**COEN275 Object-Oriented Analysis, Design and Programming**

**Quiz Application**

**OO Class Concepts Used**

**Contents**

\* OO Class Concepts Used

1. Object Oriented Design
2. Inheritance
3. Polymorphism
4. Abstraction
5. OO Class Concepts used

1. Object Oriented Design

The application's entire architecture is based on object-oriented principles, in which the state and functionalities are logically divided into distinct classes and objects based on their significance and behavior, with no functionality accessible outside of class. The UI of the Java Swing application follows an object-oriented design pattern, with different UI components exhibiting different features such as inheritance, abstraction, and polymorphism, and all child UI components inheriting from a single parent class of the window frame that invokes the Java Swing application.

1. Inheritance

Inheritance is used to pass on common properties from the generic base class Swing to more particular derived classes such as student\_homepage, teacher\_homepage and admin\_home, which can then add their own set of methods and properties in addition to the inherited ones, as well as override some methods based on their implementation.

1. Polymorphism

Polymorphism is a technique for associating appropriate methods to the object for which they are called at runtime. For example, depending on which object is invoked, all CRUD operation methods on the Swing class will call the corresponding CRUD methods on the student or teacher classes.

1. Abstraction

Abstraction is used to disguise the specific implementation, exposing simply an application interface through the Swing class while the implementation specifics are kept in the subclass.