20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Shree Om Sharma
CB.EN.U4CYS21070
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

Outline

Procedural - C

- Procedural Paradigm: C is a procedural programming language, following the procedural paradigm. It organizes code into procedures or functions that are executed in a sequential manner.
- Imperative Style: The language uses an imperative style of programming, where the emphasis is on describing how a program should achieve a task step by step.
- Functions: The building blocks of C programs are functions. Functions encapsulate a set of statements and can be called with specific arguments to perform a particular task.
- Structured Programming: C supports structured programming, promoting the use of control structures like loops (for, while) and conditional statements (if, switch) to enhance code organization and readability.
- Variables and Data Types: C uses variables to store and manipulate data. It supports various data types, including int, float, char, and more, allowing developers to work with different types of data.
- Pointers: C provides powerful pointer manipulation, allowing direct access to memory addresses. Pointers are widely used for efficient data handling, dynamic memory allocation, and array manipulation.
- Modularity: C code can be organized into modular units using functions. This modular structure enhances code maintainability and reusability.

C

- Procedural Decomposition: Large problems are broken down into smaller, more manageable procedures. Each procedure focuses on a specific aspect of the problem, simplifying the overall development process.
- Low-Level Programming: C allows low-level manipulation of hardware resources, making it suitable for system programming, embedded systems, and developing operating systems.
- No Built-in Support for OOP: C lacks built-in support for object-oriented programming (OOP) concepts like classes and objects, which are common in languages like C++ or Java.
- File I/O: C provides functions for file input and output, enabling the reading and writing of data to files.
- Preprocessor Directives: The C preprocessor allows the inclusion of header files, macro definitions, and conditional compilation, enhancing code flexibility.
- Static Typing: C is statically typed, meaning that variable types need to be declared before compilation, enhancing type safety.
- Memory Management: C requires manual memory management, including alloca and deallocation using functions like malloc and free.

C Program for factorial

```
#include <stdio.h>

// Function to calculate the factorial of a number
int factorial(int n) {
if (n == 0 || n == 1) {
return 1;
} else {
return n * factorial(n - 1);
}
}
int main() {
// Input: Number for factorial calculation
int num;
printf("Enter a number to calculate factorial: ");
scanf("%d", &num);

// Calculate and display factorial
printf("Factorial of %d is: %d", num, factorial(num));
return 0;
}
```

Functional - Haskell

- Purely Functional: Haskell is a purely functional programming language, emphasizing immutability and avoiding side effects.
- Lazy Evaluation: Haskell employs lazy evaluation, which means that expressions are not evaluated until their values are actually needed.
- High-Level and Expressive Syntax: Haskell is known for its high-level and expressive syntax, allowing concise representation of complex ideas and algorithms.
- Automatic Memory Management: The language incorporates automatic memory management and garbage collection, reducing the risk of memory-related issues.
- Strong Type System: Haskell has a strong and statically-typed system with type inference, enhancing type safety without explicit type declarations

Haskell

- Immutability by Default: Variables in Haskell are immutable by default, promoting safety and predictability in code.
- Algebraic Data Types: Haskell supports algebraic data types, providing a concise and expressive way to model data.
- Concurrency with Lightweight Threads: Concurrency in Haskell is implemented through lightweight threads, providing scalability without the need for low-level thread management.
- Focus on Mathematical Abstractions: Haskell places a strong emphasis on mathematical abstractions and correctness in programming.
- Rich Type Class System: The language features a rich type class system, supporting polymorphism and enhancing code flexibility.
- High-Level Abstractions: Haskell provides high-level abstractions, allowing developers to express complex ideas in a concise and readable manner.

Haskell Program for factorial

```
– Function to calculate the factorial of a number
factorial :: Integer -> Integer
factorial 0 = 1
factorial n = n * factorial (n - 1)

main :: IO ()
main = do
– Input: Number for factorial calculation
putStrLn "Enter a number to calculate factorial:"
num <- readLn

– Calculate and display factorial
putStrLn $ "Factorial of " ++ show num ++ " is: " ++ show
(factorial num)
```

Comparison

- Paradigm: Haskell: Functional Programming (Purely functional, lazy evaluation). C: Procedural Programming.
- Memory Management: Haskell: Automatic memory management with garbage collection. C: Manual memory management. Developers need to explicitly allocate and deallocate memory.
- Type System: Haskell: Strong, static typing with type inference. Supports type classes and dependent types. C: Weak, static typing. Requires explicit type declarations.
- Syntax: Haskell: Concise and expressive syntax, emphasizing pattern matching and functional composition. C: More verbose syntax, especially for common tasks like handling strings and arrays.
- Concurrency: Haskell: Built-in support for concurrent and parallel programming through abstractions like Software Transactional Memory (STM) and lightweight threads. C: Requires manual implementation of concurrency using threads and synchronization primitives.

References

- https://www.haskell.org/
- https://www.geeksforgeeks.org/
- https://www.w3schools.com/
- https://www.tutorialspoint.com/
- https://learnyouahaskell.com/
- https://chat.openai.com/