**Topic 7: Object-oriented Programming**

**Formative Assessment 07**

**Object-oriented Programming (115378)**

*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* ***BLUE****. 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 inheritance in Object-Oriented Programming and provide an example of when it should be used [5 Marks] (5 SO:1 AC:1-2)**

**Your answer below**

|  |
| --- |
| Inheritance is deriving a class from another class so that the methods and attributes of one class can be part of the definition of the other class. The first class that another class is derived from is referred to as the parent/base class, and the derived child class is known as the sub-class.  It can be used on objects that share some common attributes and behavior but may have their own attributes that make them unique to that particular type, e.g., sharks and goldfish are both kinds of sea creatures, but have unique attributes and behaviors. |

**Marker’s Comments**

|  |
| --- |
|  |

**2. Describe polymorphism in Object-Oriented Programming and provide an example of when it should be used [5 Marks] (5 SO:2 AC:1-2)**

**Your answer below**

|  |
| --- |
| This is a core concept that provides the ability for the same method to be called on different JavaScript objects. With polymorphism, different classes react to and implement the same method call in different ways specific to the class that called the method.  For example, both cat and parrot types can be able to respond to a ‘makeSound’ message, while the behavior is the same, the means of achieving/executing it are different and specific to each type. |

**Marker’s Comments**

|  |
| --- |
|  |

**3. What is a class? Include in your description how it incorporates abstraction and encapsulation in its design. [6 Marks] (6 SO:3 AC:1-4)**

**Your answer below**

|  |
| --- |
| Classes are special functions which are a template or blueprint for creating objects. They provide initial values for state and implementations for behavior. Classes are created with the same syntax as object methods but are not objects.  A class incorporates abstraction by using abstract classes to represent essential information and methods that multiple sub-classes inherit from, and when a sub-class wants to use an abstract class, it first inherits it from the base class and provides implementations to the abstract methods in the abstract class.  A class incorporate encapsulation by bundling data/attributes, and methods that act on that data, in a way that restricts access to that data from outside bundles, having a private implementation but giving the abstract data type that the attributes/data are bundled into a public interface. |

**Marker’s Comments**

|  |
| --- |
|  |

**4. What two features make something an “abstract class”? (4 SO:4 AC:1-3)**

**Your answer below**

|  |
| --- |
| Abstract classes cannot be made into objects, basically cannot be instantiated.  Abstract classes have special methods that are called ‘abstract methods’. |

**Marker’s Comments**

|  |
| --- |
|  |

**5. What is a design pattern? Describe it in general and give a specific example [6 Marks] (6 SO:5 AC:1-3)**

**Your answer below**

|  |
| --- |
| Design patterns in software design are solutions that are reusable to commonly occurring problems, helping developers build upon the combined experience of many developers that came before us, and ensuring that our code is maintainable and structured in an optimized way that meets the needs of the problem at hand. Design patterns are not exact solutions, they provide solution schemes. Design patterns make things significantly easy for developers by providing a common vocabulary to describe solutions.  A Constructor Pattern is an example of design patterns, this special method is used to initialize a newly created object once memory has been allocated for it. |

**Marker’s Comments**

|  |
| --- |
|  |