```java
package sim;
import java.util.ArrayList;
import java.util.Arrays;
import java.util.Collections;
import java.util.Scanner;
public class BugFixProgram {

    public static void main(String[] args) {
        System.out.println("\n**************\n");
        System.out.println("\tWelcome to TheDesk \n");
        System.out.println("**************");
        optionsSelection();

    }
    private static void optionsSelection() {
        String[] arr = {"1. I wish to review my expenditure",
                "2. I wish to add my expenditure",
                "3. I wish to delete my expenditure",
                "4. I wish to sort the expenditures",
                "5. I wish to search for a particular expenditure",
                "6. Close the application"
        };
        int[] arr1 = {1,2,3,4,5,6};
        int  slen = arr1.length;
        for(int i=0; i<slen;i++){
            System.out.println(arr[i]);
            // display the all the Strings mentioned in the String array
        }
        ArrayList<Integer> arrlist = new ArrayList<Integer>();
        ArrayList<Integer> expenses = new ArrayList<Integer>();
```

```
31          expenses.add(1000);
32          expenses.add(2300);
33          expenses.add(45000);
34          expenses.add(32000);
35          expenses.add(110);
36          expenses.addAll(arrlist);
37          System.out.println("\nEnter your choice:\t");
38          Scanner sc = new Scanner(System.in);
39          int  options =  sc.nextInt();
40          for(int j=1;j<=slen;j++){
41              if(options==j){
42                  switch (options){
43                      case 1:
44                          System.out.println("Your saved expenses are listed below: \n");
45                          System.out.println(expenses+"\n");
46                          optionsSelection();
47                          break;
48                      case 2:
49                          System.out.println("Enter the value to add your Expense: \n");
50                          int value = sc.nextInt();
51                          expenses.add(value);
52                          System.out.println("Your value is updated\n");
53                          expenses.addAll(arrlist);
54                          System.out.println(expenses+"\n");
55                          optionsSelection();
56
```

```
56
57                      break;
58                  case 3:
59                      System.out.println("You are about the delete all your expenses! \nC
60                      int con_choice = sc.nextInt();
61                      if(con_choice==options){
62                             expenses.clear();
63                          System.out.println(expenses+"\n");
64                          System.out.println("All your expenses are erased!\n");
65                      } else {
66                          System.out.println("Oops... try again!");
67                      }
68                      optionsSelection();
69                      break;
70                  case 4:
71                      sortExpenses(expenses);
72                      optionsSelection();
73                      break;
74                  case 5:
75                      searchExpenses(expenses);
76                      optionsSelection();
77                      break;
78                  case 6:
79                      closeApp();
80                      break;
81                  default:
82                      System.out.println("You have made an invalid choice!");
83                      break;
84              }
85          }
```

```
86          }
87
88      }
89      private static void closeApp() {
90          System.out.println("Closing your application... \nThank you!");
91              }
92      private static void searchExpenses(ArrayList<Integer> arrayList) {
93          int leng = arrayList.size();
94          System.out.println("Enter the expense you need to search:\t");
95          //
96          Scanner sc = new Scanner(System.in);
97          int input = sc.nextInt();
98          //Linear Search
99          for(int i=0;i<leng;i++) {
100             if(arrayList.get(i)==input) {
101                 System.out.println("Found the expense " + input + " at " + i + " position")
102             }
103         }
104     }
105     private static void sortExpenses(ArrayList<Integer> arrayList) {
106         int arrlength =  arrayList.size();
107        //Complete the method. The expenses should be sorted in ascending order.
108
109         Collections.sort(arrayList);
110         System.out.println("Sorted expenses: ");
111         for(Integer i: arrayList) {
112             System.out.print(i + " ");
```

Line: 85

```
112             System.out.print(i + " ");
113         }
114
115         System.out.println("\n");
116
117
118     }
119
120 }
121
122
```

Line: 85