

AIP assessment – parts 3, 4 & 5

brief

As explained in the assessment guide: the AIP assessment is composed of 5 pieces of work, each accounting for 20% of the module mark. All pieces of work are carried out by assessment teams with 3-4 team members in each.

The first 2 pieces of work are presentation problems chosen for assessment teams by module tutors. The remaining 3 pieces of work are from the Computer Science problem collection (see links provided on the web page). The choice of these problems will be negotiated between assessment teams and tutors.

problem choice & negotiation

This negotiation/choice of problems will typically occur as follows...

1. assessment team examines problem collection & selects one or more candidate problems;
2. team checks that chosen problem(s) have not already been allocated to other teams (this will be recorded on the website and/or VLE);
3. assessment team registers interest in problem with tutor;
4. tutor confirms or rejects problem choice;

if problem choice is confirmed...

5. tutor updates problem allocation list;
6. tutor informs team of (i) any additional problem constraints (ii) problem I/O specification (which will often be different to that specified in the problem database – see note #1);
7. tutor & team agree a contractual deadline for the work (this will normally be about 2 weeks and not more than 3 weeks from when the team registers their interest);

submission & marking

8. team submits work (see other notes);
9. tutor marks work;
10. team & tutor meet for viva (this will normally involve all members of the group, take about 10 minutes and may be done in timetabled tutorial sessions);
11. tutor provides team with feedback (this will normally be done straight after the viva).

marking criteria

Specific marking criteria may differ a little from problem to problem (since the requirements may be different) but the following considerations will be used to judge the quality of the work...

1. to what extent does the work provide (or work towards) a solution to the specified problem;
2. how appropriate is the solution to the solution (ie: is this an elegant, well-chosen solution strategy or does it make harder work of the problem than necessary);
3. how clean is the code (clean code does not contain redundant expressions, unnecessary nesting of expressions like `cond` & `if`, etc, etc);
4. how thorough is the testing;
5. how well does the team explain their work (its approach, performance, etc) in the viva;

notes

1. Most problems are specified assuming input is entered as if it were typed at a console. For these assessments we will often change this part of the problem specification. We will typically expect input to be provided in list form and to be given as an argument to a function. This list may be nested and may contain additional symbols (to identify the roles of data). This will be decided by your tutor.

In addition, problems in the database are specified assuming that output will be printed. We typically expect output to be returned as the value from some function (& then printed by the Lisp evaluator). This output will be in the form of lists which may be nested and may also contain additional symbols. All input/output structures will be agreed (in advance) with tutors.