20CYS312 - Principles of Programming Languages
Exploring Programming Paradigms

Assignment-01

Presented by Pushpanth
CB.EN.U4CYS21057
TIFAC-CORE in Cyber Security
Amrita Vishwa Vidyapeetham, Coimbatore Campus

Feb 2024

AMRITA
VISHWA VIDYAPEETHAM

Outline

An **Imperative programming** is a style of programming where the program is constructed by a series of statements that change a program's state. In imperative programming, you explicitly specify the sequence of steps that a program should take to accomplish a specific task.

1. Programs are like a set of instructions or commands.Each command tells the computer to do something specific.
2. The program keeps track of information or data. This information can change as the program runs
3. We plan for things that might go wrong. If something unexpected happens, the program knows what to do, like showing an error message.
4. Control over Memory: we can have more direct control over memory management, which can lead to efficient use of resources.

Objective-C is implemented as an extension of the C programming language,combining object-oriented programming (OOP) features with the procedural capabilities of C.The language was developed by adding Smalltalk-style messaging and OOP constructs to the C language.

1. Dynamic Typing: Objective-C is dynamically typed, meaning that the type of a variable is checked at runtime.
2. Message Passing: In Objective-C, objects communicate by sending messages to each other. This is done using square bracket notation.
3. Objective-C code is typically organized into header files (.h) and implementation files (.m). The header file declares the interface (class definition, methods, properties), while the implementation file contains the actual code.
4. Used in Apple Ecosystem: Objective-C has historically been the primary language for macOS and iOS application development. However, with the introduction of Swift, the usage of Objective-C has diminished.

In **Even-Driven paradigm** the flow of the program is determined by events that occur during its execution. Events can include user interactions, system notifications, or changes in the program's state.

1. Event Handler: An event handler is a function or method that is designed to respond to a specific type of event.
2. Event Loop: The event loop is a central component in event-driven systems. It continuously checks for the occurrence of events and dispatches them to the appropriate event handlers.
3. Callback Function: A callback function is a function that is passed as an argument to another function and is executed after the completion of a specific task or event. In event-driven programming, callbacks are commonly used as event handlers.
4. Event-driven programming often involves asynchronous operations. Instead of waiting for tasks to complete sequentially, the program can continue processing events while waiting for time-consuming operations to finish.

NodeJS is basically used as an open-source and cross platform JavaScript runtime environment.For running the server side applications we use this.For building the I/O intensive applications like video streaming sites ,online chatting applications and many other applications , it is used.

1. It allows developers to use JavaScript, traditionally associated with front-end development, for server-side programming.
2. It is built on the V8 JavaScript runtime engine, which is developed by Google for the Chrome browser. V8 compiles JavaScript directly to native machine code, resulting in fast execution speeds.
3. Node.js operates on a single-threaded event loop. it is highly efficient for handling asynchronous I/O operations.
4. Asynchronous and Non-Blocking: Leverages an event-driven, non-blocking architecture, making it efficient for handling concurrent connections and I/O operations.
5. NPM (Node Package Manager): Large ecosystem of packages and modules available through NPM, enhancing development productivity.

Objective-C:

```
import <Foundation/Foundation.h>
@interface MyClass : NSObject
- (void)method;
@end
import "MyClass.h"
@implementation MyClass
- (void)method
NSLog(@"eeeeeeeeeeeee");
@end
import <Foundation/Foundation.h>
import "MyClass.h"
int main(int argc, const char * argv[])
@autoreleasepool
```

```
MyClass *myObject = [[MyClass alloc] init];

myObject doSomething;

return 0;
```

- In the above code the method is called by using square notation. unlike the other languages we use dot notations (obj.func) to call a method.

- Here, NSLog is used to print output. In other languages we use print,printf.like these there are syntax differences

- Objective-C is dynamically typed, meaning that the type of a variable is determined at runtime. This is in contrast to statically typed languages like Java.

- Compared to languages like Python or Java, Objective-C has a more limited standard library.

- The Objective-C file is stored as "filename.m".

Node.js (JavaScript):

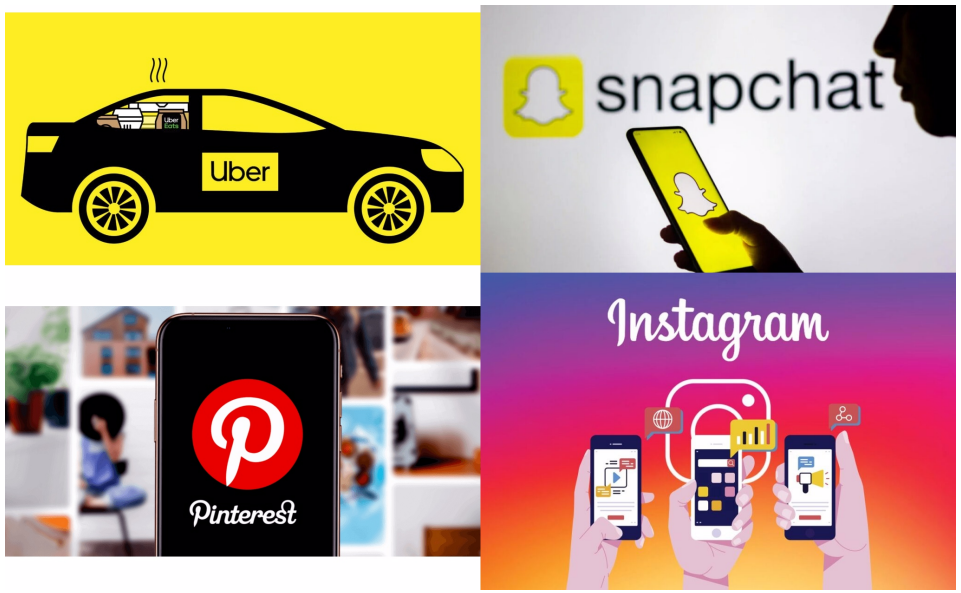
```
// Import the 'http' module
const http = require('http');
// Create an HTTP server
const server = http.createServer((req, res)
// Set the response header with a 200 OK status and content type
res.writeHead(200, 'Content-Type': 'text/plain' );
// Send the response body
res.end('Hello, World!');
);
// Listen on port 3000
const port = 3000;
server.listen(port, ()
console.log(‘Server running at http://localhost:port/‘);
```

);

1. This is a javascript code on server-side application. This code listens to the incoming request and responds with "Hello, World!".

RealWorld Applications

Objective-C:



Node.js:


Walmart
Uber
Medium
NETFLIX
PayPal
NASA
Node stands out with its speed, and intensive data exchange, so, it is not surprising that lots of major and popular industry leaders chose to rely on Node.js

References

https://www.simplilearn.com/tutorials/nodejs-tutorial

https://www.designveloper.com/blog/what-is-objective-c/

https://www.computerscience.org/resources/computer-programming-languages/objective-c