

**INFS1609 Assignment 2 (15%)**

| Revision | Date       | Changes         |
|----------|------------|-----------------|
| 1        | 02/04/2019 | Initial release |

**Assignment Design**

- This assignment is to be undertaken as an **individual** assignment
- This assignment is graded upon 15 marks and counts for **15%** of your **Overall Marks** for this course
  - This assignment contains 2 parts:  
Student Gradebook and Bike Repair Shop
- The assignment is due on **Friday 26<sup>th</sup> April 2019, by 1200hrs (noon)**
  - Do not submit a second submission after the due date or it will be counted as late
- The assignment must be submitted electronically via **Ed > Assessments**
  - Submission requirements are outlined in each respective part
- Test cases are used to do a first-round marking of your code. You should try to run your program on Ed to check if they pass the test cases. Test run your code **as early as possible** because you might need to make changes to your code.
- Please use the Ed discussion forum to discuss any issues related to this assignment.
- The readability of your code is one of the marking criteria. You should take care of your coding style and include comments in your code (wherever appropriate) to help explain it

***Please make sure you have read the information about UNSW Business School protocols, University policies, student responsibilities and education quality and support on your Course Outline:***

<https://www.business.unsw.edu.au/degrees-courses/course-outlines/archives/INFS1609-2018-S2#policies>

*If you have any questions about interpreting the assignment and its requirements, please make use of the LICs consultation sessions. To avoid confusion and misunderstanding, we will not be answering assignment-related questions over email.*

## Part 1 – Student Gradebook (5%)

Implement a simple student gradebook which prompts the user to enter the student's name, faculty, start year and grades. Your program should print the student's WAM (average of grades entered), and summary data of grades (defined by <https://student.unsw.edu.au/grade>).

You will need to design a Student class which contains the appropriate attributes to store all of the information. You must create the appropriate getter, setter and constructor methods.

We will provide you with a TestStudent class, this class is responsible for prompting the user for input and displaying the output as required below. Text written in red in the samples below indicates user input. Round to 2 decimal places when required, if there are no decimals do not show a decimal point.

You can assume that all values entered are valid.

```
Enter student's name: Jess
Enter student's faculty: Business
Enter student's year: 2018
Enter student's grades: 75,75,76,74

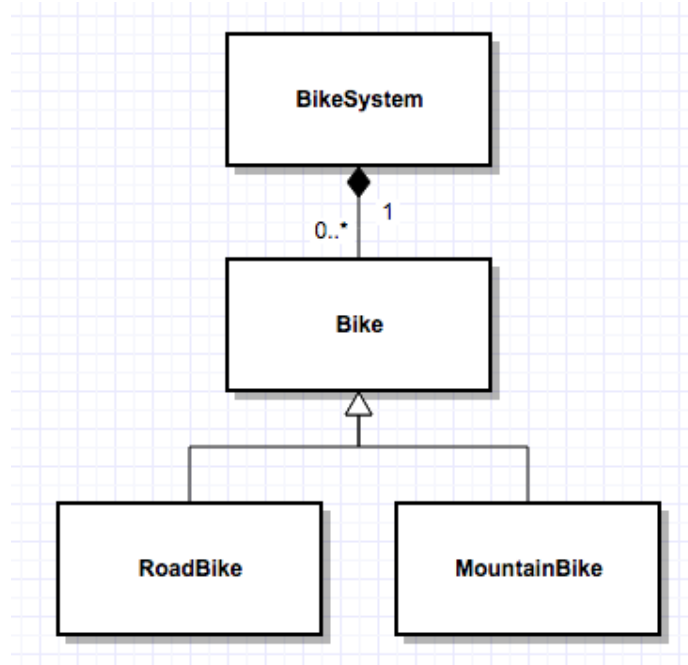
Jess - Business - 2018
WAM: 75
0 x F
0 x P
1 x C
3 x D
0 x HD
```

### Marking Criteria:

|   |  |
|---|--|
| Presentation & correctness<br>( 2 marks ) | 1 mark* Style and cleanliness, indentation, camelCase<br>1 mark* Programming correctness, variable naming, placement of methods<br><i>* No partial marks awarded</i>                         |
| Test cases<br>( 3 marks )                 | 1 mark per test case – 3 test cases total  |
| Plagiarism                                | 90% penalty for the entire assignment for substantial plagiarism<br><i>More information on the appeal process and what is considered to be plagiarism is available on the course outline</i> |

## Part 2 – Bike Repair Shop (10%)

You've been asked by a bike repair shop to implement a basic database system. The relationship between objects is dictated below:



Each class has the following attributes:

- Bike (String owner, String model, int wheelDiameter, int gears, int kmsRidden)
- MountainBike (int suspensionDepth)
- RoadBike (int reflectorsFitted)

Your Bike Shop is a bit quirky, the service interval is calculated in the following way:

- MountainBike = wheel diameter \* gears – suspension depth
- RoadBike = kms ridden \* reflectors fitted
- Bike = length of owner's name \* gears

### Programming tasks (8%):

- Using your knowledge of OOP, Inheritance and Polymorphism design and implement the classes above
- You should submit 5 Java files: BikeSystem.java, TestBikeSystem.java, Bike.java, MountainBike.java and RoadBike.java
- You will need to implement a BikeSystem class which is in charge of maintaining the objects, the BikeSystem constructor should take an int which specifies the number of bikes the BikeSystem can hold, and should provide the following publicly accessible methods:

```
public boolean insertBike(String bikeString)
```

The boolean will indicate whether the Bike was successfully created. Returns false if the bike already exists or the BikeSystem is full. You will need to parse these Strings and create objects from them

The bikeStrings will come in the following formats:

Bike|<owner>|<model>|<wheelDiameter>|<gears>|<kmsRidden>

MTB|<owner>|<model>|<wheelDiameter>|<gears>|<kmsRidden>|<suspensionDepth>

RB|<owner>|<model>|<wheelDiameter>|<gears>|<kmsRidden>|<reflectorsFitted>

```
public boolean bikeExists(String owner, String model)
```

This boolean will indicate whether the bike exists in the system or not

```
public void printSystem()
```

This should call the toString() method of each object and print it to the screen in the same format as the bikeString inputs

```
public void closeShop()
```

This should empty the bikeShop and restore it to an empty state

```
public void printServiceIntervals()
```

This should print the list of bikes, and service intervals in the format:

owner, model, serviceInterval

### Design Explanation (2%):

- You must submit a design.txt file containing your answer to the following questions (200 words maximum)
- Explain what is happening when you call toString() for each of the objects when calling printBikeSystem()?
- When you determine the service interval in printServiceIntervals() what aspect of OOP are you utilising at runtime?

## Marking Criteria:

|   |  |
|---|--|
| Presentation & correctness<br>( 2 marks ) | 1 mark* Style and cleanliness, indentation, camelCase<br>1 mark* Programming correctness, variable naming, placement of functions<br><i>* No partial marks awarded</i>   |
| Functionality Tests<br>( 6 marks )        | 1 mark insertBike functions correctly<br>1 mark bikeExists functions correctly<br>1 mark printSystem functions correctly<br>1 mark closeShop functions correctly<br>1 mark printServiceInterval functions correctly<br>1 mark mystery test case<br><i>There may be more than one test case required to determine each mark</i> |
| Design Explanation<br>( 2 marks )         | 1 mark correctly identify the relationship between objects and toString()<br>1 mark correctly identify the aspect of OOP that occurs at runtime  |
| Plagiarism                                | 90% penalty for the entire assignment for substantial plagiarism<br><i>More information on the appeal process and what is considered to be plagiarism is available on the course outline</i>   |