```
import java.util.*;

    public class Bug {

        public static void main(String[] args) {
             optionsSelection();
}
        private static void optionsSelection() {
            String[] arr = {"1. review expenditure",
                    "2. add expenditure",
                    "3. delete  expenditure",
                    "4. sort the expenditures",
                    "5. search 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>();
            expenses.add(4000);
            expenses.add(500);
            expenses.add(90);
            expenses.add(8000);
            expenses.add(110);
            expenses.addAll(arrlist);
            System.out.println("\nEnter your choice:\t");
            Scanner sc = new Scanner(System.in);
            int  options =  sc.nextInt();
            for(int j=1;j<=slen;j++){
                if(options==j){
                    switch (options){
                        case 1:
                            System.out.println("Your saved expenses are listed below: \n");
                            System.out.println(expenses+"\n");
                            optionsSelection();
                            break;
                        case 2:
                            System.out.println("Enter the value to add your Expense: \n");
                            int value = sc.nextInt();
                            expenses.add(value);
                            System.out.println("Your value is updated\n");
                            expenses.addAll(arrlist);
                            System.out.println(expenses+"\n");
```

```
                    optionsSelection();

                    break;
                case 3:
                    System.out.println("You are about the delete your expense!
\nConfirm again by selecting the same option...\n");
                    int con_choice = sc.nextInt();
                    if(con_choice==options){
                          expenses.clear();
                       System.out.println(expenses+"\n");
                       System.out.println("All your expenses are erased!\n");
                    } else {
                       System.out.println("Oops... try again!");
                    }
                    optionsSelection();
                    break;
                case 4:
                    sortExpenses(expenses);
                    optionsSelection();
                    break;
                case 5:
                    searchExpenses(expenses);
                    optionsSelection();
                    break;
                case 6:
                    closeApp();
                    break;
                default:
                    System.out.println("You have made an invalid choice!");
                    break;
            }
        }
    }

}
private static void closeApp() {
    System.out.println("exit application");
        }
private static void searchExpenses(ArrayList<Integer> arrayList) {
    int leng = arrayList.size();
    System.out.println("Enter the expense you need to search:\t");
    //Complete the method
    Scanner sc = new Scanner(System.in);
    int search = sc.nextInt();
    int index = 0;
    for (int i = 0; i < arrayList.size(); i++) {
        if (arrayList.get(i) == search) {
```

```
                index = i;
            }
        }

        if (index == 0) {
            System.out.println("Value not found in the list");
        } else {
            System.out.println("Value found at index " + index);
        }


    }
    private static void sortExpenses(ArrayList<Integer> arrayList) {
        int arrlength =  arrayList.size();
       //Complete the method. The expenses should be sorted in ascending
order.
        //Collections.sort(arrayList);
        int temp = 0;
        int temp1 = 0;
        for (int i = 0; i < arrlength; i++) {
            for (int j = 1; j < (arrlength - i); j++) {
                if (arrayList.get(j-1) > arrayList.get(j)) {
                    // swap elements
                    temp = arrayList.get(j-1);
                    temp1 = arrayList.get(j);
                    arrayList.set(j,temp);
                    arrayList.set(j-1,temp1);
                }
            }
        }
        System.out.println("Expenses are sorted in ascending order:\n");
        System.out.println(arrayList);
        System.out.println();
    }
}
```