

Islamic Education

A Summary of Lecture on Human Beings and Religion



Name

Muhammad Baihaqi Aulia Asy'ari

NIM

2241720145

Class

1I

Department

Information Technology

Study Program

D4 Informatics Engineering

Question:

1. Create class diagram/s for your program design
2. Create the program that implements class diagram to solve the problem of the case above (give comments to give explanation for the source code)
3. Put the (1) class diagram, (2) screenshot of your program source code and (3) screenshot of the output of the program in a PDF file
4. Submit the PDF file through your LMS account

Item	ItemSearch
itemCode : int name : String category : String initialStock : int	listItems : Item[]
Item(int itemCode, String name, String category, int initialStock) display() : void	addItem(int itemCode, String name, String category, int initialStock) : void displayItem() : void

Item Class:

```
1 package StockManagement;
2
3 public class Item {
4     int itemCode;
5     String name;
6     String category;
7     int initialStock;
8
9     //Item class parametric constructor
10    public Item(int itemCode, String name, String category, int
11        ↪ initialStock) {
12        this.itemCode = itemCode;
13        this.name = name;
14        this.category = category;
15        this.initialStock = initialStock;
16    }
17
18    //Display item attributes
19    public void display() {
20        System.out.printf("Item Code    : %d \n", itemCode);
```

```
20         System.out.printf("Name          : %s \n", name);
21         System.out.printf("Category       : %s \n", category);
22         System.out.printf("Initial Stock: %,d\n", initialStock);
23     }
24 }
```

ItemSearch Class:

```
1  package StockManagement;
2
3  public class ItemSearch {
4      Item[] listItems;
5
6      public ItemSearch(Item[] listItems) {
7          this.listItems = listItems;
8      }
9
10     /*
11     * adding new item in the list
12     * creating new list with +1 length and storing every item in that
13     * → list
14     * and then at last appending the new item using the parameter with
15     * → the item constructor
16     */
17     public void addItem(int itemCode, String name, String category,
18     → int initialStock) {
19         Item[] temp = new Item[listItems.length+1];
20         for (int i = 0; i < listItems.length; i++) {
21             temp[i] = listItems[i];
22         }
23         temp[temp.length-1] = new Item(itemCode, name, category,
24         → initialStock);
25         listItems = temp;
26     }
27
28     /*
29     * displaying all the item by using each item display() function
30     */
31     public void displayItem() {
32         for (Item item : listItems) {
33             item.display();
34         }
35     }
36 }
```

```

30         System.out.println("-----
        ↵ -----");
31     }
32 }
33
34 public void sortAscendingByInitialStock() {
35     Item[] temp = listItems;
36
37     for (int i = 0; i < temp.length-1; i++) {
38         for (int j = 0; j < temp.length-i; j++) {
39             if (temp[j].initialStock > temp[j-1].initialStock) {
40                 /*
41                  * swap
42                  */
43                 Item itemTemp = temp[j];
44                 temp[j] = temp[j-1];
45                 temp[j-1] = itemTemp;
46             }
47         }
48     }
49
50     listItems = temp;
51 }
52
53 public void sortAscendingByItemCode() {
54     Item[] temp = listItems;
55
56     for (int i = 0; i < temp.length-1; i++) {
57         for (int j = 0; j < temp.length-i; j++) {
58             if (temp[j].itemCode > temp[j-1].itemCode) {
59                 /*
60                  * swap
61                  */
62                 Item itemTemp = temp[j];
63                 temp[j] = temp[j-1];
64                 temp[j-1] = itemTemp;
65             }
66         }
67     }
68
69     listItems = temp;
70 }

```

```

71
72     public void sortAscendingByName() {
73         Item[] temp = listItems;
74
75         for (int i = 0; i < temp.length-1; i++) {
76             for (int j = 0; j < temp.length-i; j++) {
77                 if (temp[j].name.charAt(0) > temp[j-1].name.charAt(0))
78                     ↪ {
79                         /*
80                         * swap
81                         */
82                         Item itemTemp = temp[j];
83                         temp[j] = temp[j-1];
84                         temp[j-1] = itemTemp;
85                     }
86             }
87
88             listItems = temp;
89         }
90
91     public void sortAscendingByCategory() {
92         Item[] temp = listItems;
93
94         for (int i = 0; i < temp.length-1; i++) {
95             for (int j = 0; j < temp.length-i; j++) {
96                 if (temp[j].category.charAt(0) >
97                     ↪ temp[j-1].category.charAt(0)) {
98                     /*
99                     * swap
100                     */
101                     Item itemTemp = temp[j];
102                     temp[j] = temp[j-1];
103                     temp[j-1] = itemTemp;
104                 }
105             }
106
107             listItems = temp;
108         }
109
110

```

111 }

ItemMain Class:

```
1 package StockManagement;
2
3 public class ItemMain {
4     public static void main(String[] args) {
5         Item[] listItems = new Item[9];
6         listItems[0] = new Item(16030927, "Indomilk", "drink", 100);
7         listItems[1] = new Item(16100617, "Sprite", "drink", 70);
8         listItems[2] = new Item(16240401, "Yakult", "drink", 500);
9         listItems[3] = new Item(16270525, "Indomie", "food", 250);
10        listItems[4] = new Item(16971204, "Oreo", "food", 320);
11        listItems[5] = new Item(16100727, "Chocochips", "food", 120);
12        listItems[6] = new Item(16460329, "Ballpoint", "stationary",
13            ↪ 75);
14        listItems[7] = new Item(16320421, "Pencil", "stationary",
15            ↪ 110);
16        listItems[8] = new Item(16180729, "Book", "stationary", 57);
17        ItemSearch itemSearch = new ItemSearch(listItems);
18    }
19 }
```