ICA gui REPORT

Joshua Bentley

Student Number – A0057107

V

Contents

[**S1 - The Advantages and disadvantages of using object oriented(OO) design principles.** 2](#_Toc133494714)

[**S2 - The legal, security and ethical issues surrounding application development.** 3](#_Toc133494715)

[**S3 - Evaluation of my program development – approach to the problem.** 4](#_Toc133494716)

# S1 - The Advantages and disadvantages of using object oriented(OO) design principles.

Firstly, there are many advantages and disadvantages of using OO design principles. I will list them to start and then critically evaluate them.

Advantages:

* Reusability
* Modularity

Reusability is a main advantage of OO programming. It comes from the use of inheritance and composition. The use of these techniques allows developers to write and maintain less code. There are potential drawbacks to this, however. Inheritance can lead to tight coupling and can make it difficult to modify and extend code in the future. If a superclass changes, all subclass that inherit will also need to be updated. As a result, the reusability advantage of OO development becomes more of disadvantage due to the amount of time that would be needed to update code in the future. This would however depend on the needs for the code and whether it will need updating later or not.

Further, too much use of inheritance can lead to complex class hierarchy making code more difficult to understand. This is further amplified if the projects are larger or coded by people who are new to coding.

Another advantage of OOP is the modularity that comes with it. OOP makes it easier to understand and modify a system by breaking it down into smaller objects. While modularity is a key benefit of OOP, it also can require very careful planning and design. OO systems can become very hard to understand when they are not properly designed, so even though modularity is an advantage of OOP, it does come with some drawbacks. Further, although modularity can make it easier to understand code, it may also add another level of complexity. Interactions between objects can become very complex. Behaviours of objects can require understanding of all the objects that interact with it.

Disadvantages:

* Learning Curve
* Performance

A main disadvantage of OOP is the learning curve that comes with it. The difficulty of learning OOP will depend on the persons background and existing knowledge. Personally, OOP has been challenging. Trying to wrap my head around all the design principles and coding techniques has been challenging but very rewarding. The difficulty of learning OOP may be offset by all the many benefits it provides such as maintainability and code reuse.

Secondly, a disadvantage of OOP may be performance. It is important to critically evaluate this point however since OOP does bring so many benefits. The performance impact of creating an object may be so minimal that you don’t notice it. As a result, this point may be outweighed by the benefits that come with OOP like code reuse. Another point is that as time goes on, advances in hardware make the performance in OOP a lot better. Modern compilers and processors can optimise code and reduce the performance impact. As time goes on, the performance will keep lowering itself as a disadvantage.

# S2 - The legal, security and ethical issues surrounding application development.

Main points of the legal, security and ethical issues surrounding application development include the following:

* Data Protection (GDPR): Developers must comply with the General Data Protection Regulation (GDPR) when collecting, processing, and storing personal data. (Data Protection Act 2018). This Act brought the EU’s GDPR into the UK law. It governs personal data rights including the way companies handle your data and the compensation you can claim for misuse of your data. The GDPR requires developers to obtain consent from users before collecting their data. It also requires developers to implement appropriate security measures to protect data.
* Intellectual property: developers must respect intellectual property laws when creating applications. This includes avoiding infringement of patents, copyrights, and trademarks.
* Cybersecurity: appropriate security must be implemented to protect users from cyber-attacks. This includes coding encryption and vulnerability testing.
* Ethical considerations: developers must comply with consumer protection laws. Further, developers must ensure that users are not subjected to any form of discrimination.
* Consumer protection
* Employment law: developers must comply with employment laws when hiring and managing employees.
* Other laws that apply are the Computer Misuse Act 1990 which makes it illegal to gain unauthorized access to computer systems and the Designs and Patents Act 1988 which protects intellectual property rights.

# S3 - Evaluation of my program development – approach to the problem.

My first approach to development was creating the main class, GUI class, and the inherited parent and child classes. I used the Class diagrams to achieve this. Creating the variables to be shared between all the furniture classes with super and the individual child class variables. I also worked on the basic GUI design for the application. A main problem that I ran into was how to use my buttons to store and hold certain data and objects when a user has entered them. I was not sure how to apply my inheritance into my GUI.

Moving forward, I learnt more about how to store variables in objects and how to create them when a button was pressed. A major problem I ran into was the creation of images. I could not get them to work and pop up on the image display grid. Further, some of the math of my program may be wrong too. I was unsure with some variables and how they interacted. Overall, my approach to the problem could have been improved, and will be on my next project. I also believe my GUI to be a decent first attempt at this kind of coding project.