ASSIGNMENT BRIEF

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module Code | COM4005 | Module Leader | | Dr Andrew Guest | |
| Module Title | Programming 02 | | | | |
| Level | 4 | Credit Value of Module | | 20 | |
| Assessment Task | Portfolio – OOP Essay, Application Design, Application, Class Exercises | | | | |
| Word Count | Essay (1000 words) | | | | |
| Assessment No | 1 | of | 1 | Weighting | 100% |
| Type of Submission | Portfolio | | | | |
| Method of Submission | Electronic through moodle & gitlab | | | | |
| Publication Date | 7/2/22 | | | | |
| Due Date | 16/05/22 12:00pm (Noon) | | | | |
| Expected Feedback Date | 06/06/22 | | | | |
| Resit Date | 15 August 2022 12:00pm Noon | | | | |
| Format of Feedback | Through moodle | | | | |
| Anonymous marking | Not anonymous. Documentation is closely linked to code and code cannot be anonymous.  Code of Assessment 29.5.2  c) Subject-specific skills exercises or product creation e.g. artwork, product design, video production, performances.  f) Collaboratively produced or peer-marked assessments e.g. group work. | | | | |
| Learning Outcomes | | | | | |
| 1. Demonstrate understanding of essential facts, concepts, principles and theories of Object Oriented Programming; 2. analyse, model, design, test, and evaluate object orientated programmes; 3. apply appropriate theory, practices and tools for the specification, design, development and evaluation of computing systems including programming in a high level language; 4. Solve problems and represent ideas at varying levels of abstraction. | | | | | |
| Assignment Description | | | | | |
| **Overview**    *Please ensure you read all of this document. Pages 4-6 contain the marking rubric that shows how the module will be marked.*  The assessment for this module consists of four parts. Part one is an essay on Object Oriented concepts, Part two is a design exercise, to come up with a design for a code challenge and to explain it. Part three is to write code that implements that design. Part Four is a collection of five class exercises given out during the lab sessions. Each of these parts is described in the sections below, each is worth one quarter of the overall mark for this module. | | | | | |
| **1 – Essay – Principles of Object Oriented Programming (1000 words)**  Write an essay describing the principles of Object Oriented programming. You should describe what classes and objects are. You should also explain the fundamental principles of OOP (polymorphism, abstraction, encapsulation, inheritance). You should also discuss the advantages and disadvantages of OOP.  You should reference any external sources you use, and quotations should be kept short and be used to illustrate something you have described in your own words. Diagrams can be very helpful and should also be referenced if you have not created them yourself. | | | | | |
| **2 – Design A Zoo**  Imagine a virtual York St John zoo. We want to have a variety of animals from around the world and show them off. We want to have a mixture of mammals, birds, fish and reptiles. We want to have animals from deserts, arctic, jungle, forests, etc. We need to be able to organise them by the type of animal they are. We want to be able to ask questions like “what does each animal eat?”, “what animals do we have from a specific environment”  Create a design for a zoo application. You should design the classes required to implement a system for tracking animals in a zoo. Use inheritance to distinguish between the various types of animals. You will need to store information on each animal such as food type, environment, description, etc.  You should use good Object Oriented practices. Think about the hierarchy of classes. Look for “is a” relationships to determine parent/child classes. Think about which class attributes and methods should be in to reduce repeated code and improve maintainability.  Your design should include a class diagram showing the classes you would create, the relationships between them. Showing the member variables and methods on this diagram would be good. | | | | | |
| **3 – Zoo Application**  Take your design from part 2 and implement it in Java.  Your application should implement the following features   1. Use an ArrayList to store a list of all animals in the zoo 2. Display the details of an individual animal 3. Display the details of each animal in an ArrayList 4. Generate an ArrayList of all animals of a particular type 5. Output “Food required” for all animals in an ArrayList (might be all animals, by be a list from 4) 6. Output all animals from a specific environment. | | | | | |
| **4 – Class Exercises**  During the lab sessions some of the exercise sheet have an exercise marked **Assessment Exercise.**   1. Complex Numbers 2. Shapes 3. Bank Accounts   *Please note that the exercise below is no longer part of the assessment.*   1. *Little Lucy’s Fabledrealm* | | | | | |

|  |
| --- |
| **Marking**  This assessment will be marked by rubric. The rubric is included below.  The marks break down as   * OOP Essay 25% * Zoo Design 25% * Zoo Application 25% * Assessment Exercises 25% (Three exercises) |
| |  |  |  | | --- | --- | --- | | **1. OOP Essay (25%)** |  |  | | **Component** | **Overall** | **Required Content** | | Introduction | 10 | An introduction describing the purpose of the essay. | | What & Why OOP? | 20 | What is OOP and why is it a good idea | | Classes & Objects | 20 | What are classes and objects? How do they relate to each other? What are member variables and methods? | | OOP Principles | 30 | What is encapsulation, abstraction, inheritance and polymorphism? | | Summary | 10 | A brief conclusion/summary | | Quality | 10 | Correct spelling, punctuation, grammar. Sensible Structure | | **2. Zoo Design (25%)** |  |  | | **Component** | **Overall** | **Required Content** | | High level design | 30 | A high level description of the application you are designing | | Class diagram | 50 | A class diagram showing the classes that will be part of the application. This should show parent/child relationships. Marks will not be lost if you hand draw the diagram, take a photo of it with your phone and use that image. | | Feature list and other design notes | 20 | A list of features the application will implement (or attempt to implement) with notes on how they would be implemented. Any other design notes | | **3. Zoo Application (25%)** |  |  | | **Component** | **Overall** | **Required Content** | | Code runs without errors | 10 | Code, submitted through gitlab, runs without errors | | Animal ArrayList | 10 | Application stores all the animals in single ArrayList | | Animal Classes | 20 | Application has an appropriate set of classes for storing the animals | | Display Details | 10 | Display the full details of a single animal | | Display List | 10 | Display the details of a list of animals | | Get Type List | 10 | Generate a list of all animals of a specific type – mammal, fish, bird or reptile | | Display Food | 10 | Display the food required for a list of animals | | Get Animals From Environment | 20 | Display a list of all animals that come from a specific environment | | **4. Assessment Exercises (25%)** |  |  | | **Component** | **Overall** | **Required Content** | | Exercise 1 | 5 | public double magnitude() implemented | |  | 5 | public double argument() implemented | |  | 5 | public double add() implemented | |  | 10 | public double toString() implemented | | Exercise 2 |  |  | | Square | 2 | constructors | |  | 2 | setSize(), getSize() | |  | 2 | area(), circumference() | |  | 2 | draw() | |  | 2 | Pass all Square unit tests | | Rectangle | 3 | constructors | |  | 3 | setWidth(), getWidth(), setHeight(), getHeight() | |  | 3 | area(), circumference() | |  | 3 | draw() | |  | 3 | Pass all Rectangle unit tests | | Exercise 3 | 10 | Bank Account | |  | 5 | Saving Account | |  | 10 | Overdraft Changes | |
| Assessment Regulations |
| * Your attention is drawn to the University policy on academic misconduct ([Academic Misconduct Policy](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/23.Academic_Misconduct_Policy_2021-22.pdf)). Penalties will be applied where a student is found guilty of academic misconduct, including termination of programme. * You are required to keep to the word/time limit set for an assessment and to note that you may be subject to penalty if you exceed that limit ([Agreed Penalties Policy](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/30.Agreed_Penalties_Policy_2021-22.pdf)). You are required to provide an accurate word count on the cover sheet for each piece of work you submit. * For a first assessment attempt a penalty may be applied for late or non-submission of work by the published deadline or an approved extended deadline ([Agreed Penalties Policy](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/30.Agreed_Penalties_Policy_2021-22.pdf)). * Where a re-assessment opportunity exists, late or non-submission of work receives a mark of zero and is not eligible for a capped mark ([Agreed Penalties Policy](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/30.Agreed_Penalties_Policy_2021-22.pdf) and [Reassessment](https://www.yorksj.ac.uk/students/exams-and-assessment/reassessment/)) * An extension to the published deadline may be granted to an individual student if they meet the eligibility criteria of the [Exceptional Circumstances policy](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/16.Exceptional_Circumstances_Policy_2021-22.pdf). |
| Note |
| [Feedback Policy:](https://www.yorksj.ac.uk/media/content-assets/registry/policies/code-of-practice-for-assessment/31.Feedback_Policy_2021-22.pdf) Marks are to be returned to students with the caveat that all marks are provisional until final approval by the School Assessment Board (SAB). Confirmed marks will be made available via e:Vision after the SAB meeting. |