

**Student Name: Abdulla Ibrahem**

**Student ID: 19027109**

**Programming Languages: Dr Matthey Shardlow**

**Software Engineering Frameworks: Prof Rob Aspin**

**Manchester Metropolitan University**

**Examination Period: 22/23 Semester 1**

**Faculty: Faculty Of Science & Engineering**

**Department: Department Of Computing And Mathematics**

**Programme information**

**Programme: COMPUTING AND DIGITAL TECHNOLOGY NETWORK**

**Level of study: 6**

**Examination information**

**Unit code: 6G6Z1115\_2223\_9Z6**

**Paper title: PROGRAMMING LANGUAGES AND SE FRAMEWORKS**

**Exam Duration: 48 hours**

**Question(1)**

**(1A) Name 3 key features of the procedural programming paradigm**

* Structured control flow
* Variables and data types
* modularity

(1B) **Name 3 key features of the Object-Oriented programming paradigm**

* Polymorphism
* Inheritance
* Encapsulation

**(1C) Using appropriate code examples, explain the differences between the procedural and object-oriented paradigms. You should refer to the features you have identified in parts 1 and 2 of this question**.

One of the main differences between procedural and object-oriented programming is the way code is organized. In procedural programming, code is organized into procedural functions, which are collections of functions that perform a specific task. Here’s an example of a procedural program that calculates the area of a circle

Graphical user interface, application

Description automatically generated

In object-oriented programming, code is organized into objects, which are instances of classes. Classes are templates that define the properties and behaviour of a particular type of object. Here’s an example of an object-oriented program that calculates the area of a circle:

Graphical user interface, text, application

Description automatically generated

In this example, the Circle class is the primary unit of the organization. it has two methods: one is the constructor, which initializes the class with a given radius and the second that returns the area of the circle with that radius.

**(1D) Analyse the following code and describe the features that are present in the language. For each feature you identify, you should also explicitly identify the piece of the code that exhibits this feature and name the paradigm that it comes from.**

This code appears to be written in a programming language that is similar to or based on the syntax of a language such as Python or Julia.

Function definition, the code defines several functions, including “Catalan”, “Catalan2”, and “Catalan3”, these are defined using the (def) keyword, and the syntax for defining functions is similar to that in python. def Catalan(n) = choose (2n, n)/(n + 1), def Catalan2(n) product( (n + k)/k |k <- 2...n), def Catalan3(0) def Catalan3(0), def Catalan3(n) function call the code calls the Catalan, Catalan2, Catalan3 functions multiple times with different parameters. This code likely belongs to the imperative programming paradigm, which is characterized by the use of statements that change a program’s state. This code is following the imperative programming paradigm as it uses control flow and assignments to control the program.

**(1E)**

Haskell is a functional programming paradigm that would be well-suited for developing a new power distribution system for the national infrastructure of the UK. This is due to several key features of Haskell that align well with the requirements of the project, such as its emphasis on immutability, strong type system, and support for concurrent and parallel programming.

One of the key features of Haskell that would be beneficial for this project is its emphasis on immutability. In functional programming, variables cannot be reassigned, and data structures are immutable, which means they cannot be modified once they are created. This makes it easier to reason about the behaviour of the program and prevent bugs caused by unexpected side effects.

Haskell also has a very strong type of system that helps catch errors early in the development process. It uses a type of inference system, which means that the programmer does not need to explicitly specify the types of variables and functions, but the compiler can infer them based on the program’s structure. This results in a more concise and readable codebase, making it easier to maintain and modify. Haskell also has built-in support for concurrent and parallel programming. Haskell’s lightweight threading and the ability to use software transactional memory out of the box make it a suitable choice for large-scale projects such as power distribution systems. Concurrent and Parallel programming in Haskell is more manageable because it avoids problems such as race conditions and deadlocks, which can occur when multiple threads of execution access shared resources.

On top of that Haskell has a very powerful, expressive and composable syntax that makes it possible to express complex ideas simply and cleanly. The use of algebraic data types and pattern matching can also help in making code more expressive and readable.

Pure functions, functions in Haskell are defined as mathematical functions, they take input, produce output and do not have any side effects. This can help ensure the system is free from error, as pure functions are much easier to test and debug. Furthermore, it is easy to reason about the flow of data in the system and makes the system more modular and maintainable, which is essential in large-scale systems that are regularly updated.

Haskell has a large and active community, which means that there are many libraries and frameworks available for various tasks, such as data processing, data visualization, and web development. This can help to speed up the development process and reduce the amount of code that needs to be written from scratch.

Haskell’s emphasis on immutability, strong type system, support for concurrent and parallel programming and expressive and composable syntax, makes it well-suited for developing a large-scale, multi-site power distribution system that must run without failure and real-time across multiple servers throughout the United Kingdom.

**Question(2)**

**2. This question relates to the practical side of algorithm design using appropriate programming languages. There are three parts to this question: (a) to (c).**

**(2A)**

Pred\_1([], 1) is not a valid predicate definition in Prolog. It appears to be an attempt to define a predicate named pred\_1, with an empty list and the number 1 as the arguments, but it is messing with the (:-) which is used to specify the predicate’s logic or rule. Pred <(H), ! is also not a valid predicate definition in Prolog. It seems to be an attempt to define a predicate named “pred”, with an argument <(H), followed by a cut “!” and a comma, but it is incomplete and doesn’t make sense in context.

The following screenshot below shows the equivalent solution of the procedural paradigm using python.

Graphical user interface, application, Teams

Description automatically generated

This is the equivalent solution of the procedural paradigm using C.

Graphical user interface

Description automatically generated

**(2B)**

The possible solutions to question 2 part (b) using python and Haskell.

A screenshot of a computer

Description automatically generated with medium confidence

A screenshot of a computer

Description automatically generated with medium confidence

**(2C) Describe the features of each language you have selected to answer question 2b. For each language, describe how the features of that language allowed you to solve the program brief. Compare your solutions in each language.** **Max 750 words**.

Python is a high-level, interpreted, general-purpose programming language that is widely used in industry and academia. It supports multiple programming paradigms, including imperative, object-oriented, and functional programming. One of the key features of Python is its simple, easy-to-read syntax, which makes it an accessible language for beginners and experienced developers alike. Additionally, Python has a vast standard library, which provides a wide range of modules and functions for performing common tasks, such as working with strings, lists, and dictionaries.

In the imperative solution, I used a combination of for loops to iterate over the list of items and generate all possible combinations of 3 items and append them to the final list of triplets. The simplicity of the Python syntax and its built-in list data structure allowed me to express this algorithm in a straightforward, easy-to-read manner. Additionally, the built-in sort of function made it easy to order the items in each sub-list.

On the other hand, Haskell is a purely functional programming language, which means that it does not have any side effects and relies on the application of functions to values. This means that the output of a function depends only on its input, and there is no state to be maintained. It is a statically typed language and has strong type inference, which can make it more robust and easier to reason about code. Haskell also has a powerful type of system and allows algebraic data types, pattern matching and a rich set of list manipulation functions.

In the functional solution, I used recursion to generate all possible sub-lists of the input list, and then I used functions such as concatMap, map, and nub to sort, delete the elements and remove the duplicates. The recursion in this solution can be more elegant to read, but harder to implement in some cases, also the powerful list manipulation functions that Haskell provides can be used to generate the result and make sure the output is correct.

Comparing both solutions, the first one is relatively simple and can be easy to understand for developers with any level of experience, and it's more efficient in terms of time complexity, as it uses loops to generate all combinations, this solution also guarantees that the output is sorted as required. On the other hand, the second solution is elegant and concise but it's also more complex because it uses recursion, it also needs more knowledge of the language and its functions to ensure the output is correct and it takes a longer time to execute.

In conclusion, while both languages can be used to solve the problem at hand, they have different features that make them suitable for different types of solutions. The imperative solution in Python is easy to understand and efficient, while the functional solution in Haskell is elegant and powerful. The choice between them would depend on the problem requirements and the developer's preference.

**SE Frameworks**

***Questions (3)***

**(3A)**

Agile is a methodology for software development that emphasizes flexibility, collaboration, and customer satisfaction. It is based on a set of values and principles outlines in the Agile Manifesto, which prioritizes working software over comprehensive documentation and customer collaboration over contract negotiation.

While Agile does have a defined process, it is not meant to be strictly adhered to. In fact, one of the key tenants of Agile is to be adaptive and responsive to change. The Agile process is designed to be flexible and allow for changes and adjustments throughout the development process.

Adhering strictly to a defined process can lead to inflexibility and can potentially hinder the ability to respond to changing customer needs and requirements. It’s important to understand that Agile is not a size fits all methodology and the team.

Agile provides a framework for software development, but teams must be open to change and adapt their approach as needed to deliver the best value to their customers.

**(3B)**

Agile is a popular methodology for software development that has been shown to be effective in many situations. However, it may not be the best fit for all projects in the modern computing environment.

The main benefit of Agile is its flexibility and ability to adapt to changing customer needs and requirements. This makes it well-suited for projects with high levels of uncertainty or complexity. Agile also promotes collaboration and customer satisfaction.

However Agile may not be the best fit for projects that have strict regulatory requirements or a defined scope that cannot change. In these cases, a more traditional Waterfall approach may be more appropriate.

While Agile can be an effective methodology for many projects in the modern computing environment, it is not appropriate for every project. It’s important to carefully evaluate the specific needs and constraints of a project before deciding on an appropriate methodology.

**(3C)**

Continuous Integration is a software development practice that involves frequently integrating code changes into a shared repository, and then automatically building, testing, and deploying the code. This practice is designed to detect integration errors early and often, which can lead to faster development and fewer bugs. Agile, on the other hand, is a methodology for software development that emphasizes flexibility, collaboration, and customer satisfaction. CI is not a replacement for Agile, it’s rather a practice that supports Agile development. Agile development teams use CI to automate the integration and testing of code changes, which allows them to deliver new features and enhancements to customers more quickly and with fewer bugs.

Agile and CI are not mutually exclusive, but rather complementary practices that can be used together to improve the quality and speed of software development.

**(3D)**

Microservice Architecture is a software development approach that involves building an application as a collection of small, independent services that communicate with each other through well-defined interfaces. This approach has several benefits, including scalability, flexibility, and ease of deployment.

However, microservices are not always the best fit for all projects, regardless of their scale or scope. Microservices can be complex to design and implement, especially for teams that are not experienced with the architecture. Microservices are more suitable for projects that have a high degree of scalability and variability, for projects that are composed of multiple teams and for projects that are composed of multiple services that can be independently developed and deployed.

In conclusion, while microservice architecture can provide many benefits, it’s not always the best fit for all projects. Microservices should be used when they add value to the project, rather than being used for the sake of using them.

**(3E)**

Design patterns are a set of best practices and solutions to common software development problems that have been identified and documented by experts in the field. While they may have originated in the academic setting, they have been widely adopted and used in industry-based software engineering. Many experienced software engineers consider design patterns to be essential tools in their tools, in their toolkits, as they provide a common language and a set of proven solutions that can save time and improve the quality of software development. Design patterns are not only used to solve specific issues but also to improve the overall design of the software by marking it more modular, more flexible, and more maintainable. They help to improve the communication between developers working on a project, and they also help to make the code more readable and understandable.

Design patterns are a valuable tool for software engineers and have been widely adopted in industry-based software engineering.

***Questions (4)***

**(4A)**

Based on this case study provided, I would recommend using a Scrum development methodology to support the XXX project. Scrum is an Agile methodology that provides a framework for managing and delivering complex projects in an incremental and iterative manner. It emphasizes regular review and evaluation of the emerging platform, which aligns well with the company’s goal of rapidly evolving its platform in response to user feedback and changing requirements.

The Scrum methodology includes several key features that align well with specific aspects of the case study. These include;

* List of features and requirements that the development team will work on. This aligns well with the company’s goal of launching a Minimally Viable Product MVC and rapidly revolving it over time to incorporate more service providers, features, and functionality.
* Time-boxed iterations of development, typically lasting 1-3 weeks, where the team delivers a potentially shippable increment of the product. This aligns well with the company’s need for a steady rollout of features and enhancements.
* Planning a meeting where the team plans out the work they will complete during the upcoming sprint. This aligns well with the company’s need for clear visibility of the project.
* A daily meeting where the team members discuss their progress and plan their work for the day.

In the context of the project, the Scrum methodology will operate by first creating a prioritized list of features and requirements (Product Backlog) that the development team will work on. The team will then work on these features in time-boxed iterations (Sprints) and deliver a potentially shippable increment of the product. There will be regular meetings to discuss the progress and plan the work.

**(4B)**

My proposed high-level, architectural approach for the XXX project would be to use a microservices architecture. This approach involves breaking down the project into smaller, independent services that can be developed, deployed and scaled independently.

This approach is well-suited to the study as it allows for a high degree of flexibility and scalability. For example, if a particular service becomes a bottleneck or requires significant updates, it can be isolated and addressed without disrupting the entire system. It also allows for a more modular and reusable codebase, which can simplify the development process.

The advantage of this approach include:

* Service can be scaled independently, allowing for efficient use of resources.
* Services can be developed and deployed independently reducing the risk of introducing bugs or breaking existing functionality.
* Reused services across different parts of the system, reducing development time and costs.

However, there are also some potential downsides to this approach. One potential issue is that it can increase complexity, as they’re now more moving parts to manage. Finally, it can make debugging more difficult, as issues may span multiple services. To mitigate these risks, it is essential to have robust monitoring and logging in place, as well as clear documentation and communication among the development team.

**(4C)**

There are several modern development methodologies that can be used to support the software development process, including Agile, Sacrum, Kanban, Waterfall, and more. Each approach has its own set of tools and techniques that can be used to make the development process more efficient and cost-effective.

For the case study, it would be appropriate to use Agile methodologies as they are well-suited for projects with rapidly changing requirements and tight deadlines. Agile methodologies also promote a collaborative and flexible approach to software development, which would be beneficial in this scenario.

Scrum, a specific Agile methodology, would also be appropriate as it provides a framework for managing and completing complex projects. It involves the use of a cross-functional team, known as the Scrum team, to deliver incremental pieces of working software, which would be beneficial in this scenario.

Waterfall, a traditional development methodology, would not be appropriate for this scenario as it is more suited for projects with well-defined requirements and fixed deadlines. It also does not promote a collaborative and flexible approach to software development which would be beneficial in this case.

Agile methodologies such as Scrum, and Kanban would be appropriate for this scenario as they promote a collaborative and flexible approach to software development and provide a framework for managing and completing complex projects, whereas Waterfall would not be appropriate as it is not well-suited for projects with rapidly changing requirements and tight deadlines.