CS1005 Logic and Computation

Assessment/Coursework for 2015/16

|  |  |
| --- | --- |
| **Assessment Title** | PROLOG Programming Assessment |
| **Module Leader** | Allan Tucker |
| **Distribution Date** | Late October 2015 |
| **Submission Deadline** | 17th January 2015 |
| **Feedback by** | 12th February 2015 |
| **Contribution to overall module assessment** | 25% |
| **Indicative student time working on assessment** | 25 Hours |
| **Word or Page Limit (if applicable)** | N/A |
| **Assessment Type (individual or group)** | Individual |

**Main Objectives of the assessment**

This piece of coursework will involve designing and implementing rules in PROLOG to answer 8 specific questions about data. The data represents the relationship between Proteins, Amino Acids and DNA.

**Description of the Assessment**

**Part 1 - Coursework (25 % of overall assessment)**

This PROLOG programming project involves the analysis of *protein* sequences, *DNA* sequences and *amino acids*.

DNA (Deoxyribonucleic acid) is a molecule that stores biological information and genetic instructions and is present in the nucleus of the cells of all living tissue. DNA sequences consist of a long string of the letters:

***c,a,g or t***

Proteins are created by cells (genes) that then carry out some function within an organism. Protein sequences consist of a sequence of 20 possible amino acids - represented as letters. For example the following protein sequence (a sequence of amino acids) is one of the mitochondrial genes of a Neanderthal man.

***MPQLNTTVWPTMIIPMLLTLFLITQLKMLNTNYHLPPSPKPMKMKNYNKPWEPKWTKICSLHSLPPQS***

Each amino acid in a protein has a corresponding DNA sequence: each amino acid translates to three DNA molecules. Most amino acids can have more than one sequence. For example the amino acid Tryptophan (Trp for short and is code ‘W’ - as found in the Neanderthal example above) has the following DNA encoding: tgg, whilst Cysteine (Cys and ‘C’ - also found in the example) has two encodings tgc and tgt (see the figure below for a graphical representation of the relationship between proteins, amino acids and DNA).



For this coursework, three types of facts are provided that form a database describing properties of amino acids (included at the bottom of this document). The first gives the name of the amino acid, the second describes the amino acid to DNA encoding and the third describes features and attributes of each amino acid.

***aminoname*(*<Letter>,<Long Name>,<Short Name>*)*.***

Where <Letter> is the single letter code for an amino acid, <Long Name> is the long name of the amino acid and <Short Name> is an abbreviation for the amino acid.

***aminodna*(*<Letter>,<DNA Sequence>*)*.***

where <Letter> is the single letter code for the amino acid and <DNA Sequence> is a PROLOG list that contains the three DNA molecules (letters) that the amino acid encodes to. If there exists more than one mapping then there will be multiple facts, e.g. one fact for Tryptophan and two facts for Cysteine.

***aminoclass*(*<Short Name>,<Class Type>,<Class Value>*)*.***

where <Short Name> is the abbreviation for the amino acid, <Class Type> is a description of a classification/feature/attribute of an amino acid and <Class Value> is the value for the corresponding classification.

For example ***aminoclass*(*'Trp','Hydropathy','hydrophobic'*)*.*** means that the amino acid Tryptophan has a Hydropathy rating/value of Hydrophobic.

This PROLOG assessment is split into a number of questions. The weighting for each question is given in parentheses with each question description. All questions should be attempted, all rules and all variables should be given sensible names and all rules should be commented. Please see the marking scheme in Appendix A.

Note that you MUST NOT edit the database file (AminoAcids.pl from Appendix B), write your answers in a single additional file called <Student ID>.pl where <Student ID> is your student ID number. Note that some of the questions may be answered by reusing the rules that you have written for other answers. If you are unsure what any of the questions mean it is your responsibility to ask about them during the specified laboratory times.

|  |  |  |
| --- | --- | --- |
| **Question** | **Task** | **Weight %** |
| Q1 | Write a rule to determine if a particular symbol corresponds to an amino acid or not. For example, ***isaminoacid(a)*** would return true and ***isaminoacid(z)*** would return false. | 8 |
| Q2 | Write a rule that when given the short name of an amino acid it displays the long name. For example ***aminolong('Tryptophan')*** would display ‘Trp’ | 12 |
| Q3 | Write a rule that displays the long name of all of the small neutral amino acids. | 12 |
| Q4 | Display all of the aromatic and not positively charged amino acids. | 12 |
| Q5 | For three DNA molecules, display the amino acid, for example ***findamino(t,g,g)*** would display w for Tryptophan. | 12 |
| Q6 | For a list of amino acids display each amino acid’s property in a report format, higher scoring solutions will have an attractively "formatted" report. | 16 |
| Q7  Q8 | For a given protein sequence in a list, i.e. a list of amino acids, display all possible DNA sequences, i.e. taking into account all possible permutations of sequences for all of the amino acids that have more than one set of DNA sequences.  Along the same lines as question 7, instead of displaying the sequence produce a report for each possible sequence that counts how many times each of the four DNA molecules occur. | 16  12 |

**Part 2 – Viva (PASS/FAIL)**

Once the code has been submitted, the student must attend the viva event where they will talk a member of staff through their code and demonstrate how it works. This will involve a through test of the code showing where the code works and where it fails. They will also be expected to answer some questions. Passing the viva will require you to:

1. demonstrate your code with examples,

2. test your code and

3. answer questions about your code.

**Learning Outcomes and Marking Criteria**

|  |  |
| --- | --- |
| **Learning outcomes for the assessment** | **Assessment and marking criteria** |
| Demonstrate an applied understanding of logic programming | See Appendix A. |

**Format of the Assessment**

The assessment content that you need to upload to Blackboard must consist of your commented PROLOG source code only.

**Submission Instructions**

Coursework must be submitted electronically via the University’s [Blackboard Learn](https://blackboard.brunel.ac.uk) system. The required file format for this report is a prolog .pl file. Your student ID number must be used as the file name (e.g. 0123456.pl).

**Avoiding Plagiarism**

Please ensure that you understand the meaning of plagiarism and the seriousness of the offence. Information on plagiarism can be found on the [College’s Student Handbook](https://intra.brunel.ac.uk/cedps/Pages/handbook.aspx).

**Late Coursework**

Please refer to the [College’s Student Handbook](https://intra.brunel.ac.uk/cedps/Pages/handbook.aspx) for information on submitting late, penalties applied and procedures in the case of mitigating circumstances.

**Appendix A – Grading Scheme**

|  |  |
| --- | --- |
| Criteria | grade |
| 1. The code is functionally correct for all tasks 2. All code is written and documented in a logical way with good structure 3. There are comprehensive comments to help understanding throughout 4. The student has demonstrated understanding of their code during the viva 5. The student has made excellent use of test cases during the viva 6. The student has answered all of the questions fully during the viva | A |
| 1. The code is functionally correct for nearly all tasks 2. Most code is written and documented in a logical way with good structure 3. There are sufficient comments to help understanding in all of the code 4. The student has demonstrated understanding of their code during the viva 5. The student has made very good use of test cases during the viva 6. The student has answered all of the questions fully during the viva | B |
| 1. The code is functionally correct for most tasks 2. The code is mostly written and documented in a readable way 3. There are some comments to help understanding 4. The student has demonstrated understanding of their code during the viva 5. The student has made good use of test cases during the viva 6. The student has answered all of the questions fully during the viva | C |
| 1. The code is functionally correct for some tasks 2. Some code is written and documented, but not always readable 3. There are few comments to help understanding 4. The student has demonstrated understanding of their code during the viva 5. The student has made good use of test cases during the viva 6. The student has answered some of the questions fully during the viva | D |
| 1. The code is partially correct for some tasks but not all questions 1-4 2. Some code is written and documented, but not always readable 3. There are one or two comments to help understanding 4. The student has demonstrated little understanding of their code during the viva 5. The student has not made good use of test cases during the viva 6. The student has answered one or two of the questions fully during the viva | E |
| 1. The code is not functionally correct for any tasks 2. The code written is unclear 3. There are no comments to help understanding 4. The student has not demonstrated understanding of their code during the viva 5. The student has not made good use of test cases during the viva 6. The student has NOT answered any of the questions fully during the viva | F |

The + / - grading criteria will be used to differentiate between intermediate grades

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Question | 1  (8%) | 2  (12%) | 3  (12%) | 4  (12%) | 5  (12%) | 6  (16%) | 7  (16%) | 8  (12%) |
| Functionality |  |  |  |  |  |  |  |  |
| Structure |  |  |  |  |  |  |  |  |
| Comments |  |  |  |  |  |  |  |  |
| Questions |  |  |  |  |  |  |  |  |
| Marks |  |  |  |  |  |  |  |  |

**Appendix B – PROLOG Amino Acids Database**

The AminoAcids.pl database file can be found in the embedded file as follows (double click to open it):

