# HEX BARON – Theory Questions (MS) (Total Marks: 50)

| **Question** | | **Suggested Solution** | **Marks** | **Guidance** |
| --- | --- | --- | --- | --- |
| **1** | a | By using player1.getName() and player2.getName() | 1 mark | * A. any sensible mention of getName() |
|  | b | To provide an interface to private or protected attributes | 1 mark | * A. either private or protected * A. access or any other similar word for interface |
| **2** |  | The variable player1Turn is toggled using not player1Turn;  Player Two always gets their turn if Player One ends the game as the game can only end when player1Turn is true | 2 marks | * 1 mark for each point |
| **3** |  | hasMethod is used to check whether the command that the player typed in  is available in the class for the piece in the tile | 2 marks | * A. to retrieve the method in the Piece class if it exists |
| **4** | a | It returns the maximum of three possible numbers:  The absolute difference between the x coordinate of the tile t and itself;  The absolute difference between the y coordinate of the tile t and itself;  The absolute difference between the z coordinate of the tile t and itself | 3 marks | * 1 mark for the first point * 1 mark for any of the other three points or 2 marks for all of the last three points * MAX 2 if no mention of absolute (or equivalent) |
|  | b | First, max(abs(2-4),abs(-3-(-4))) which is max(2,1) = 2  then max(2,abs(0-0)) which is max(2,0) = 0 | 2 marks | * 1 mark for each max calculation if correct * MAX 1 mark if just max(2,1,0) or similar short answer |
| **5** | a | Inheritance is where you create a new class that gains all the attributes and methods of its parent | 1 mark | * A. similar words or meanings, e.g. behaviours |
|  | b | Because the implementation for this is already in the Piece class | 1 mark | * A. answers that refer to Piece as a concrete class or talk about avoiding it being an abstract class |
| **6** | a | Overriding means having a method of the **same name**;  But with a **different implementation** (in the child class) | 2 marks |  |
|  | b | Because the cost of a move for a Baron is always 1;  Whereas the cost for a Serf can be 2 (when starting or ending in a peat bog) | 2 marks |  |
| **7** | a | O(n2) | 1 mark | * A. n2 or n squared |
|  | b | Space complexity | 1 mark | * A. memory complexity/efficiency |

| **Question** | | **Suggested Solution** | **Marks** | **Guidance** |
| --- | --- | --- | --- | --- |
| **8** |  | Because it is a standard format;  That can be used/read easily on all platforms | 1 mark | * MAX 1 mark, accept any reasonable point |
| **9** |  |  | 4 marks | * 1 mark for both missing class names (Piece and PBDSPiece) * 1 mark for the composition from HexGrid to Tile * 1 mark for the aggregation from Tile to Piece * 1 mark for the two inheritance arrows from BaronPiece and LESSPiece to Piece * R. incorrect arrowheads or arrowheads on the wrong end |
| **10** | a | The upgrade was unsuccessful | 1 mark | * A. any similar statement |
|  | b | The upgrade was successful AND 5 lumber should be deducted from the player | 1 mark | * Only accept answers that make both points |
| **11** |  | In the dig method, a random number is generated;  If this is less than 0.9 then it returns 1, otherwise it returns 5;  Once the return value is received in the executeCommandInTile method, it checks to see whether it is greater than 2;  And if it is, then the terrain is set to a field | 4 marks | * A. blank or space for field |
| **12** | a | It uses the getConnectionsNeededToDestroy method on the Piece class which is the parent; and could be (but isn’t) overridden in and of the child classes as they all must have the **same method due to inheritance** | 2 marks |  |
|  | b | Private attributes are not inherited | 1 mark |  |

| **Question** | | **Suggested Solution** | **Marks** | **Guidance** |
| --- | --- | --- | --- | --- |
| **13** | a | upgrade (pbds|less) (0|[1-9][0-9]\*) | 3 marks | * 1 mark for upgrade and (pbds|less) * 1 mark for integers including ones starting with 0, such as 01, which are not strictly valid (e.g. [0-9]\*) or 2 marks for only strictly positive integers including 0 |
|  | b | Upgrade (pbds|less) (0|[1-9][0-9]?) | 1 mark | * 1 mark for any expression that only allows integers from 0‒99 * DPT problem with integers starting with 0, e.g. 01 from part a |
|  | c | One that can be represented by a FSM (with no outputs) | 1 mark | * A. one that can be represented by a regular expression |
| **14** | a | gridAsString  listPositionOfTile  returnObjects | 1 mark | * Any one of the variables listed for 1 mark MAX |
|  | b | Their scope is limited to the current subroutine;  So that means that you can test the subroutine in isolation;  And, therefore, allow people to use the subroutine through its interface only so they do not need to concern themselves with the implementation | 3 marks | * A. any valid point about scope being the subroutine or a related advantage (e.g. no need to worry that there might be changes elsewhere) * A. equivalent ideas * A. method or function for subroutine |
|  | c | A private attribute is only available within the class where it is defined;  Whereas a protected attribute can also be accessed by sub-classes | 2 marks | * A. children for sub-classes * A. equivalent points presented from a different perspective |
| **15** |  | GROUP 1 Marks:  It checks to see whether there is at least 1 piece in supply;  and at least 3 lumber;  and that the tile index given is valid;  and that the tile is empty  GROUP 2 Marks:  It iterates through all of the neighbouring tiles;  to see if one of them contains the Baron for the **current player**;  it returns -1 if **any** of the conditions are not met;  Or 3 if **all** the conditions are met | 6 marks | MAX 3 from Group 1 Marks  MAX 4 from Group 2 Marks  MAX 6 marks total  DPT for a condition not being met if they didn’t state the condition the first time around when awarding the mark for returning -1 |