# **Software Development 2011 Unit 3 – Outcome 2 (60 marks)**

**Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Description**: **Represent a software design and apply a range of functions and techniques using a programming language to develop a prototype solution to meet a specific need.**

|  |  |  |  |
| --- | --- | --- | --- |
| Criteria for assessment | Allocated Marks | | **Comments** |
| **Design of the software module** |  |  |  |
| **Write a set of criteria for evaluating the efficiency and effectiveness of the solution.**  **effectiveness**   * buttons do what they claim to * accurate spelling and punctuation * **Be easy to use** * Little training required   easy –to-understand help   * **Intuitive** design – it should be obvious how to do something even for a beginner. * **Easy to read** * simple language * Avoid words where possible and use icons * good colour contrast * Avoid background images * **Be consistent, predictable** * What an icon means must not change from page to page * The same theme (colours, typefaces, text formatting) should be used throughout a document * Don’t change things for the sake of change * **follows standard conventions** * Conventions are used to reduce the amount of learning users need to invest in a product. They expect each product to follow standard behaviour * e.g. The leftmost menu is ‘File’ and the rightmost is ‘Help’ * The software version is always found under Help > About |  |  | / 3 |
| **`Efficiency**  Easy-to-find  Most commonly-used tasks should be easiest to find, not hidden deep-down in menus  Give users alternative ways of doing things (e.g. Ctrl+C shortcut, a “Copy” in the toolbar, “Copy” menu item, a “Copy” action in the right-click context menu.  Conventions are used to reduce the amount of learning users need to invest in a product. They expect each product to follow standard behaviour  To evaluate **time** efficiency, you could look at the time taken to:   * start the system * enter data * process data * produce output * communicate information |  |  | /2 |
| **Data types and structures are accurately described using appropriate design methods: Data Dictionary and Data Structure Diagram**  Data element name (e.g. a field's name, like "Surname") \* Short description \* Data type (characters, numeric, etc.), size \* Null value allowed: Null or non-existing data value may or may not be allowed for an element. \* Default value: Data element may have a default value. Default value may be a variable, like current date and time of the day \* Allowed values and validation \* Whether the field is a key field. \* Rules used in calculations producing the values are usually written here. |  | / 10 |  |
| **Solution requirements are accurately and clearly represented in a detailed algorithm using pseudo-code.**  BEGIN  Display Form Admin Screen  If Queen Bee price is changed in Form Admin then  Queen Bee price in Form Customer = Queen Bee price in Form Admin  End if  If Worker Bee price is changed in Form Admin then  Worker Bee price in Form Customer = Worker Bee price in Form Admin  End if  If Discount is selected then  Apply Discount to Discount in Form Customer  Else  No Discount applied  End if  If Half-Price Postage is selected then  Apply Half-Price to Half-Price Postage in Form Customer  Else  No Half-Price Postage applied  End if  END  BEGIN  Display Form Customer Screen  Get Firstname  Get Surname  Get Address  Get State  Get Postcode  If Certain Postcode then  Postal Cost = 0  Else if Certain Postcode then  Postal Cost = Variable rates  End if  Get Queen Bee Quantity  Queen Bee Subtotal = price of Queen Bees times quantity  Worker Bee Subtotal = price of Worker Bees times quantity  Surcharge = Subtotal plus surcharge percentage of Queen Bees  Total = Queen Bee Subtotal plus Worker Bee Subtotal plus Surcharge plus Postal Cost  Grand Total = Total minus Discount  Display Grand Total  END |  | / 5 |  |
| **Development of the software module** |  |  |  |
| **Manipulation -** The prototype solution is efficiently coded with all data types and data structures being appropriately used and integrated.  2 d array |  | / 20 |  |
| **Documentation** - Internal documentation is complete, contains relevant program comments, and is clearly stated and well formatted. |  | / 5 |  |
| The solution meets specifications as stated in the SRS |  | / 10 |  |
| **Testing/Validation** - An appropriate range of test data is expressed in a testing table, with both expected and actual output stated. All tests are successfully applied to enable the performance of the prototype solution to be verified.   * carrying out functions to see if they exist and work as expected * timing how long an operation takes * counting errors recorded in an error log * visually inspecting output for quality * counting and assessing incidents of failure or downtime * stress-testing the system or abusing it in ways that would be expected in real-life operation * using the system to its stated capacity * using the system in conjunction with another system to see if data and hardware are compatible * attempting to break into the system or get access to secured data * attempt to configure the system to different users, or add modules, or change the way it works to suit new need |  | / 5 |  |
| **Total :** |  | **/ 60** |  |