

Write-up:

- 1) Start the program.
- 2) Initialize an ArrayList called arrlist to store expenses.
- 3) Define a method addExpenses:
 - Create a new ArrayList called expenses.
 - Add some sample expenses to the expenses list.
 - Add all the expenses from the expenses list to the arrlist.
- 4) Define a method addAdditional:
 - Create a Scanner object to get user input.
 - Prompt the user to enter a new expense value.
 - Read the input value from the user.
 - Add the user's input value to the arrlist.
 - Print the updated arrlist.
- 5) Define a method optionsSelection:
 - Create an instance of the FixBugs class named errorHandle.
 - Create a list of options in an array called arr.
 - Create an array of integers called arr1 to represent option numbers.
 - Get the length of the arr1 array and store it in slen.
 - Display the available options to the user.
 - Get the user's choice.
 - Iterate through the options, and if the user's choice matches an option number, execute the corresponding action:
 - Option 1: Display the list of saved expenses.
 - Option 2: Add a new expense by calling the addAdditional method.
 - Option 3: Delete all expenses if confirmed.
 - Option 4: Add sample expenses by calling the addExpenses method.
 - Option 5: Search for a specific expenditure using the searchExpenses method.
 - Option 6: Close the application and print a closing message.
 - Handle an invalid choice with an error message.
 - Continue to prompt for user input until the user chooses to close the application.
- 6) Define a method closeApp to display a closing message.

- 7) Define a method `searchExpenses` that takes a list of expenses as a parameter:
 - Get the size of the list.
 - Prompt the user to enter an expense value to search for.
 - Read the user's input value.
 - Check if the value exists in the list:
 - If found, print a message indicating its existence and index.
 - If not found, print a message indicating that it doesn't exist.
- 8) Define a method `sortExpenses` to sort the expenses in ascending order.
- 9) In the main method:
 - Display a welcome message.
 - Create an instance of the `FixBugs` class named `errorHandle`.
 - Create an empty list named `myList`.
 - Initialize the list of expenses by calling the `addExpenses` method.
 - Start the main options selection loop by calling the `optionsSelection` method.
- 10) The program continues to run and process user input until the user chooses to close the application.
- 11) End the program.