A blue and yellow crest with a bird

AI-generated content may be incorrect.

**The University of Technology, Jamaica**

**The School of Computing and Information Technology**

**CIT4004 – Analysis of Programming Languages**

**Lecturer: Dr. David White**

**Kemar Christie 2005904**

**Roberto Davis**

**Dwayne Gibbs**

**Tyoni Davis**

**Danielle Jones**

**April 5, 2025**

**Group Project**

**Project Report**

* **Paradigm the language you developed belongs to.**The programming language developed in this project adheres to the procedural programming paradigm. This paradigm is centered around the concept of procedure calls, where the program is structured into procedures or routines that operate on data. It emphasizes a step-by-step sequence of instructions, making it suitable for tasks that require a clear and logical flow of control.
* **Explaining whether your language is general purpose or domain specific.**Our booking programming language is domain-specific. It is designed solely to facilitate booking operations such as scheduling, availability tracking, reservation processing, and user management. Because it focuses on this specialized area and lacks the broad capabilities of a general-purpose language, it cannot be used effectively outside the booking domain.
* **Explaining whether your language is low level or high level.**

The programming language developed in this project is classified as a high-level language. This classification is based on the fact that it was implemented using Python, a widely recognized high-level programming language, along with parsing tools such as PLY (Python Lex-Yacc). As such, our language prioritizes readability, maintainability, and ease of use , hallmarks of high-level language design.

* **Correct grammar for the language you developed.**

START : COMMAND SYMBOL

COMMAND : LIST\_COMMAND  
 | BOOKING\_COMMAND

| RESERVATION\_COMMAND

| CONFIRMATION\_COMMAND  
 | CANCELLATION\_COMMAND

| PAYMENT\_COMMAND

| INQUIRY\_COMMAND

* **Complete parse tree/AST for a sample program in your language.**sample\_input\_data = “Book a Ticket from Montego Bay to Miami on February 17, 2025 at 8:30 AM returning on March 17, 2025 at 8:30 AM.
* **Full list of tokens for the language you developed.**

**Tokens:**

* ACTION\_KEYWORD
* CONTEXT\_KEYWORD
* LOCATION\_MARKER
* CONNECTIVE\_WORD
* DATE
* START\_DATE
* END\_DATE
* NUMBER
* SYMBOL
* MONEY
* RESOURCE
* CONDITIONS
* TIME
* USERNAME
* DEPARTURE
* ARRIVAL
* LOCATION
* SERVICE
* ARTICLE\_CONJUNCTION
* **Regular expressions you used to recognize all the tokens for the language you developed.**

**Regular Expressions Used:**

* ACTION\_KEYWORD
  + action\_keywords = ['List', 'Book a', 'Confirm a', 'Pay', 'Cancel a', 'Reserve a', 'How many', 'Duration of']
  + t\_ACTION\_KEYWORD = r'\b(?:' + r'|'.join(action\_keywords) + r')\b'
* CONTEXT\_KEYWORD
  + context\_keywords = ['on', 'For', 'Schedule', 'are there', 'Returning', 'cost']
  + t\_CONTEXT\_KEYWORD = r'\b(?:' + r'|'.join(context\_keywords) + r')\b'
* LOCATION\_MARKER
  + location\_markers = ['in', 'at', 'from', 'to']
  + t\_LOCATION\_MARKER = r'\b(?:' + r'|'.join(location\_markers) + r')\b'
* CONNECTIVE\_WORD
  + connective\_words = ['that']
  + t\_CONNECTIVE\_WORD = r'\b(?:' + r'|'.join(connective\_words) + r')\b'
* DATE
  + t\_DATE = r'(?<=\bon\s)((?!\b(?:in|at|from|to)\b).)+?(?=\.)'
* START\_DATE
  + t\_START\_DATE = r'(?<=\bfrom\b\s).+?(?=\s\bto\b)|(?<=\bon\b\s).+?(?=\s\bat\b)'
* END\_DATE
  + t\_END\_DATE = r'(?<=\breturning on\s).+?(?=\s(?:at)\b)|' \

r'(?<=\bto\s).+?(?=\s(?:for)\b)|' \

r'(?<=\bto\s).+?(?=\.)'

* NUMBER
  + t\_NUMBER = r'\b\d+\b'
* SYMBOL
  + t\_SYMBOL = r'\.+(?=[ \t]\*$)|,|:'
* MONEY
  + t\_MONEY = r'\$\d+(\.\d+)?'
* RESOURCE
  + t\_RESOURCE = r'Reservations|Reservation|Tickets|Ticket|tickets|Flights|Flight|Rooms|Room|Hotels|Hotel'
* CONDITIONS
  + t\_CONDITIONS = r'\b(?:less than|more than|equal to|greater than|if|then)\b'
* TIME
  + t\_TIME = r'\b(?:([0-9])?[0-9]):[0-9][0-9]\s\*(?:AM|PM)\b'
* USERNAME
  + t\_USERNAME = r'(?<=\bfor\b\s)[A-Za-z0-9\_]+'
* DEPARTURE
  + t\_DEPARTURE = r'(?<=\bfrom\b\s)([a-zA-Z\s]+?)(?=\s\band\b)|(?<=\bFrom\b\s)([a-zA-Z\s]+)(?=\s\bTo\b)|(?<=\bFrom\b\s)([a-zA-Z\s]+)(?=\s\bThat\b)|(?<=\bFrom\b\s)([a-zA-Z\s]+)(?=\s\*\.)|(?<=\bFrom\b\s)([a-zA-Z\s]+)(?=\s\b(?:' + r'|'.join(all\_keywords) + r')\b)'
* ARRIVAL
  + t\_ARRIVAL = r'(?<=\bTo\b\s)([a-zA-Z\s]+?)(?=\s\bFrom\b)|'\

r'(?<=\bTo\b\s)([a-zA-Z\s]+?)(?=\s\*\.)|'\

r'(?<=\bTo\b\s)([a-zA-Z\s]+?)(?=\s\b(?:' + r'|'.join(all\_keywords) + r')\b)'

* LOCATION
  + t\_LOCATION = r'(?<=\bin\b\s)([a-zA-Z\s]+?)(?=\s\b(?:' + r'|'.join(all\_keywords) + r')\b)|'\

r'(?<=\bin\b\s)([a-zA-Z\s]+?)(?=\s\*\.)'

* SERVICE
  + t\_SERVICE = r'(?<=\ba\s)(?!(?:' + r'|'.join(all\_keywords) + r')\b)([A-Za-z]+(?:\s[A-Za-z]+)?)(?=\s(?:' + t\_RESOURCE + r')\b)|'\

r'(?<=\bList\s)([A-Za-z]+(?:\s[A-Za-z]+)?)(?=\s\bSchedule\b)|'\

r'(?<=\bfor\s)([A-Za-z]+(?:\s[A-Za-z]+)?)(?=\s\bfor\b)|'\

r'(?<=\bat\s)([A-Za-z]+(?:\s[A-Za-z]+)?)(?=\s(?:From|from)\b)'

* ARTICLE\_CONJUECTION
  + t\_ARTICLE\_CONJUNCTION = r'\b(a|and)\b'
* **Demonstration of scope and binding in sample code written in your programming language.**
* **Details on the programming language you used to develop your compiler.**
* **Two characteristics of a good programming language (from those you studied in class) that are evident in your designed programming language, and examples of how do these characteristics affect the readability, writability and reliability of your designed programming language.**