**Summative Assessment - Module 5**

*Hello and welcome to the assessment. Here you’ll prove to the world just how much you know and understand about what you’ve just learnt in the learner guides. This is an important part of your time at Umuzi because once this is done, you’ll be certified! So please, take this time to learn everything you can! Take a look at some pointers below with regard to answering the questions…*

* *Be specific*
* *Write professionally - no shorthand!*
* *Your answers must be original and come from your brain and your brain only.*
* *No copy/paste tricks! Our markers have seen it all and will know if you’re taking shortcuts.*
* *Remember, sloppy or poor work will be sent back to you to do again, so do it properly the first time and you’ll be done in no time.*
* *Ask for help at any time. Ask your friends, a manager, anybody!!*
* *Don’t skip any questions! You must do them all!*
* *You’ll see two boxes after each question - one for your answer and one for the marker’s comments. DO NOT delete the marker’s comments if you are required to resubmit your work after the first attempt. Should you have to do it again you will see a new box* ***under*** *the marker’s comments, so fill that one out in* ***PURPLE****. Remember!! It’s not the end of the world if you have to resubmit. You’re here to learn, so don’t beat yourself up if you don’t get it right on the first go. Obviously, try your best to get it right on the first attempt, but if not, you have another chance to do it properly!*

*Ok, and that’s that! Time to get to it! Good luck, have fun and enjoy! :)*

**Enter your name and surname below**

|  |
| --- |
| **Luvo Spofana** |

**1. Describe the concept of abstraction in object-oriented programming (14)**

**Your answer below**

|  |
| --- |
| Abstraction is the process of hiding all but the relevant information about a thing to make things less complex and more efficient for the user. In Object-Oriented programming, Through the process of abstraction, a programmer hides all but the relevant data about an object in order to increase efficiency and reduce complexity. Abstraction lets you focus on what the thing does instead of how it does it. |

**Marker’s Comments**

|  |
| --- |
|  |

**2. Provide an example of one design pattern in software development and describe what it is used for (14)**

**Your answer below**

|  |
| --- |
| Bridge: This design pattern is structural, and it allows you to split a set of closely related classes or a large class into two separate hierarchies: implementation and abstraction – which can be developed independently of each other. |

**Marker’s Comments**

|  |
| --- |
|  |

**3. Describe the difference between creating a fork and creating a branch when using GIT (15)**

**Your answer below**

|  |
| --- |
| Creating a fork creates a full copy of your repository, while creating a branch only adds a branch to the existing tree. A fork is independent from the original repository and does not get affected when changes are made to the original repository. When one forks a repository, they get all of that repository’s branches. |

**Marker’s Comments**

|  |
| --- |
|  |

**4. What are the three steps of any test-driven development process? (16)**

**Your answer below**

|  |
| --- |
| **Red** – The first step is a phase in one in which the developer writes the tests without having written the code with the program’s functions and methods and executes the tests to fail. When the tests fail, the unit test framework the developer is using will give a red flag.  **Green** – In this step the developer then writes the code that will pass the 1st test. With the Test-driven development approach, the developer writes code just enough to pass 1 test at a time, and then goes on to add another test that will fail and raise a red flag for which the developer will write more code that will then pass the 2nd tests and keeps going.  **Refactor** – The 3rd step comes into place when all the tests and the code to pass the tests has been written, the goal of this step is to adopt and incorporate good design patterns, consider the code’s maintainability and quality so it can be adjusted to be up to the expected standard. |

**Marker’s Comments**

|  |
| --- |
|  |

**5. What information should you include in the high-level documentation for a software file you are working on? (13)**

**Your answer below**

|  |
| --- |
| * The title of the design document, the people who will be working on the project, and the reviewers of the document. * An overview of the project, which consists of a high-level summary that every developer or engineer in the company can understand. * The context – A description of the problem identified that the software project is looking to address, the reasons outlining why this project is necessary, the information stakeholders need to know to assess this project, and how it fits into the team’s overall goals. * The solution and the objectives of the proposed solution. This describes the user-driven impact of the project, and metrics on how to measure success. * The project milestones, which outline the start date, and every checkpoint of the development lifecycle of the project, so that the stakeholders can have a clear idea of when different parts of the project will be done. * The description of the current existing solution. This describes how the users are currently dealing with this problem, the system they use and how they interact with that system. * The description of the proposed software solution, which entails how users will deal with the problem, and shows how they will interact with the software they will use. * Cross-team impact: This measures the cost of the project and details the impact the project will have on the team. * Detailed scoping and timeline – This section breaks down how and when each part of the project is planned to be executed, and the resources to be used to execute the different parts of the project. |

**Marker’s Comments**

|  |
| --- |
|  |

**6. Describe how the merge sort algorithm works (19)**

**Your answer below**

|  |
| --- |
| This algorithm works in two ways:   1. It starts at the top and works its way down, splitting the array in half, making a recursive call, and merging the results until it reaches the bottom of the array tree. 2. The second way this algorithm works by starting with a ‘single-element’ array and then merges two nearby items while also sorting them. The combined sorted arrays are merged and sorted again until only one single unit of the sorted array remains. |

**Marker’s Comments**

|  |
| --- |
|  |