

Anghela M. Aliza bsit3b

The screenshot shows two code editors in Microsoft Visual Studio. The top editor contains the main program logic, and the bottom editor contains the implementation of the `AddContact()` method.

Top Editor (Main Program Logic):

```
06 Hands - On Activity 1
using System;
using System.IO;
using System.Collections.Generic;
namespace SIXHandsonActivity
{
    class Program
    {
        // File path (relative to program folder)
        static string filePath = "contacts.txt";

        static void Main()
        {
            while (true)
            {
                Console.Clear();
                Console.WriteLine("Welcome to the Contact Management System");
                Console.WriteLine("Please choose an option:");
                Console.WriteLine("1. Add a new contact");
                Console.WriteLine("2. View all contacts");
                Console.WriteLine("3. Delete a contact");
                Console.WriteLine("4. Exit");
                Console.Write("Enter your choice: ");
                string choice = Console.ReadLine();

                if (choice == "1")
                    AddContact();
                else if (choice == "2")
                    ViewContacts();
                else if (choice == "3")
                    DeleteContact();
                else if (choice == "4")
                {
                    Console.WriteLine("\nThank you for using the Contact Management System. Goodbye!");
                    break;
                }
                else
                    Console.WriteLine("Invalid choice. Please try again.");

                Console.WriteLine("\nPress any key to continue...");
                Console.ReadKey();
            }
        }

        // Add a new contact
        static void AddContact()
        {
        }
```

Bottom Editor (AddContact() Implementation):

```
06 Hands - On Activity 1
// Add a new contact
static void AddContact()
{
    try
    {
        Console.Write("\nEnter the contact name: ");
        string name = Console.ReadLine();

        if (string.IsNullOrWhiteSpace(name))
        {
            Console.WriteLine("Error: Name cannot be empty. Please try again.");
            return;
        }

        Console.Write("Enter the contact phone number (10 digits): ");
        string phone = Console.ReadLine();

        if (phone.Length != 10 || !long.TryParse(phone, out _))
        {
            Console.WriteLine("Error: Phone number must be 10 digits. Please try again.");
            return;
        }

        // StreamWriter used to write text to a file
        using (StreamWriter writer = new StreamWriter(filePath, true))
        {
            writer.WriteLine($"{name},{phone}");
        }

        Console.WriteLine("Contact added successfully!");
    }
    catch
    {
        Console.WriteLine("Error: Unable to access contacts file.");
    }
}
```

```
81 // View all contacts
82
83     static void ViewContacts()
84     {
85         try
86         {
87             // File.Exists checks if file is present
88             if (!File.Exists(filePath))
89             {
90                 Console.WriteLine("No contacts found.");
91                 return;
92             }
93
94             // StreamReader used to read from a file
95             using (StreamReader reader = new StreamReader(filePath))
96             {
97                 string line;
98                 int count = 0;
99                 List<string> contacts = new List<string>();
100
101                while ((line = reader.ReadLine()) != null)
102                {
103                    contacts.Add(line);
104                }
105
106                if (contacts.Count == 0)
107                {
108                    Console.WriteLine("No contacts found.");
109                    return;
110                }
111
112                Console.WriteLine("\nAll Contacts:");
113                foreach (string contact in contacts)
114                {
115                    count++;
116                    string[] parts = contact.Split(',');
117                    Console.WriteLine($"{count}. {parts[0]}, {parts[1]}");
118                }
119            }
120        catch
121        {
122            Console.WriteLine("Error: Unable to read contacts file.");
123        }
124    }
125}
126
```

06 Hands - On Activity 1

SIXHandsonActivity.Program

```
127 // Delete a contact
128
129     static void DeleteContact()
130     {
131         try
132         {
133             if (!File.Exists(filePath))
134             {
135                 Console.WriteLine("Error: No contacts found.");
136                 return;
137             }
138
139             List<string> contacts = new List<string>(File.ReadAllLines(filePath));
140
141             Console.Write("\nEnter the name of the contact to delete: ");
142             string name = Console.ReadLine();
143
144             bool found = false;
145
146             for (int i = 0; i < contacts.Count; i++)
147             {
148                 string[] parts = contacts[i].Split(',');
149                 if (parts[0].Equals(name, StringComparison.OrdinalIgnoreCase))
150                 {
151                     contacts.RemoveAt(i);
152                     found = true;
153                     break;
154                 }
155
156                 if (found)
157                 {
158                     File.WriteAllLines(filePath, contacts);
159                     Console.WriteLine("Contact deleted successfully!");
160                 }
161                 else
162                 {
163                     Console.WriteLine("Error: Contact not found.");
164                 }
165             }
166         catch
167         {
168             Console.WriteLine("Error: Unable to access contacts file.");
169         }
170     }
171 }
172 }
```

Welcome to the Contact Management System

Please choose an option:

1. Add a new contact
2. View all contacts
3. Delete a contact
4. Exit

Enter your choice: |

Welcome to the Contact Management System

Please choose an option:

1. Add a new contact
2. View all contacts
3. Delete a contact
4. Exit

Enter your choice: 1

Enter the contact name: PICARD

Enter the contact phone number (10 digits): 1234567890

Contact added successfully!

Press any key to continue... |

Welcome to the Contact Management System

Please choose an option:

1. Add a new contact
2. View all contacts
3. Delete a contact
4. Exit

Enter your choice: 2

All Contacts:

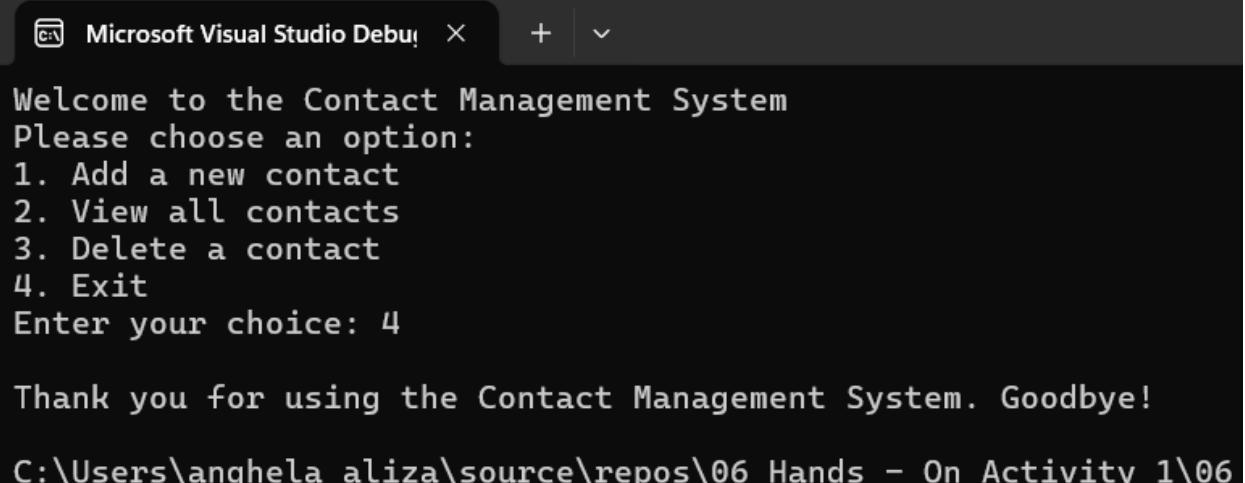
1. PICARD, 1234567890

Press any key to continue... |

```
Welcome to the Contact Management System
Please choose an option:
1. Add a new contact
2. View all contacts
3. Delete a contact
4. Exit
Enter your choice: 3
```

```
Enter the name of the contact to delete: PICARD
Contact deleted successfully!
```

```
Press any key to continue...
```



The screenshot shows a Microsoft Visual Studio Debug window. The title bar says "Microsoft Visual Studio Debug". The main area displays the same output as the previous terminal window, including the welcome message, menu options, user input "Enter your choice: 4", and the final message "Thank you for using the Contact Management System. Goodbye!". At the bottom of the window, the file path "C:\Users\anghela aliza\source\repos\06 Hands - On Activity 1\06" is visible.

```
Welcome to the Contact Management System
Please choose an option:
1. Add a new contact
2. View all contacts
3. Delete a contact
4. Exit
Enter your choice: 4

Thank you for using the Contact Management System. Goodbye!
C:\Users\anghela aliza\source\repos\06 Hands - On Activity 1\06
```

