Programming Paradigms – Model Paper (2025 July Batch)

Imagine that you have been tasked with the creation of a programming language. The purpose of the programming language is to replace Python as an academic language. Python has been largely used in the academic world as a quick and expressive language, able to create prototypes and experiments with relative ease on the part of the developer. Ideally, the language should be easy to use for a range of different scientists and engineers, with only minimal experience in programming. Some of the tasks involved will be very CPU-intensive, as the language will be used for tasks such as processing data from radio telescopes or other data-heavy systems. Despite this, performance is not as strictly needed due to the power of modern processors – and anything truly high-performance will go to supercomputing centres anyway.

Any decisions you make will need to be justified. With this context in mind, answer the following questions:

1) Will you develop this language to use dynamic scope, static scope, or both? Justify your decision.

2) As part of this language design, you have been instructed that the language must include 'object orientation'. What parts of object orientation do you propose to implement and why?

3) With respect to the programming characteristics and criteria, highlight which aspects you think are the most critical to focus on for this language.

4) A colleague suggests that you should use C++ as a model of Simplicity. Do you agree or disagree? Why?

5) Which of the two forms of type equivalence would you use for this language and why?

6) What is a 'side-effect' in programming and what are the implications with regards to programming characteristics/criteria?

7) Functional programming languages are often (but not exclusively) linked to data 'immutability'. Explain why data immutability

Without the use of built-in functions, write a function called **print_odd** in Elixir that takes a list of integers as an argument. The function will print out all even integers within the list. You may assume the input will always be an appropriate list. You may use **Integer.is_odd/1** to check whether a given integer is even (:true) or odd (:false). You may write multiple functions if desired. **[8 marks]**

```
Integer.is_odd(5)
# => true
Integer.is_odd(2)
# => false

print_odd([1,2,3,4])
# => 1
#    3
print_odd([2,4,6])
# =>
```

8)

9) Why is traditional logic insufficient to model the real world? Does Prolog suffer from this issue?

10) What is the difference between forward and backward chaining? Use examples to augment your answer.

11) Why is Perl's language design a bad idea? Explain why with reference to two of the Programming Principles.

12) Despite being dynamically-typed, Python is largely considered a strongly-typed language. With an example, provide an explanation for why this is.

13) With respect to the characteristics and criteria taught in the unit, what advantages do scripting languages have over general-purpose languages?

14)

In this scenario, you are being tasked with creating a domain-specific language for providing instructions to the autopilot system of a space ship. The purpose of this system is to allow untrained crew to easily navigate from one moon to the next without having to fly the ship themselves. As part of this system, they should be able to do the following commands:

- List all moons in the area
  - All moons have a name made up of a number, followed by a dash, followed by an alphabetical name.
  - A moon for example may be called 71-Gordion or 41-Titan
- Navigate to a designated moon according to a specified speed
  - Speed must be a numeric value with a precision of 2dp
  - i.e. 2.54 is a valid speed; 2.5 is not
- Filter all moons by their atmospheric conditions
  - Atmospheric conditions are one of: 'Barren', 'Stormy', 'Foggy'