

Muhammad Younis

Lab 07

Task 1 Company Management system

```
2 usages
1  class Company {
    1 usage
2      String name;
    3 usages
3      Store[] arraystore=new Store[2];
    no usages
4      int i=0;
    1 usage
5      Company(String name){this.name=name;}
    2 usages
9      int j=0;
    2 usages
10     void AddStore(Store store) {
11         if (j<=10) { arraystore[j++]=store; }
12         else { System.out.println("Can Not Have More than Ten Store "); }
13     }
    1 usage
14     int SearchNoOfProducts(String ProductName) {
15         int count=0;
16         for (Store store:arraystore) {
17             for (Product product : store.ProductList) {
18                 if (product.name.equals(ProductName)) { count++; }
19             }
20         }
21         return count;
22     }
23     Store[] store =new Store[2];
    1 usage
24     void displayAll() {
25         int i=1;
26         for(Store store:arraystore) {
27             for (Product product : store.ProductList) {
28                 System.out.print("Store " + i + " ");
29                 System.out.println(store.name);
30                 i++;
31                 break;
32             }
33         }
34     }
35 }
16 usages
36 class Product {
    5 usages
37     String name;
    1 usage
38     int quantity;
    1 usage
39     double price;
    3 usages
40     Product(String name,int quantity,double price) {
41         this.name=name;
42         this.price=price;
43         this.quantity=quantity; }
```

```

44      String Getname() { return name; }
48  }
    11 usages
49  class Store{
    3 usages
50      String name;
    1 usage
51      String Location;
    4 usages
52      int nbProduct;
    11 usages
53      Product ProductList[];
    2 usages
54      Store (String name,String Location) {
55          ProductList=new Product[3];
56          this.name=name;
57          this.Location=Location;
58      }
    no usages
59      String GetName() { return name; }
    2 usages
63      int i=0;
    6 usages
64      void addProduct(Product product) {
65          if (i>100) { System.out.println("Can Not Have More than Hundred Store ");
66          else { ProductList[i++]=product; }
67      }
68      boolean SearchProduct(String ProductName) {
69          boolean flag=false;
70          for (Product product : ProductList) {
71              if (product.name.equals(ProductName)) { flag=true; }
72              else { flag=false; }
73          }
74          return flag;
75      }
76      }
    1 usage
77      Product deleteProduct(String name) {
78          for (Product product:ProductList)
79          {
80              if(product.Getname().equalsIgnoreCase(name))
81              {
82                  for (int i=0;i<nbProduct;i++)
83                  {
84                      if (ProductList[i]==product)
85                      {
86                          break;
87                      }
88                      for (int j=0; j<nbProduct-1; j++)
89                      {
90                          ProductList[j]=ProductList[j+1];
91                      }
92                      ProductList[nbProduct-1]=null;
93                      nbProduct--;

```

```

94         return product;
95     }
96 }
97 }
98     return null;
99 }
100     2 usages
101 void displayAll()
102 {
103     for (Product product : ProductList) {
104         System.out.print("Store :");
105         System.out.println(product.name);
106     }
107 }
108 }
109 no usages
110 public class task_1
111 {
112     no usages
113     public static void main(String[] args){
114
115         Product p1 = new Product(name: "TV", quantity: 4, price: 34000);
116         Product p2 = new Product(name: "Bicycle", quantity: 4, price: 5500);
117         Product p3 = new Product(name: "Oven", quantity: 3, price: 70000);
118
119         Store s1 = new Store(name: "Makro", Location: "Karachi");
120         Store s2 = new Store(name: "Hypermart", Location: "Karachi");
121
122         s1.addProduct(p1);
123         s1.addProduct(p2);
124         s1.addProduct(p3);
125
126         s1.displayAll();
127         System.out.println();
128
129         Product tempProduct = s1.deleteProduct(name: "Bicycle");
130         if (tempProduct!=null) {
131             System.out.println("Product " + tempProduct.Getname() + " is deleted");
132         }
133         else {
134             System.out.println("There is no product to delete");
135         }
136         System.out.println();
137         s2.addProduct(p1);
138         s2.addProduct(p2);
139         s2.addProduct(p3);
140         s2.displayAll();
141
142         Company c1 = new Company(name: "Unilever");
143         c1.AddStore(s1);
144         c1.AddStore(s2);
145         System.out.println();
146         c1.displayAll();
147         System.out.println();
148         int n= c1.SearchNoOfProducts(ProductName: "TV");
149         System.out.println("Number of stores have TV "+n); }}

```

Task 1 Output

Store :TV

Store :Bicycle

Store :Oven

There is no product to delete

Store :TV

Store :Bicycle

Store :Oven

Store 1 Makro

Store 2 Hypermart

Number of stores have TV 2

Process finished with exit code 0

Task 2 Person

```
1  class Person {
    5 usages
2      String First_name;
    3 usages
3      String last_name;
    2 usages
4      Person(String First_name,String Last_name){
5          this.First_name=First_name;
6          this.last_name=Last_name;
7      }
8
9      4 usages 2 overrides
10     void DispalyDetails() {
11         System.out.println("Person First Name : "+First_name);
12         System.out.println("Person Last Name : "+First_name);
13     }
14
15     4 usages
16     class Students extends Person {
17         2 usages
18         int id;
19         2 usages
20         String course;
21         2 usages
22         String Teacher_name;
23         Students(String First_name,String Last_name,int id, String course,String teacher_name)
24             super(First_name,Last_name);
25             this.id=id;
26             this.course=course;
27             this.Teacher_name=teacher_name;
28     }
29
30     4 usages
31     void DispalyDetails() {
32         System.out.println("Student First Name : "+First_name);
33         System.out.println("Student Last Name : "+last_name);
34         System.out.println("Student ID : "+id);
35         System.out.println("Student Course : "+course);
36         System.out.println("Student Teacher Name : "+Teacher_name);
37         System.out.println();
38     }}
39
40     4 usages
41     class Teacher extends Person {
42         2 usages
43         String Subject_name;
44         2 usages
45         double Salary;
46         2 usages
47         Teacher (String First_name,String Last_name,String Subject_name,double Salary) {
48             super(First_name,Last_name);
49             this.Subject_name=Subject_name;
50             this.Salary=Salary;
51         }
52     }
53 }
```

```

41 void DispalyDetails()
42 {
43     System.out.println("Teacher First Name : "+First_name);
44     System.out.println("Teacher Last Name : "+last_name);
45     System.out.println("Teacher Subject Name : "+Subject_name);
46     System.out.println("Teacher salary : "+Salary);
47     System.out.println();
48 }
49 public class Task_2 {
50     public static void main(String[] args) {
51         Students s1=new Students(First_name: "Ali", Last_name: "Ahmed", id: 100, course: "CIT", teacher_name: "Suresh Kumar");
52         Students s2=new Students(First_name: "Abrar", Last_name: "Ali", id: 101, course: "DIT", teacher_name: "Muhammad Farooque");
53
54         Teacher t1=new Teacher(First_name: "Suresh", Last_name: "Kumar", Subject_name: "MS Office", Salary: 40000);
55         Teacher t2=new Teacher(First_name: "Muhammad", Last_name: "Farooque", Subject_name: "HTML & CSS", Salary: 45000);
56
57         s1.DispalyDetails();
58         s2.DispalyDetails();
59         t1.DispalyDetails();
60         t2.DispalyDetails(); }}
61

```

Task 2 Output

```

Student First Name : Ali
Student Last Name : Ahmed
Student ID : 100
Student Course : CIT
Student Teacher Name : Suresh Kumar

Student First Name : Abrar
Student Last Name : Ali
Student ID : 101
Student Course : DIT
Student Teacher Name : Muhammad Farooque

Teacher First Name : Suresh
Teacher Last Name : Kumar
Teacher Subject Name : MS Office
Teacher salary : 40000.0

Teacher First Name : Muhammad
Teacher Last Name : Farooque
Teacher Subject Name : HTML & CSS
Teacher salary : 45000.0

Process finished with exit code 0

```

Task 3 SMS, Email, Text Encoder

```
1  class Message {
2      2 usages
3      String Text;
4
5      4 usages 2 overrides
6      public void setText(String text) {
7          Text = text;
8      }
9
10     7 usages 2 overrides
11     public String getText() {
12         return Text;
13     }
14
15     2 overrides
16     public String toString() {
17         return getText();
18     }
19 }
20
21 2 usages
22 class SMS extends Message {
23     2 usages
24     int RecipientsContactNum;
25     no usages
26     public void setRecipientsContactNum(int recipientsContactNum) {
27         RecipientsContactNum = recipientsContactNum;
28     }
29     1 usage
30     public int getRecipientsContactNum() {
31         return RecipientsContactNum;
32     }
33
34     4 usages
35     @Override
36     public void setText(String text) {
37         super.setText(text);
38     }
39
40     7 usages
41     @Override
42     public String getText() {
43         return super.getText();
44     }
45
46     public String toString()
47     {
48         return getRecipientsContactNum()+getText();
49     }
50 }
51
52 2 usages
53 class Email extends Message {
54     2 usages
55     String Sender;
56     2 usages
57     String Reciever;
58     2 usages
59     String Subject;
60     no usages
61     public void setSender(String sender) {
62         Sender = sender;
63     }
64
65     no usages
66     public void setReciever(String reciever) {
67         Reciever = reciever;
68     }
69 }
```



```

43     }
    no usages
44     public void setSubject(String subject) {
45         Subject = subject;
46     }
    4 usages
47     @Override
48     public void setText(String text) {
49         super.setText(text);
50     }
    1 usage
51     public String getSender() {
52         return Sender;
53     }
    1 usage
54     public String getReciever() {
55         return Reciever;
56     }
    1 usage
57     public String getSubject() {
58         return Subject;
59     }
    7 usages
60     @Override
61     public String getText() {
62         return super.getText();
63     }
64     @Override
65     public String toString() {
66         return getSender()+getSubject()+getReciever()+getText();
67     }}
    2 usages
68     public class Task_3 {
    2 usages
69     @ boolean containsKeyword(String Text, String keyword) { return Text.contains(keyword); }
    1 usage
72     @ void encodeMessage(String message) {
73         int codedToAscii;
74         char coded ;
75         System.out.print("Encoded Text : ");
76
77         for (int i=0; i<message.length(); i++)
78         {
79             codedToAscii =message.charAt(i);
80             codedToAscii=codedToAscii+1;
81             coded= (char) codedToAscii;
82             System.out.print(coded);
83         }}
    no usages
84     public static void main(String[] args) {
85         Email email=new Email();
86         SMS sms=new SMS();
87         Task_3 obj=new Task_3();

```

```

89     String keyword="How";
90
91     sms.setText("Hello , How Are You?");
92     email.setText("Hi ! Where Are You Going");
93
94     boolean smscheck= obj.containsKeyword(sms.getText(),keyword);
95     System.out.println("Does Sms Contains Keyword '"+keyword+"' : " +smscheck);
96     boolean emailCheck= obj.containsKeyword(email.getText(),keyword);
97     System.out.println("Does Email Contains Keyword '"+keyword+"' : " +emailCheck);
98
99     String simpletext="Welcome To IBA";
100    System.out.println();
101    System.out.println("Simple Text : "+simpletext);
102    obj.encodeMessage(simpletext);
103    }}

```

Task 3 Output

```

"C:\Program Files\Java\jdk1.8.0_201\bin\jav
Does Sms Contains Keyword 'How' : true
Does Email Contains Keyword 'How' : false

Simple Text : Welcome To IBA
Encoded Text : Xfmdpnf!Up!JCB
Process finished with exit code 0

```

Task 4 Alien Game

```
1  class Alien
2  {
3      4 usages
4      public int health;
5      4 usages
6      public String name;
7      3 usages
8      Alien(String name,int health)
9      {
10         this.health=health;
11         this.name=name;
12     }
13 }
14
15 2 usages
16 class Snake_Alien extends Alien {
17     1 usage
18     private final int SnakeDamage=1;
19     3 usages
20     int numberOfEnemiesKilled;
21     1 usage
22     Snake_Alien(int numberOfEnemiesKilled, String name,int health)
23     {
24         super(name,health);
25         this.numberOfEnemiesKilled=numberOfEnemiesKilled;
26     }
27     int getDamage()
28     {
29         return SnakeDamage;
30     }
31     1 usage
32     int calculateDamage()
33     {
34         return getDamage()*numberOfEnemiesKilled;
35     }
36     public String toString()
37     {
38         return "Alien "+name+" "+health+" Health"+" And Caused "
39             +calculateDamage()+" Damage"+" By Killing "+numberOfEnemiesKilled+" Enemies";
40     }
41 }
42
43 2 usages
44 class Ogre_Alien extends Alien
45 {
46     1 usage
47     private final int OgreDamage=3;
48     3 usages
49     int numberOfEnemiesKilled;
50     1 usage
51     Ogre_Alien(int numberOfEnemiesKilled ,String name,int health)
52     {
53         super(name,health);
```

```

40         this.numberOfEnemiesKilled=numberOfEnemiesKilled;
41     }
42     1 usage
43     int getDamage() {
44         return OgreDamage;
45     }
46     1 usage
47     int calculateDamage()
48     {
49         return getDamage()*numberOfEnemiesKilled;
50     }
51     public String toString()
52     {
53         return "Alien "+name+" "+health+" Health"+" And Caused "+calculateDamage()+" Damage"+" By Killing "+numberOfEnemiesKilled+" Enemies";
54     }
55 }
56 2 usages
57 class MarshmallowMan_Alien extends Alien
58 {
59     1 usage
60     private final int MarshmallowDamage=3;
61     3 usages
62     int numberOfEnemiesKilled;
63     1 usage
64     MarshmallowMan_Alien(int numberOfEnemiesKilled ,String name,int health)
65     {
66         super(name,health);
67         this.numberOfEnemiesKilled=numberOfEnemiesKilled;
68     }
69     1 usage
70     int getDamage() {
71         return MarshmallowDamage;
72     }
73     1 usage
74     int calculateDamage()
75     {
76         return getDamage()*numberOfEnemiesKilled;
77     }
78     public String toString()
79     {
80         return "Alien "+name+" "+health+" Health"+" And Caused "
81             +calculateDamage()+" Damage"+" By Killing "+numberOfEnemiesKilled+" Enemies";
82     }
83 }
84 no usages
85 public class Task_4 {
86     no usages
87     public static void main(String[] args) {
88
89         Snake_Alien alien1=new Snake_Alien( numberOfEnemiesKilled: 150, name: "Rattle", health: 30);
90         Ogre_Alien alien2=new Ogre_Alien( numberOfEnemiesKilled: 80, name: "Bitter", health: 70);
91         MarshmallowMan_Alien alien3=new MarshmallowMan_Alien( numberOfEnemiesKilled: 220, name: "Tofee", health: 7);
92
93         System.out.println(alien1.toString());
94         System.out.println(alien2.toString());
95         System.out.println(alien3.toString());
96     }
97 }

```

Task 4 Output

```
"C:\Program Files\Java\jdk1.8.0_201\bin\java.exe" ...
```

```
Alien Rettle 30 Health And Caused 150 Damage By Killing 150 Enemies
```

```
Alien Bitter 70 Health And Caused 240 Damage By Killing 80 Enemies
```

```
Alien Toffee 7 Health And Caused 660 Damage By Killing 220 Enemies
```

```
Process finished with exit code 0
```