ACIT 2515 – Week 1

OOP stands for Object Oriented Programming

-programming language model organized around objects and data rather than actions and logic

-objects have attributes (data) and behaviour

-objects may correspond to real-world entities or abstract entities

4 pillars of OOP:

-Abstraction

-Encapsulation

-Inheritance

-Polymorphism

Procedural Programming vs OOP

-Procedural programming requires data to be passed between procedures

-OOP objects encapsulates code and data, and interact with other objects

-Procedural programming: Top down design, limited code reuse, complex code, global data focused

-OOP: Object focused design, code reuse, complex design, protected data

OOA/OOD (Object Oriented Analysis/Design)

-Building block approach:

1. identify objects needed in software application

2. Determine interactions&/relationships between objects

3. for each object, determine attributes and behaviours

OOA: identify objects and relationships from end user’s perspective (business perspective)

OOD: identify additional objects, interactions, and constraints from technical perspective

OOP: implementation of design using OO practices (4 Pillars. Abstraction, encapsulation, inheritance, polymorphism)

OOA/OOD – Attributes and Behaviours

|  |  |
| --- | --- |
| Student | Name |
| First\_name  Last\_name  Student\_num  School  courses | Attributes |
| Take\_course  Drop\_course  Change\_school | behaviours |

Benefits of OOP/OOD:

-reuse & recycling within and across software applications

-design – forces better upfront planning and design for larger projects

-testability – at object level

-extensibility – adding new data/behaviour to objects

Drawbacks of OOP/OOD:

-overhead for small projects (extra upfront design)

-potential over-engineering (eg. too few classes, too many classes, overuse of design patterns…)

**Definitions:**

Class:

-defines a general category

-blueprint/template for creating an object

-a custom data type

Attributes/Fields:

-defined data attributes of a class

Methods/Message

-defined behaviours/capabilities of a class

Object/Instance

-specific instance of a class

State:

-current values of attributes in an object

Constructor:

-method that is called when an object is created

Instance Variables:

-variables that contain values specific to an instance of an object

-attributes of the object

Visibility:

-whether method of attribute is public or private

-public: can be used or accessed from clients (eg. external users)

-private: can be accessed and used only within the object

Python Best Practices Basics:

Class Name – CamelCase

Attributes – lower\_case\_with\_underscores

Methods – lower\_case\_with\_underscore