**Advanced Programming 2017 – Year 2**

**Labwork 8: (6% - or 60 points out of 500 points for labwork this semester)**

**NOTE: ALL LABS TO BE COMPLETED IN PROJECTS USING ECLIPSE**

**NOTE: YOU MUST USE YOUR OWN EXAMPLES FOR THESE EXERCISES, I.E., YOU CANNOT RE-USE LECTURE EXAMPLE(S) AS SUBMISSIONS (THESE WILL BE GIVEN ZERO MARKS\POINTS)**

**REVISION EXERCISE 4 – (10 points)**

Create a project called **Revision4**. Create and abstraction of a CreditCard with the attrtibutes, nameOnCard, expiryDate, dateOfIssue and cvnNumber (3 digit verification code). Create at least three **non-blank overloaded constructors** for the CreditCard class for constructing objects. Test all three constructors in a test class called **CreditCardTester**.

Required activities and marking guideline:

* Create abstraction with attributes (4 points)
* Write three constructors (all non-blank and overloaded) (3 points)
* Test the three overloaded constructors in test class (3 points)

**Part 1 – Polymorphism: use an interface instead of a class (15 points)**

Create an Eclipse Project called **Lab8Part1**. Re-create the **Gun** hierarchy example used in the lecture EXCEPT use an interface for the Gun instead of an abstract class (include sub-classes Revolver and Semi-automatic and a test class to create your own objects of each type and print out their details using **polymorphism** and **dynamic binding**). Add a new sub-class to the hierarchy called **Cannon** that has the special loading instructions for a cannon, e.g., “place wading into the cannon, place cannon ball into cannon”. Edit the test class to include some cannon objects and re-run.

Required activities and marking guideline:

* Gun interface implemented (3 points)
* Revolver class implementing interface (2 points)
* Semi-automatic class implementing interface (2 points)
* Add the new cannon sub-class (5 points)
* Class to create Gun objects and print details using polymorphism (3 points)

**Part 2 – Polymorphism example (15 points)**

Create an Eclipse Project called **Lab8Part2**. Create your own hierarchy to represent polymorphism based on some everyday objects of which there are many types (think of something from you own life, e.g., sport or hobbies). Create at least one generic superclass and at least two specific subclasses. Set specific attributes for the subtypes and create a polymorphic method for output of the specific information that can be placed in a generic Vector<Superclass>. Create a test program to output a list of ten of the objects in your hierarchy and use dynamic binding to output the details of the objects from the generic vector.

Required activities and marking guideline:

* Superclass created (3 points)
* First sub-class with specific attributes\behaviours (3 points)
* Second sub-class with specific attributes\behaviours (3 points)
* Create at least 10 objects of the sub-types in test class vector (3 points)
* Demonstrate dynamic binding by outputting objects in vector (3 points)

**Part 3 – Applying Polymorphism with images and GUI (20 points)**

Create an Eclipse Project called **Lab8Part3**. Create a class called **MyAlbum** which will display your favourite images and some information about what is shown in the image in a GUI with a JLabel (e.g., Animals and Sports) by applying polymorphism using a polymorphic methods called **getImage()** and **getFacts()**. The facts listed for each sub-type can be displayed in a JTextArea added to the GUI. Set specific attributes for each sub-type. Sports can have the attributes originOfSport and mainScoreType (e.g. “goal” for soccer). Your program will provide a superclass called **AlbumItem** and the two subclasses (containing concrete implementations of the polymorphic getImage() and getFacts() methods e.g. **FavouriteSport** and **FavouriteAnimal** classes). The MyAlbum class will store a Vector of **AlbumItem** objects used for displaying in the JLabel by calling the polymorphic **getImage()** method. The GUI will provide a forward button to page through your favourite images and facts relating to each of the images currently being displayed. (Note: Do not use large images for this project it will just needlessly make the project submission too large.)

Required activities and marking guideline:

* Create the abstract AlbumItem class (3 points)
* Create the 1st subclass (e.g. FavouriteSport) – with constructor (4 points)
* Create the 2nd subclass (e.g. FavouriteAnimal) – with constructor (4 points)
* Create the MyAlbum class with Vector (3 points)
* Create at least 6 subclasses of each and add to Vector (3 points)
* Use **dynamic binding** to display images and facts (3 points)