");
        System.out.println("|_____________________________________________|\n\n");


    /* Interface menu */

     System.out.println("Enter 1 to Add new Product.");
     System.out.println("Enter 2 to Upgrade Quantity of a Product.");
     System.out.println("Enter 3 to Search a Product.");
     System.out.println("Enter 4 to Show All Products.");
     System.out.println("Enter 5 to Register Customer.");
     System.out.println("Enter 6 to Show All Registered Customers.");
     System.out.println("Enter 7 to Check out  ");
     System.out.println("Enter 8 to Return Product ");
     System.out.println("Enter 9 to Exit");
        choice = sc.nextInt();

        /* Multiple-choice selection using a switch statement */

        switch(choice)
        {
             case 1:
                Product b = new Product();
                obj.addProduct(b);
                break;

            case 2:
                obj.upgrade();
                break;

            case 3:
                System.out.println("Enter Serial Number of Product");
                s = sc.nextInt();

                obj.searchno();


                break;

            case 4:
                obj.showAllProducts();
                break;
            case 5:
                Customer sm = new Customer();
                std.addCustomer(sm);
                break;
            case 6:
```

```
                    std.showAllCustomers();
                    break;
                case 7:
                    std.checkOutProduct(obj);
                    break;
                case 8:
                    std.returnProduct(obj);
                    break;
                    case 9:
                    System.exit(0);
                default:
                    System.out.println("wrong choice");

            }

        }
        while (true);
    }
    catch(Exception e) // Catch block of exception handling
    {
        System.out.println("Error");
    }
    }
    }

    abstract class variable // Abstract class that holds the instance variables
    {
     int serialno;
     String productname;
     int MRP;
     int productqty;
    }
     class Product extends variable // Class 'Product', which represents the actual product
     {


     int productqtycopy;

    Scanner sc = new Scanner(System.in);

     /* Default-constructor of class 'Product' */

     Product(){
        System.out.println("Enter Product Name:");
        this.productname = sc.nextLine();
        System.out.println("Enter MRP of Product");
        this.MRP = sc.nextInt();
        System.out.println("Enter Serial no. of Product");
        this.serialno = sc.nextInt();
        System.out.println("Enter Quantity");
        this.productqty = sc.nextInt();
        productqtycopy = this.productqty;
    }
    }

    /* Class 'Products', which holds an array of 'Product' objects, and methods to deal
    with them */

    class Products
    {

    Product bk[] = new Product[10];    // Creation of an array of 'Product' objects
    static int c;  // A static instance variable which keeps a count of the number of
    objects created

    Scanner sc = new Scanner(System.in);
```

```
137
138     int compareProduct(Product b1, Product b2){          // Function to compare two products
139
140         if (b1.productname.equalsIgnoreCase(b2.productname)){      // equalsIgnoreCase is a
            member of the String class
141
142             System.out.println("Product Already Exists.");
143             return 0;
144
145         }
146         if (b1.serialno==b2.serialno){
147
148             System.out.println("Product Already Exists.");
149             return 0;
150         }
151         return 1;
152     }
153
154     void addProduct(Product b){ // A method to add a product to the catalog
155
156         for (int i=0; i<c; i++){
157
158             if (this.compareProduct(b, this.bk[i]) == 0)
159                 return;
160
161         }
162
163         if (c<10){
164
165             bk[c] = b;
166             c++;
167
168         }
169         else{
170
171             System.out.println("No Space");
172
173         }
174
175     }
176
177     public void searchno(){ // A method to search for a product from the catalog
178         int sNo;
179         System.out.println("Enter Serial No of Product:");
180         sNo = sc.nextInt();
181         int flag = 0;
182         for (int i=0; i<c; i++){
183
184             if (sNo == bk[i].serialno)
185             {
186                 System.out.println("found");
187                 System.out.println(bk[i].serialno + "\t\t" + bk[i].productname + "\t\t" +
                    bk[i].MRP + "\t\t" + bk[i].productqty);
188                 flag++;
189                 return;
190             }
191
192         }
193         if (flag == 0)
194             System.out.println("No Product for Serial No " + sNo + " Found.");
195     }
196
197
198     public void showAllProducts(){ // A method to display a list of all products from the
        catalog
199
200         System.out.println("books");
201         System.out.println("SERIAL NO       PRODUCT NAME       MRP       PRODUCT QTY");
202
```

```
        System.out.println("-----------------------------------------------------------------
        --");
203     for (int i=0; i<c; i++){
204
205         System.out.println(bk[i].serialno + "\t\t" + bk[i].productname + "\t\t" +
            bk[i].MRP + "\t\t" + bk[i].productqty);
206
207
208     }
209
210 }
211
212 public void upgrade() // A method to add a product to the catalog
213 {
214     int sNo;
215     System.out.println("Enter Serial No of Product");
216      sNo = sc.nextInt();
217     for (int i=0; i<c; i++){
218
219         if (sNo == bk[i].serialno){
220
221             System.out.println("Enter No of Products to be Added:");
222             int add = sc.nextInt();
223             bk[i].productqty += add;
224             bk[i].productqtycopy += add;
225             return;
226
227         }
228
229     }
230
231 }
232
233 public int isAvailable(int sNo){ // A method to check whether a given product is
    available or not
234
235
236
237
238
239     for (int i=0; i<c; i++){
240
241         if (sNo == bk[i].serialno){
242             if(bk[i].productqtycopy > 0){
243
244                 System.out.println("Product is Available.");
245                 return i;
246
247             }
248             System.out.println("Product is Unavailable");
249             return -1;
250
251         }
252
253     }
254     return -1;
255
256
257 }
258
259 public Product checkOutProduct(){ // The method called when the customer wants to
    check-out of the supermarket
260
261     System.out.println("Enter Serial No of Product to be Checked Out.");
262     int sNo = sc.nextInt();
263
264     int productIndex =isAvailable(sNo);
265
266     if (productIndex!=-1){
```

```
            bk[productIndex].productqty--;

            return bk[productIndex];
        }

        return null;

    }

    public void returnProduct(Product b){ // A method to return a product to the supermarket

        for (int i=0; i<c; i++){

            if (b.equals(bk[i]) ){

                bk[i].productqty++;
                return;

            }

        }

    }

    }

    /* Class 'Customer', which represents the actual customer */

    class Customer {

    /* Instance variables of the class 'Customer' */

    String customerName;
    String phoneNum;

    Product orderedProducts[] = new Product[60]; // Creation of an array of Product objects
    public int productsCount = 0; // To keep a count of the products

    Scanner sc = new Scanner(System.in);

    /* Default-constrcutor of class 'Customer' */


    Customer(){

        System.out.println("Enter Customer Name:");
        this.customerName = sc.nextLine();

        System.out.println("Enter Registered Phone Number:");
        this.phoneNum = sc.nextLine();
    }
    }

    /* Class 'Customers', which holds an array of 'Customer' objects, and methods to deal
    with them */

     class Customers {

    Scanner sc = new Scanner(System.in);

    Customer theCustomers[] = new Customer[50]; // Creation of an array 'Customer' objects
    public static int count = 0;

    public void addCustomer(Customer s){ // A method to add a customer to the Customer type
    array

        for (int i=0; i<count; i++){

            if(s.phoneNum.equalsIgnoreCase(theCustomers[i].phoneNum)){
```

```
334
335             System.out.println("Customer of Reg Phone Number " + s.phoneNum + " is
                Already Registered.");
336             return;
337         }
338
339     }
340
341     if (count<50){
342
343         theCustomers[count] = s;
344         count++;
345
346     }
347
348 }
349 public void showAllCustomers(){ // A method to display a list of all customers
350
351     System.out.println("Customer Name\t\tReg Phone Number");
352     System.out.println("-----------------------------------------------");
353     for (int i=0; i<count; i++){
354
355         System.out.println(theCustomers[i].customerName + "\t\t         " +
            theCustomers[i].phoneNum);
356
357     }
358
359
360 }
361
362
363 public int isCustomer(){ // A method to check whether a given customer has already been
    registered or not
364
365     System.out.println("Enter Reg Phone Number:");
366     String phoneNum = sc.nextLine();
367
368     for (int i=0; i<count; i++){
369
370         if (theCustomers[i].phoneNum.equalsIgnoreCase(phoneNum)){
371
372             return i;
373
374         }
375
376     }
377     System.out.println("Customer is not Registered.");
378     System.out.println("Get Registered First.");
379
380
381     return -1;
382
383
384 }
385
386  void checkOutProduct(Products product){ // A method that is called when the customer
     wants to check out
387     int customerIndex =this.isCustomer();
388
389     if (customerIndex!=-1){
390
391
392         product.showAllProducts();
393         Product b = product.checkOutProduct();
394         System.out.println("checking out");
395         if (b!= null){
396
397             if (theCustomers[customerIndex].productsCount<=60){
398                 System.out.println("adding Product");
```

```
399
400
                    System.out.println("*************************INVOICE*********************
                    ********");
401
402
403
404
405
406
407
408
                    theCustomers[customerIndex].orderedProducts[theCustomers[customerIndex].p
                    roductsCount] = b;
409                 theCustomers[customerIndex].productsCount++;
410
411                 return;
412
413             }
414             else {
415
416                 System.out.println("Customers can't order more than 60 Products at a
                    time");
417                 return;
418
419             }
420         }
421         System.out.println("Product is not Available.");
422
423     }
424
425 }
426
427
428 public void returnProduct(Products product){ // A method that is called when a product
    is to be returned
429
430     int customerIndex = this.isCustomer();
431     if (customerIndex != -1){
432         System.out.println("S.No\t\t\tProduct Name\t\t\tMRP");
433         Customer s = theCustomers[customerIndex];
434         for (int i=0; i<s.productsCount; i++){
435
436             System.out.println(s.orderedProducts[i].serialno+ "\t\t\t" +
                s.orderedProducts[i].productname + "\t\t\t"+
437                     s.orderedProducts[i].MRP);
438
439         }
440         System.out.println("Enter Serial Number of Product to be Checked In:");
441         int sNo = sc.nextInt();
442         for (int i=0; i<s.productsCount; i++){
443
444             if (sNo == s.orderedProducts[i].serialno){
445
446                 product.returnProduct(s.orderedProducts[i]);
447                 s.orderedProducts[i]=null;
448                 return;
449
450             }
451
452
453         }
454         System.out.println("Product of Serial No "+sNo+"not Found");
455
456     }
457
458
459
460
```

```
461     }
462
463     }
464
465
```