Soft 7004 – OOP1 - Labs

## LAB 3: Completion Date: 2nd November 2018

## On completion please zip up your files including any documents used for drawing the class diagram etc. Upload to BlackBoard. This zip file should include all Labs.

**Q1**

Part a) What follows is the main class called TestObjectList. Write this code into Eclipse and comment each line.

/\*\*

\* Write a description of class TestObjectList here.

\*

\* @author (your name)

\* @version (a version number or a date)

\*/

public class TestObjectList

{

public static void main(String[] args)

{

ObjectList ol = new ObjectList(3); // using array as it is static, note fixed length of 3

String s = "Im Happy";

Dog d = new Dog();

DVD v = new DVD();

Integer i = 1234;

System.out.println(ol.add(s));

System.out.println(ol.add(d));

System.out.println(ol.add(v));

System.out.println(ol.add(i));

ol.remove(0);

System.out.println(ol.add(i));

System.out.println("Is the list full? "+ isFull());

System.out.println("Is the list empty? "+ isEmpty());

System.out.println("Total number of objects in the list: " + getTotal());

Object g = ol.getObject(1);

g.bark(); // is it safe to do this?

}

}

Part b) Draw the class diagram for the ObjectList class. When complete write the class you created. Remember to run your test programme again when completed and remove any remaining errors. The ObjectList user an array.

**Q2**

Part a) Write code for a class Dog. A Dog object is to have attributes name, age and address.

Part b) Write code for a class Flea. A Flea object is to have attributes name and age.

Part c) Give any additional code in the Dog and Flea classes that is required to setup a bidirectional association between a Dog object and a Flea object. A Dog object acts as an

owner for a flea object and the Flea object acts as a parasite for the Dog object.

Part d) Add new sections to your Dog class so that a Dog object can act as owner for up to 20 Flea objects.

**Use comments to define each section so that all sections are presented**.

**Q3**

Part a) Write the code for the following class. Include all sets and gets:

Book

title: String

ISBN: int

author: String

price: double

Book(String, int, String, double)

…………

getPrice(): double

…………

Part b) In order to write the code for the class below you have to create another class.

What class must you create?

Once known create the UML for this dummy class and include only methods you need. No need for gets or sets.

Now create that dummy class and the following class below.

# Lecturer

name: String

ID: int

books: BookList // use inheritance

MAXNOOFBOOKS: int (15 books)

Lecturer : (String , int)

getName () : String

getID () : int

addBook *(Book)*

getBookList () : BookList

……………

toString(): String

print()