

TEAM-CODE WARRIORS

MEMBERS

PRERNA SINGH 24SCSE1180097

SUNITI 24SCSE1180348

SANJH MAHESHWARI 24SCSE1180350

SHRISTI PANDEY 24SCSE1180425



[This Photo](#) by Unknown Author is
licensed under [CC BY-SA](#)




Project Structure (Basic Version)

CampusChatBotProject/

|—  CampusChatBot.java ← Your chatbot
code goes here

|—  README.md ← (Optional)
Description, instructions, features

|—  out/ ← (Auto-created) Compiled
class files

```
import java.util.*;

public class CampusChatBot {
    private static final Map<String, String> knowledgeBase = new HashMap<>();

    public static void main(String[] args) {
        initializeKnowledgeBase();
        Scanner scanner = new Scanner(System.in);

        System.out.println("CampusBot: Hi! Ask me anything about our campus (type 'exit' to quit).");

        while (true) {
            System.out.print("You: ");
            String userInput = scanner.nextLine().toLowerCase();

            if (userInput.contains("exit") || userInput.contains("bye")) {
                System.out.println("CampusBot: Goodbye! Have a great day.");
                break;
            }

            boolean found = false;
            for (String keyword : knowledgeBase.keySet()) {
                if (userInput.contains(keyword)) {
                    System.out.println("CampusBot: " + knowledgeBase.get(keyword));
                    found = true;
                    break;
                }
            }

            if (!found) {
                System.out.println("CampusBot: Sorry, I don't have an answer for that. Try rephrasing.");
            }
        }

        scanner.close();
    }
}
```

```
private static void initializeKnowledgeBase() {  
    knowledgeBase.put("admission", "Admissions are open  
from June to August. Visit the admissions office or our  
website.");  
    knowledgeBase.put("library", "The library is open from 8  
AM to 10 PM on weekdays.");  
    knowledgeBase.put("hostel", "Hostels are available for  
both boys and girls. Contact the hostel office for more  
info.");  
    knowledgeBase.put("placement", "Our placement cell  
is active year-round. Companies visit mainly from October  
to March.");  
    knowledgeBase.put("canteen", "The canteen serves  
food from 9 AM to 7 PM daily.");  
    knowledgeBase.put("fees", "Fee details are available  
on the official website or the accounts department.");  
    knowledgeBase.put("courses", "We offer courses in  
engineering, management, arts, and science.");  
    knowledgeBase.put("transport", "Bus services are  
available from 6 AM to 7 PM covering major city routes.");  
    knowledgeBase.put("sports", "The campus has facilities  
for cricket, football, basketball, and a gym.");  
}  
}
```

Let's go through your **CampusChatBot** code line by line and explain everything in **simple terms**, so you understand **what each part does**:

`import java.util.*;`

This line imports **all utility classes** from the java.util package, like Scanner, Map, HashMap, etc., which you'll use later in the program.

`public class CampusChatBot {`

This starts the **definition of your main class** named CampusChatBot. Everything in your program is written inside this class.

`private static final Map<String, String> knowledgeBase = new HashMap<>();`

- private: This means only this class can access the variable.
- static: It belongs to the class itself, not objects.
- final: You cannot reassign knowledgeBase to another map later.
- Map<String, String>: A key-value pair data structure.
- new HashMap<>(): You are creating a new empty HashMap where:
 - the **key** is a String (like "library", "fees")
 - the **value** is also a String (the answer)

So this is your **knowledge base** – where you store question keywords and their answers.

`public static void main(String[] args) {`

This is the **main method** – the entry point of your Java program. When you run your code, this is the method that starts executing first.

```
initializeKnowledgeBase();
```

You're calling a method that fills the knowledgeBase with pre-defined question-answer pairs.

```
Scanner scanner = new Scanner(System.in);
```

This creates a Scanner object named scanner to **take user input from the keyboard** (standard input).

```
System.out.println("CampusBot: Hi! Ask me anything about our campus (type 'exit' to quit).");
```

This prints a welcome message on the screen to let the user know they can start chatting.

```
while (true) {
```

Starts an **infinite loop** — it keeps running until you manually break out (which you do when the user types "exit" or "bye").

```
System.out.print("You: ");
```

This displays a prompt to make it look like a chat.

```
String userInput = scanner.nextLine().toLowerCase();
```

- scanner.nextLine() takes a full line of text input from the user.
- .toLowerCase() converts it to lowercase so your bot can match keywords **case-insensitively**.

```
if (userInput.contains("exit") || userInput.contains("bye")) {
```

Checks if the user typed "exit" or "bye" anywhere in the input.

System.out.println("CampusBot: Goodbye! Have a great day.");

Prints a farewell message if the user wants to leave.

break;

Breaks out of the while (true) loop, stopping the chatbot.

boolean found = false;

This variable tracks whether the bot found a matching keyword in the user's message.

for (String keyword : knowledgeBase.keySet()) {

This loop goes through **each keyword** in the knowledge base (like "library", "hostel", etc.).

if (userInput.contains(keyword)) {

Checks if the user's message contains any of the known keywords.

System.out.println("CampusBot: " + knowledgeBase.get(keyword));

Prints the **answer** (value) from the knowledgeBase that matches the keyword.

found = true;

Sets found to true so we know a match was found.

break;

Exits the for loop after finding the **first matching keyword**.

if (!found) {

Checks if no keyword matched the user's input.

System.out.println("CampusBot: Sorry, I don't have an answer for that. Try rephrasing.");

If no keyword matched, this message is printed to suggest the user try again.

scanner.close();

Closes the Scanner object to release system resources. (Good practice)

private static void initializeKnowledgeBase() {

This method fills the knowledge base with fixed question-answer pairs (like "library" → "Library is open from...").

knowledgeBase.put("admission", "...");

Each of these lines adds a new **keyword-answer** pair into the knowledgeBase map.



Summary:

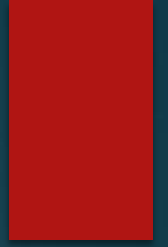
This code is a **simple keyword-based chatbot** that:

- Takes user input,
- Looks for known words like "library", "fees", "hostel", etc.
- Replies with the answer if the keyword is found
- Says goodbye when the user types "exit" or "bye"



Example Questions

- "Tell me about hostel"
- "What are the canteen timings?"
- "exit"



CampusBot: Hi! Ask me anything about our campus (type 'exit' to quit).

You: What are the library timings?

CampusBot: The library is open from 8 AM to 10 PM on weekdays.

You: Do we have hostel facility?

CampusBot: Hostels are available for both boys and girls. Contact the hostel office for more info.

You: What about placements?

CampusBot: Our placement cell is active year-round. Companies visit mainly from October to March.

You: Can you tell me the fees?

CampusBot: Fee details are available on the official website or the accounts department.

You: Tell me about the food

CampusBot: Sorry, I don't have an answer for that. Try rephrasing.

You: canteen?

CampusBot: The canteen serves food from 9 AM to 7 PM daily.

You: bye

CampusBot: Goodbye! Have a great day.

SCREENSHOT

The screenshot shows an IDE with the following components:

- Explorer:** Shows the project structure with 'CampusChatBot.java' open.
- Editor:** Displays the code for 'CampusChatBot.java'.
- Terminal:** Shows the execution of the program.

```
1  import java.util.*;
2
3  public class CampusChatBot {
4      private static final Map<String, String> knowledgeBase = new HashMap<>();
5
6      Run main | Debug main | Run | Debug
7      public static void main(String[] args) {
8          initializeKnowledgeBase();
9          Scanner scanner = new Scanner(System.in);
10
11         System.out.println(x:"CampusBot: Hi! Ask me anything about our campus (type 'exit' to quit).");
12
13         while (true) {
14             System.out.print(s:"You: ");
15             String userInput = scanner.nextLine().toLowerCase();
16
17             if (userInput.contains(s:"exit") || userInput.contains(s:"bye")) {
18                 System.out.println(x:"CampusBot: Goodbye! Have a great day.");
19                 break;
20             }
21         }
22     }
23 }
```

Terminal Output:

```
PS C:\Users\prern> cd "c:\Users\prern\Downloads\JAVA PROJECT\" ; if ($?) { javac CampusChatBot.java } ; if ($?) { java CampusChatBot }
CampusBot: Hi! Ask me anything about our campus (type 'exit' to quit).
You: hi
CampusBot: Sorry, I don't have an answer for that. Try rephrasing.
You: admission
CampusBot: Admissions are open from June to August. Visit the admissions office or our website.
You: thank you
CampusBot: Sorry, I don't have an answer for that. Try rephrasing.
You: bye
CampusBot: Goodbye! Have a great day.
PS C:\Users\prern\Downloads\JAVA PROJECT>
```

✓ Conclusion of CampusChatBot Project

The **CampusChatBot** project is a simple, beginner-friendly Java application that demonstrates how to:

- 🧠 Create a **keyword-based chatbot** using HashMap for storing campus-related FAQs.
- 🖨️ Use Scanner to take real-time user input.
- 🔄 Implement control flow using loops (while) and conditionals (if-else).
- 📁 Organize a basic Java project with clean structure and logical flow.

✓ Outcomes of the CampusChatBot Project

By completing the **CampusChatBot** project, here are the key **learning outcomes** and **skills developed**:

📁 Technical Outcomes:

1. Understanding of Java Basics

- Class & object structure
- main() method
- Access modifiers (public, private, static, final)

Working with Collections

- Learned how to use HashMap to store and retrieve keyword-answer pairs.

User Input Handling

- Took dynamic user input using Scanner.

Control Structures

- Applied if, else, and while loop to manage chatbot flow.

String Processing & Matching

- Used .toLowerCase() and .contains() to compare input with keywords case-insensitively.



Soft Outcomes:

- Improved confidence in Java programming
- Practiced debugging and testing
- Learned to structure small-scale projects
- Gained experience in writing real-world use-case programs



Soft Outcomes:

- Improved confidence in Java programming
- Practiced debugging and testing
- Learned to structure small-scale projects
- Gained experience in writing real-world use-case programs

THANK YOU.....