

- 1a Leg uit wat er met *middleware* bedoeld wordt. 5pt
- 1b Wat is een *multitiered client-server architecture*? 5pt
- 1c Geef voorbeelden hoe je een client-server systeem schaalbaar kunt maken. 5pt
-
- 2a Geef de principes van een algemene implementatie voor het aanroepen van een methode van een *remote object*. 5pt
- 2b Leg uit hoe een globale object referentie geïmplementeerd kan worden als een *proxy*. 5pt
- 2c Maak een korte vergelijking tussen *remote method invocations (RMIs)* en *remote procedure calls (RPCs)*. 5pt
- 2d Wat is een belangrijk schaalbaarheidsprobleem bij *remote objects* en hoe zou je dat op kunnen lossen? 10pt
-
- 3a Leg uit wat een transactie is. 5pt
- 3b Leg uit wat een gedistribueerde transactie is. 5pt
- 3c Wat is het verschil tussen *centralized two-phase locking*, *primary two-phase locking*, en *distributed two-phase locking*? 10pt
-
- 4a In a *k fout-tolerante* server groep, hoeveel servers zijn er nodig als je aanneemt dat er willekeurige fouten kunnen optreden. Leg je antwoord uit. 5pt
- 4b Volledige *failure transparency* bereiken in het algemeen, is vrijwel onmogelijk. Illustreer dit aan de hand van een voorbeeld. 10pt
- 4c Leg uit hoe *two-phase commit* werkt. 5pt
- 4d Leg uit wat er gebeurt als een proces faalt (*crashed*) tijdens het uitvoeren van het *two-phase commit* protocol. 10pt

Cijferbepaling: Het cijfer wordt bepaald door de punten die per onderdeel behaald zijn, bij elkaar op te tellen (totaal maximaal 90 punten), en vervolgens daar 10 extra punten bij te tellen. Er zijn dus totaal 100 punten te behalen.