[Functional Programming-22S2](https://my.bucks.ac.uk/webapps/blackboard/execute/courseMain?course_id=_126684_1)

Stefan Allen

Id: 22135474

Functional Programming Paradigm

Functional programming uses mathematical functions to solve problems. It focuses on what is the problem and how to solve it. Functional programming makes it easier to control as it always responds with the same output from an input. Function programming is used to improve the overall code quality and module capabilities while also making it easier to evaluate.

Pure functions, Functional programming uses these with two main functions, they will always give the same output without having any differences. When using an application using this method it makes testing so much easier as it allows the tester to be able to expect a result and not have any differences or variations if the input stays the same.

Recursion, In functional programming, means there is no need for loops, but it used recursion. Recursion works by calling the functions calling themselves until they reach the base cases, this attack can solve a lot of problems while being for accurate than a loop.

Higher order functions, these functions are treated as a first-class variable, where functions can take other functions as an argument or return a function, allowing the code to be more reusable.

Variables and immutable, these variables cannot be altered or changed after they are made, but a new variable can be made. Once it has its values it avoids issues from immutable being and shared while also making it easy to understand the behaviour of the program.

Referential transiency works the same as Variables and is immutable where the variable will not change its values, when the values must be stored a new variable can be created improving the overall programming performance.

Advantages of functional programming:

* Pure functions do not change, and work based on an input but do not give different outputs while being easy to understand.
* It is easier to have one or more events happening at the same time because pure functions do not change outputs also known as concurrency.
* Debugging and testing are earlier as it has uncharted outputs while immutability stops hidden outputs meaning debugging and testing become easier.

Disadvantages of functional programming:

* Having mutable values and recursion may lead to worse performance.
* Although writing pure function is easy combining it with the rest of an application can be a challenge.
* Not being able to use loops can be a bit more complex and can take a while to get correct.

How functional programming can be used to find a solution to a problem?

Functional programming uses functions to solve different problems in code, solving a problem in functional programming can be done by defining the problem and breaking down different steps. In functional programming, you need the data that gets imputed and the result, when breaking down the project it is important to plan the function to make sure each function can solve a specific problem.

Differences between OOP and functional programming.

Function programming works by having a set input and expecting to have a set output meaning functional programming is more dependable and consistent. While OOP work by having different classes and implementing one or more methods while these classes stay by themselves OOP and be easier to learn by can have inconsistent results.

|  |  |
| --- | --- |
| **Functional programming** | **Object-oriented programming** |
| Pure functions | Encapsulation |
| Mutable data | Abstraction |
| Uses recursion for loops | Inheritance |
| Functions work like data | Polymorphism |

Differences between Procedural programming and functional programming.

Procedural programming works around a sequence of instructions, procedure programming works by instructions down to small blocks of code. In procedure programming the developer must make sure each function is called at the correct time and place while also making sure the functions take and hold data while giving an output.

Procedural programming is focused on how to do it while function works by immutability, higher order, functions, and pure functions. In procedure programming it works by having different instructions but functions work by having separate functions but they both work by splitting each code peace into smaller tasks.

|  |  |
| --- | --- |
| **Functional programming** | **Procedural programming** |
| Pure functions | Works as primary functions and variables |
| Mutable data | Not suitable for large apps |
| Uses recursion for loops | Does not provide data security |
| Functions work like data | Works of sequential steps to active a result |

How is functional programming used in other paradigms?

A lot of other paradigms used the function paradigm methods of doing a specific task, and a lot of paradigms use it as all functions are side effect free. Other paradigms also use it to store and process a large amount of data quickly and effectively using a lazy evaluation approach.

Some other paradigms that use functional programming are Scala, f#, erlang and many others. In, some of them use the purely functional programming language approach but also some of them may use multiple programming styles in one language for example ft being a dot net approach uses functional programming but can also use other like object orientated to get the most out of the language to best meet the end users need for the language.

Benefits of using functional programming in other paradigms

As stated at the beginning of the report function programming has many of the unique features that make it different from the other paradigms out there some of these include Pure functions, Recursion, Higher-order functions, Variables, and immutable, Referential transiency. As stated, before functional programming works by having an input and always having the same output meaning when coding an application it could make it more effective and lead to fewer issues in the future also known as immutability.

Functional programming makes a lot of the code reusable as it split the code down into smaller segments making it more manageable and modular. Functional programming is typing the same meaning errors in the system can be found in compile rather than runtime making it more robust and dependable. Functional programming also used high-order functions meaning functions making the code more reusable.

Overall other paradigms may use functional programming to make their code more reusable, dependable, maintainable and many other benefits.

Conclusion

Function programming uses mathematical functions to solve specific problems on what the problem is and how to solve it. Function programming uses pure function, recursion, higher odder function, and immutability all to functional programming as usable and dependable as possible.

As stated throughout the report functional programming has a lot of benefits to it that make it one of the most used paradigms today. Being easy to understand while also having a lot of benefits. Its unique features are why a lot of the paradigms chose to use functional programming or at least include it in the system to have the best possible applications.

In conclusion function programming is one of the best tools for developing software and is lucky to grow and continue in popularity and important for the future to come.

References

### Functional Programming Paradigm - GeeksforGeeks

#### *Functional Programming Paradigm - GeeksforGeeks* (2018). Available at: https://www.geeksforgeeks.org/functional-programming-paradigm/ (Accessed: 29 April 2023).

### Functional Programming: Concepts & Advantages

#### *Functional Programming: Concepts & Advantages* (2023). Available at: https://hackr.io/blog/functional-programming (Accessed: 29 April 2023).

### Functional Programming 101

#### *Functional Programming 101* (2023). Available at: https://github.com/readme/guides/functional-programming-basics (Accessed: 30 April 2023).

### The Complete Coder's Guide to Functional Programming vs OOP

#### *The Complete Coder's Guide to Functional Programming vs OOP* (2022). Available at: https://careerfoundry.com/en/blog/web-development/functional-programming-vs-oop/ (Accessed: 30 April 2023).

### What is the Procedural Programming Paradigm?

#### *What is the Procedural Programming Paradigm?* (2022). Available at: https://programiz.pro/resources/what-is-procedural-programming/ (Accessed: 30 April 2023).