Outer Banks Compiler (OBX Compiler)

A Custom Programming Language Inspired by the Show Outer Banks

1. Introduction

The Outer Banks Compiler, also known as OBX Compiler, is a custom-built compiler that translates a domain-specific programming language inspired by the show Outer Banks into intermediate code.
It uses Flex for lexical analysis and Bison for parsing source files with the .obx extension.
This compiler introduces unique syntax using keywords like pogues, hideout, trade, and ward, allowing users to write programs in a thematic style while learning the fundamentals of compiler design.

2. Purpose & Inspiration

Why This Compiler?

- Educational Tool: Demonstrates how lexical analysis, parsing, and semantic actions work
- Theme-Based Design: Keywords are inspired by characters and elements from the TV series Outer Banks
- Hands-on Learning: Provides a practical implementation of Flex and Bison in real-world compiler development

Inspirations Behind the Keywords

Keyword	Inspired By	Use in Language
pogues	POGUES Crew	Start of program block
hideout	Secret Hideouts	Block delimiter
trade	Trading Secrets	Print statement
ward	Sheriff Ward	Conditional (if)
topper	Topper	Else block
cleo	Cleo	Loop initialization
search	Treasure Hunt	Function definition
dive	Underwater Diving	Function call

3. Compiler Overview

Technologies Used

- Flex - For lexical analysis

Outer Banks Compiler (OBX Compiler)

A Custom Programming Language Inspired by the Show Outer Banks

- Bison - For syntactic parsing
- C Language - For semantic actions and output generation

Output Format

The compiler generates Three-Address Code (TAC) in a file named output.tac.

Example TAC Output:

ASSIGN x 10

PRINT x

4. Keywords and Their Meaning

Keyword	Type	Meaning / Usage
pogues	Program	Marks the start of the main program
hideout	Delimiter	Ends a block (like } or end)
jj	Variable	Declares an integer variable
cleo	Loop	Initializes loop variables
treasure	Assignment	Assigns value to a variable
ward	Conditional	Starts a conditional block (like if)
topper	Conditional	Else block
trade	I/O	Prints a value or string
loot	I/O	Another print-like command
search	Function	Defines a function
dive	Function	Calls a defined function
highwater	Comparison	Greater than operator
eq	Comparison	Equality check

5. Grammar and Syntax Rules

High-Level Structure

program: POGUES HIDEOUT statements HIDEOUT ;

Outer Banks Compiler (OBX Compiler)

A Custom Programming Language Inspired by the Show Outer Banks

Supported Statements:

- jj age; => DECLARE_INT
- treasure age = 42; => ASSIGN age 42
- ward (age highwater 18) ... => Conditional logic
- cleo i = 0; => LOOP_INIT i 0
- search greet hideout ... => Function greet defined
- dive greet(); => CALL_FUNCTION greet
- trade(age); => PRINT age
- trade("Hello"); => PRINT_STRING "Hello

6. How the Compiler Works

Step-by-Step Flow:

1. Lexer (Flex)
   - Tokenizes input using regular expressions
   - Returns tokens to the parser
2. Parser (Bison)
   - Uses grammar rules to parse token stream
   - Executes semantic actions
3. Semantic Actions
   - Emit TAC to output.tac
4. Output Generation
   - .tac file can be interpreted or compiled further

7. Sample Programs

Sample 1: Simple Print

```
pogues hideout
    jj age;
    treasure age = 16;
```

```
    trade(age);
    loot(age);
hideout
```

Output:

```
DECLARE_INT
ASSIGN age 16
PRINT age
PRINT age
```

Sample 2: Conditional Logic

```
pogues hideout
    jj age;
    treasure age = 18;
    ward (age highwater 18) hideout
        trade("Adult");
    hideout
    topper hideout
        trade("Minor");
    hideout
hideout
```

Sample 3: Function Definition + Call

```
pogues hideout
    search greet hideout
        trade("Hello from OBX!");
    hideout
    dive greet();
hideout
```

Outer Banks Compiler (OBX Compiler)

A Custom Programming Language Inspired by the Show Outer Banks

8. Future Enhancements

Feature	Description
Arithmetic Expressions	Support + - * /
Boolean Conditions	Add logical operators
Arrays and Strings	Support complex data
Functions with Parameters	Add arguments to functions
Code Optimization	Improve TAC efficiency
Virtual Machine	Run .tac files
GUI Editor	Visual IDE for .obx

9. Conclusion

The Outer Banks Compiler (OBX) is not just a tool for learning compiler design - it's also a tribute to one of the most thrilling adventure shows ever made.

It demonstrates how to build a working compiler using Flex and Bison while embracing creative, themed syntax and structure.