

**Report
for**

**Term Project:
Rubber Duck Chat Bot**

**BY
Ducky 6**

Aan Nazmus Sakib

Atharva Kadam

Jyoti Suhag

Rohith Kola

Grace Chandler

UNDER THE GUIDANCE OF

Prof. Leo Ureel



Michigan Tech
DEPARTMENT OF COMPUTER SCIENCE

2023

INDEX

Description of the Program	1
Transcripts of a debugging conversation	3
Evidence	7
Methods Used	10
Strengths, Weaknesses, and Limitations	12
Implications and Ethical Considerations	14
References	16

DESCRIPTION OF THE PROGRAM

The Rubber Duck Debugging Assistant, affectionately known as "Ducky," is an interactive Python program designed to assist programmers in debugging their code. Inspired by the concept of explaining code issues to a rubber duck, this AI-powered chatbot provides helpful.

Imagine you're grappling with a bug in your code. Instead of scratching your head alone, you can talk to this chatbot. You throw in your coding problem, and it responds with suggestions and tips. It's a bit playful too – sometimes, it even quacks!

The code is organized into rules, each handling a specific type of programming issue. For instance, if you mention an "ImportError," it understands that you're struggling with missing modules and gives advice on how to fix it.

The bot doesn't stop at programming. It has a personality! You can ask it questions like "sing a song" or "flip a coin," and it responds with enthusiasm. It even handles casual chit-chat, responding to greetings or asking about its name. responses and suggestions based on a wide range of programming problems.

Key Features:

1. Error Resolution Guidance:

Ducky is equipped with a set of predefined patterns to identify and address common programming issues. From syntax errors to runtime exceptions, it offers concise and practical guidance on resolving these problems.

2. Interactive Chat Interface:

The program engages users in a conversational manner, prompting them to describe their programming issues. Users can interact with Ducky by typing queries or statements related to their code troubles.

3. Quirky Touch:

Ducky adds a touch of whimsy by occasionally appending a cheerful "quack" to its responses, injecting a bit of humor into the debugging process.

4. User-Friendly Loop:

After providing a response, Ducky engages the user by asking if they are satisfied or if they would like another solution. This interactive loop aims to ensure users receive the assistance they need.

Usage:

Users can run the program, initiate a conversation by describing their programming problems, and receive thoughtful responses along with occasional duck-inspired humor. The chatbot is designed to make the debugging process both informative and enjoyable.

Test Function:

To verify its effectiveness, the program includes a testing function "test_simplifieducky" that simulates various programming scenarios. This function aids in evaluating Ducky's responsiveness and the accuracy of its suggestions.

TRANSCRIPTS OF A DEBUGGING CONVERSATION

In the transcript section of the report, we dive into the interactions with our code companion, the Rubber Duck AI. This AI, with its friendly demeanor, engages in conversations about programming problems. Users throw coding issues at it, and it responds with helpful tips and suggestions. The back-and-forth between users and the Rubber Duck AI showcases its capability to understand and address various programming issues, making it a valuable debugging tool. The playful touch, including occasional quacks, adds a unique and enjoyable aspect to the debugging process. The transcript captures the diverse range of queries and the bot's responses, highlighting its versatility and user-friendly nature.

You: My program is not working

Rubber Duck AI: Let's try to figure out the issue together. What error message are you getting? quack!

You: I feel frustrated with this bug

Rubber Duck AI: Feeling frustrated is natural. Share more if you'd like. quack!

You: My program is giving an error

Rubber Duck AI: I'm sorry to hear that your program is having issues. Can you provide more details?

You: I'm stuck in an infinite loop

Rubber Duck AI: Look for any missing or incorrect loop termination conditions.

You: I'm having an indentation error

Rubber Duck AI: Incorrect indentation or missing braces can lead to syntax errors. quack!

You: My variable x is not defined

Rubber Duck AI: Make sure there are no typos in the variable name. quack!

You: Do you have any programming tips?

Rubber Duck AI: Comment your code to make it easier to understand and maintain. quack!

You: My code is causing a division by zero error

Rubber Duck AI: Oops! It looks like you are trying to divide by zero. This operation is undefined.

You: I'm getting a TypeError in my function

Rubber Duck AI: If you're getting TypeError, double-check the arguments you're passing to the function or operation. quack!

You: I have a SyntaxError on line 10

Rubber Duck AI: Oops! It seems there is a syntax error in your code. Check for proper indentation and look for any missing or misplaced parentheses.

You: I got a NameError in my script

Rubber Duck AI: NameError warning! 'Check for typos or make sure you've declared the name before using it,' advises the user, 'and make sure the name you're using is correctly defined in your code. quack!

You: I can't resolve this ImportError

Rubber Duck AI: Encountering an ImportError? It could be because the module you're trying to import isn't installed.

You: I'm facing a RuntimeError in my application

Rubber Duck AI: To resolve the RuntimeError, consider analysing the specific conditions under which the error occurs and adjust your code accordingly. quack!

You: There's an `AttributeError` in my object

Rubber Duck AI: When facing an `AttributeError`, review the documentation for the object to understand its attributes and methods.

You: I'm getting a `KeyError` when accessing a dictionary

Rubber Duck AI: Oops! You've stumbled upon a `KeyError`. Make sure the key you're using to access a dictionary exists in the dictionary. quack!

You: I can't find my file, `FileNotFoundError` thrown

Rubber Duck AI: It looks like you're dealing with a `FileNotFoundError`. Verify that the file path you provided is correct and the file exists.

You: My script is running out of memory, `MemoryError`

Rubber Duck AI: `MemoryError` alert! Review your code for any memory-intensive operations and explore ways to reduce memory usage. quack!

You: I'm getting a `ValueError` in my calculation

Rubber Duck AI: When encountering a `ValueError`, examine the input data and make adjustments to align with the expected format or range. quack!

You: My assert statement failed, `AssertionError`

Rubber Duck AI: When facing an `AssertionError`, carefully examine the conditions asserted in your code and adjust as needed.

You: My script stopped due to a `KeyboardInterrupt`

Rubber Duck AI: `KeyboardInterrupt` alert! Investigate if your program contains infinite loops or code that may prevent it from responding to interruptions.

You: I'm experiencing an `OverflowError` in my calculations

Rubber Duck AI: To avoid `OverflowError`, ensure that your numeric computations do not exceed the maximum or minimum values allowed for the data type.

You: My file access mode is wrong, `FileNotFoundError`

Rubber Duck AI: Encountering a `FileNotFoundError`? Ensure that the file you're attempting to open is accessible from your code.

You: I'm having an `OSError` while trying to access a file

Rubber Duck AI: `OSError` alert! Review the error message details to understand the specific nature of the operating system-related issue.

You: I'm facing a `UnicodeError` with string encoding

Rubber Duck AI: To resolve `UnicodeError`, ensure that your code properly handles Unicode characters, and consider using the correct encoding/decoding methods. quack!

You: I hit a `RecursionError` in my recursive function! I'm facing a memory leak in my application

Rubber Duck AI: Oops! You've hit a `RecursionError`, which happens when the maximum recursion depth is exceeded in a recursive function. Check your recursive function for proper termination conditions.

You: My app has a deadlock issue

Rubber Duck AI: Concurrency issues like deadlocks and race conditions require careful handling of threads. Are your locks well managed?

You: My database query is too slow

Rubber Duck AI: Indexing might help improve your query performance. quack!

You: I'm having trouble with API requests

Rubber Duck AI: Check if your API endpoints are correctly defined and the server is responsive. quack!

You: I keep getting a null reference exception

Rubber Duck AI: Consider using 'null' checks or optional chaining to prevent these errors.

You: How do I debug a regex problem?

Rubber Duck AI: Ensure your regex syntax is compatible with the language you're using. quack!

You: I need help with cross-platform compatibility

Rubber Duck AI: When coding for cross-platform compatibility, consider the different file paths and environment settings. quack!

You: My project has a dependency issue

Rubber Duck AI: Check if there are any version conflicts between your libraries. quack!

You: How do I handle a security vulnerability in my code?

Rubber Duck AI: Be aware of common web vulnerabilities like XSS and CSRF. Validate and encode data appropriately. quack!

You: My mobile app crashes on Android

Rubber Duck AI: Ensure compatibility with different OS versions and screen sizes. quack!

You: I'm struggling with a race condition

Rubber Duck AI: Make sure shared resources are properly synchronized to avoid race conditions.

In []:

EVIDENCE

Evidence 1:

User encountered an import module error, asked for clarification, and the chatbot provided a detailed explanation leading to successful resolution by the user.

```
Hello! I'm your friendly Rubber Duck AI. Quack! How can I help you debug today?
You: Hey Rubber duck, what is this import error?? It buggs me while executing
Rubber Duck AI: Looks like you are trying to import a module that Python can't locate. Double-check the module name
for any typing mistakes and make sure it's installed in your Python environment. quack!
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
fixedddd!!!! Thanks. I'm satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: exit
Rubber Duck AI: Goodbye! Keep quacking at those bugs!
```

Evidence 2:

The user encountered an indentation error preventing code execution. The Chat bot provided tips until the user successfully resolved the issue and expressed satisfaction.

```
Hello! I'm your friendly Rubber Duck AI. Quack! How can I help you debug today?
You: Hello, I'm getting indentation error
Rubber Duck AI: Check if you have balanced braces or parentheses in your code. quack!
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
another
Ducky: Make sure your code has consistent indentation. Mixing spaces and tabs can cause issues.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: exit
Rubber Duck AI: Goodbye! Keep quacking at those bugs!
```

Evidence 3:

The user initially faced an infinite loop issue, resolved it, and subsequently encountered another error, which was successfully addressed with assistance from the chatbot.

```
Hello! I'm your friendly Rubber Duck AI. Quack! How can I help you debug today?
You: Hi can you help resolve infinite loop
Rubber Duck AI: Look for any missing or incorrect loop termination conditions. quack!
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
another
Ducky: Check the loops in your code. Make sure there's a proper exit condition.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: Now I'm getting new error, it is file not found
Rubber Duck AI: Encountering a FileNotFoundError? Ensure that the file you're attempting to open is accessible from
your code.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another):
satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: exit
Rubber Duck AI: Goodbye! Keep quacking at those bugs!
```

Evidence 4:

The chatbot will exit only when the user requests it, it will provide resolution to any number of user issues. In the below scenario the user asked the bot six different issues he's facing and got resolution.

```
Hello! I'm your friendly Rubber Duck AI. Quack! How can I help you debug today?
You: Hi, I'm facing issue with ZeroDivisionError
Rubber Duck AI: Oops! It looks like you are trying to divide by zero. This operation is undefined.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): another
Ducky: Oops! It looks like you are trying to divide by zero. This operation is undefined.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: also issue with FileNotFoundError
Rubber Duck AI: It looks like you're dealing with a FileNotFoundError. Verify that the file path you provided is correct and the file exists.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: I'm now getting OSError
Rubber Duck AI: Oops! An OSError suggests there's a generic operating system-related error. Examine the specific details provided in the error message for more information. quack!
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): another
Ducky: Oops! An OSError suggests there's a generic operating system-related error. Examine the specific details provided in the error message for more information.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: now I'm facing issue with MemoryError
Rubber Duck AI: Uh-oh! A MemoryError indicates your program has run out of available memory. Consider optimizing your code or using more efficient data structures.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: still I'm facing issue, now with new issue it is AttributeError
Rubber Duck AI: It seems you're encountering an AttributeError. Check if the attribute you're trying to access exists for the object.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): another
Ducky: AttributeError alert! Ensure that the object you're working with has the attribute you're trying to access.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: It's a new error now, RecursionError
Rubber Duck AI: RecursionError alert! Review your recursive function to ensure it has proper termination conditions and doesn't exceed the maximum recursion depth. quack!
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): another
Ducky: Oops! You've hit a RecursionError, which happens when the maximum recursion depth is exceeded in a recursive function. Check your recursive function for proper termination conditions.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: Now, it's API error
Rubber Duck AI: Check if your API endpoints are correctly defined and the server is responsive.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): another
Ducky: Check if your API endpoints are correctly defined and the server is responsive.
Ducky: Are you satisfied with the response, or do you want another solution from the same rule? (satisfied/another): satisfied. No errors now thatnks
Thank you! Please tell me if you have any more queries or type 'exit' or 'bye' or 'quit' to exit!!!
You: exit
Rubber Duck AI: Goodbye! Keep quacking at those bugs!
```

METHODS USED

1. `"__init__"` Method:

Description: This method initializes an instance of the 'Simpleducky' class.

Purpose: It sets up the initial state of the Rubber Duck Debugging Assistant, preparing it for interactions.

2. `"get_response"` Method:

Description: The `"get_response"` method analyzes user input and matches it against predefined patterns to generate appropriate responses.

Purpose: This is the core method responsible for interpreting user queries and selecting relevant responses from the predefined patterns.

3. `'chat'` Method:

Description: The `'chat'` method facilitates an interactive conversation between the user and the Rubber Duck AI. It continuously prompts the user for input and provides responses until the user decides to exit.

Purpose: This method orchestrates the flow of the interactive chat session, ensuring a dynamic and engaging debugging experience.

4. `"test_simpleducky"` Method:

Description: The `"test_simpleducky"` method assesses the Rubber Duck Debugging Assistant's performance by simulating various programming scenarios.

Purpose: It serves as a testing function, validating the effectiveness of the debugging assistance provided by Ducky in different situations.

5. "__main__" Block:

Description: The "__main__" block contains the entry point of the program, creating an instance of 'Simpleducky' and initiating the chat.

Purpose: It ensures that the Rubber Duck Debugging Assistant is executed when the script is run, starting the interactive debugging session.

6. Patterns and Responses:

Description: The code includes a dictionary named `patterns.responses` containing regex patterns as keys and corresponding lists of responses as values.

Purpose: These patterns define the types of user queries that Ducky can recognize, allowing it to provide tailored responses based on common programming issues.

7. Interactive Loop:

Description: The code includes an interactive loop within the 'chat' method that seeks user feedback after each response.

Purpose: This loop enhances the user experience, allowing them to express satisfaction or request additional solutions, creating a conversational and iterative debugging process.

STRENGTHS, WEAKNESSES, AND LIMITATIONS

Strengths:

- The program demonstrates a creative and interactive approach to debugging by emulating the Rubber Duck Debugging method. This engages users in a conversation, potentially helping them think through and solve programming issues on their own.
- The use of regular expressions for pattern matching allows the program to recognize a variety of programming-related queries, covering a broad range of potential issues.
- The program includes a testing function "test_simplifieducky" that assesses the assistant's responses in different scenarios, contributing to a more robust and reliable debugging tool.

Weaknesses:

- The reliance on predefined patterns may limit the program's ability to understand highly nuanced or context-specific queries. It may struggle with queries that deviate significantly from the recognized patterns.
- The program lacks true natural language understanding and learning capabilities. It operates based on fixed patterns and responses, making it less adaptable to novel or complex programming scenarios.

- While the interactive loop allows users to request additional solutions, the program's ability to provide genuinely diverse and contextually relevant responses might be limited.

Limitations:

- The program does not have awareness of real-time information or the ability to access external resources. It may not provide up-to-date solutions for evolving programming languages or frameworks.
- The chatbot's understanding is constrained to the predefined patterns, and it may not handle ambiguous queries well. Ambiguities or unanticipated scenarios might lead to less helpful responses.

Natural Language Understanding and Accuracy:

- The natural language understanding is basic, relying on regular expressions for pattern matching. It excels in recognizing common programming issues but may struggle with more intricate language constructs or non-standard queries.
- The accuracy of responses is dependent on the specificity of the predefined patterns. For queries that align with recognized patterns, the responses are likely accurate. However, for unique or highly specific issues, the accuracy may decrease.

In summary, the program provides a fun and interactive debugging experience but is best suited for addressing common programming issues within the scope of its predefined patterns. While it lacks advanced natural language understanding, it compensates with simplicity and engagement. The testing function contributes to its reliability, but users should be aware of its limitations in handling highly unique or evolving programming challenges.

BROADER IMPACTS, HOW IT FITS INTO MODERN AI RESEARCH, AND ITS ETHICAL IMPLICATIONS

Broader Impacts:

The program's approach, mimicking the Rubber Duck Debugging method in a conversational manner, has the potential to make programming assistance more accessible and engaging for individuals, especially those new to coding. It fosters a creative and interactive debugging process.

Fit into Modern AI Research:

In the landscape of AI research, this program represents a simplified and creative application of natural language processing (NLP) concepts. It uses regular expressions for pattern recognition, a fundamental technique in NLP, to understand and respond to specific queries.

Ethical Implications:

- The program aligns with ethical principles by providing assistance and guidance in a lighthearted manner. However, its limitations in true natural language understanding and learning capabilities emphasize the importance of users being aware of the program's constraints.
- It raises ethical considerations related to transparency. Users should understand

that the program operates on predefined patterns and lacks the ability to evolve or learn from interactions. This transparency is crucial for managing user expectations.

- The program's playful nature may contribute positively to the user experience, making the learning process more enjoyable. However, there's a need for vigilance in ensuring that users do not overly rely on the program for complex problem-solving, as it may not cover all the intricacies of programming challenges.

In summary, the program's impact is primarily on accessibility and engagement in programming learning. Its simplicity aligns with ethical principles, emphasizing transparency and user awareness. While it doesn't push the boundaries of advanced AI research, it contributes to making AI-assisted programming support more approachable and enjoyable.

REFERENCES

1. Russell, Stuart J., and Peter Norvig. Artificial intelligence a modern approach. London, 2010.
2. VanLehn, K., Jones, R. M., and Michelene T. H. Chi. (1992). A Model of the Self-Explanation Effect. *The Journal of the Learning Sciences*, 2(1), 1–59.
<http://www.jstor.org/stable/1466684>
3. GeekforGeeks <https://www.geeksforgeeks.org/> [accessed on 06-12-23]
4. Stack Overflow <https://stackoverflow.com/> [accessed on 06-12-23]