Department of Computer Science

CS2004 Algorithms and their Applications
Description of Worksheet Based Assessment
Task 3 TSP: Assessment/Coursework for 2016/2017

TABLE OF CONTENTS

Assessment Title	CS2004 Worksheet Based Assessment (2016-2017) Task 3 TSP
Module Leader	Dr Stephen Swift
Distribution Date	See below
Submission Deadline	See below
Feedback by	See below
Contribution to overall module assessment	30%
Indicative student time working on assessment	30 Hours not including scheduled laboratories
Word or Page Limit (if applicable)	No limit
Assessment Type (individual or group)	Individual

MAIN OBJECTIVE OF THE ASSESSMENT

This document is a description of the Task 3 (TSP – sheet 15) worksheet assessment for CS2004 Algorithms and their Applications (2016-2017).

Worksheet 15 "The Travelling Salesman Problem" will be verbally tested during a short viva, where the student is given an opportunity to explain their answers to the worksheet, demonstrate their program code and answer questions on what they have done.

The table below describes the assessment for ALL of the CS2004 assessment tasks, listed for information.

The learning outcomes (LO1, LO3, LO4) must be passed in the coursework to pass the module, i.e. if a student does not pass the threshold for the learning outcomes in task two then they must be passed in task three.

Task	Assignment Title	Available on Blackboard	Submission deadline	Viva Date	Weighting (%)
T1	Worksheet 3	The day of the corresponding lecture	15-01-2017 (Sunday week 16)	Not applicable	5% (of module)
T2	Sample questions from worksheets 4-14	The day of the corresponding lecture	12-03-2017 (Sunday week 24)	22-03-2017 (Wednesday week 26)	25% (of module)
T3	Worksheet 15	The day of the corresponding lecture	23-04-2017 (Sunday week 30)	02-05-2017 (Tuesday week 32)	30% (of module)

Description of the Assessment

The purpose of this assessment [Task 3] is to complete and report on CS2004 worksheet 15 "The Travelling Salesman Problem". What is expected for this assignment is fully documented in the worksheet description. The marking scheme for this task - task three (worksheet 15) can be found in Appendix A.

Learning Outcomes and Marking Criteria

Learning outcomes for the assessment	Assessment and marking criteria
LO1. Understand fundamental issues concerned with computation and algorithms	**See Appendix A and Worksheet**
LO2. Describe and evaluate both classic (e.g. sorting, searching, graph traversal) algorithms and meta-heuristic (e.g. RMHC, simulated annealing and population-based) algorithms	
LO3. Successfully implement (i.e. in a computer language) classic sorting, searching, graph traversal or non-population meta-heuristic algorithms	
LO4. Take real-world problems and identify relevant characteristics to guide the selection of an appropriate algorithm	

Format of the Assessment

The student **MUST** upload their worksheet Java code and associated report to **WiseFlow** after it has been completed and before the deadline as shown in the table above. Note that the report is not a formal report, but a document that contains any tables and graphs that may help to answer the questions within the worksheet. The student should not spend too much time making the report look "professional"; rather it should simply contain any graphs and tables (with a caption and any supporting text as appropriate) that show the experimental results from completing the worksheet. If a spreadsheet is deemed more appropriate, then this is acceptable. Note that where relevant the report should contain page numbers, the module title and code, along with the student's name and ID.

Submission Instructions

Please make sure that you submit the Java code part of your program along with any supporting report. This will be one or more .java file(s) and perhaps [probably] a .docx and/or .xlsx file(s). Make sure that you check that you have the correct file(s) [Java] by examining it/them in a text editor such as Notepad before you submit the file(s). No other part of the java/Eclipse application should be uploaded, e.g. .class or Eclipse project files. More details are provided on Blackboard and in the laboratories. All submissions should be a single ZIP file. Your student ID number must be used as the file name (e.g. 0123456.zip).

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.

Late Coursework

The clear expectation is that you will submit your coursework by the submission deadline stated in the study guide. In line with the University's policy on the late submission of coursework (revised in July 2016), coursework submitted up to 48 hours late will be accepted, but capped at a threshold pass (D- for undergraduate or C- for postgraduate). Work submitted over 48 hours after the stated deadline will automatically be given a fail grade (F).

Please refer to the Computer Science Student Handbook, available on Blackboard Learn, for information on submitting late work, penalties applied and procedures in the case of mitigating circumstances.

APPENDIX A - MARKING SCHEME (TASK 3 – WORKSHEET 15 - TSP)

For a particular grade, the minimum threshold for the following six criteria must be met (see below). For example, to attain a C grade, the lowest grade for one of the criteria would be a C.

1. DATASETS

F no datasets were used in the experiments.

E a very small number of datasets were used in the experiments.

D,C the number and size of the datasets are just about appropriate to support the results.

B,A the number and size of the datasets are appropriate to support the results.

2. EXPERIMENTAL DESIGN

F no working experiments were demonstrated or reported on.

E very inappropriate values were selected for the number of repeats and number of iterations.

D,C the number of repeats and number of iterations are just about appropriate to support the results. They are the same across all implemented methods.

B,A the number of repeats and number of iterations are appropriate to support the results. They are the same across all implemented methods.

3. EVALUATION

F no evaluation was demonstrated and/or the fitness function (and/or evaluation techniques) were not implemented correctly.

E,D only the fitness function is used to judge the quality of the experimental results.

C,B the fitness function and one other technique is used to judge the quality of the experimental results.

A the fitness function and two other techniques are used to judge the quality of the experimental results.

4. METHODS

F All attempted methods have been implemented very poorly and do not work correctly.

E All attempted methods have been implemented poorly and one or more almost work correctly.

D One implemented method works correctly.

C Two implemented methods work correctly.

B Three implemented methods work correctly.

A Four implemented methods work correctly.

5. ACCURACY OF RESULTS

F All results are completely inaccurate (as indicated by the evaluation critera #3 above)

E All results are inaccurate, but almost right (as indicated by the evaluation critera #3 above)

D,C,B All results are accurate (as indicated by the evaluation critera #3 above)

A All results are highly accurate (as indicated by the evaluation critera #3 above)

6. CODE QUALITY (STRUCTURE, COMMENTS, NAMING CONVENTIONS, BUGS)

F No program code runs.

E The code runs, but contains many bugs. The code quality is of an exceptionally low standard.

D The code runs and contains some bugs. The code quality is of a barely acceptable standard.

C The code runs and contains a few bugs. The code quality is of an acceptable standard.

B The code runs and contains almost no bugs. The code quality is of a high standard.

A The code runs and contains no apparent bugs. The code quality is of an exceptionally high standard.

The use of the + and - grade descriptors will be used to indicate border line cases within each grade boundary.