

- 1 Geef een praktische oplossing voor het synchroniseren van meerdere *isochronous* datastromen. 5pt
- 2a Leg uit hoe een objectreferentie omgezet kan worden ("*unmarshaled*") naar een *proxy*. 5pt
- 2b Beschouw een systeem met mobiele agenten. Een agent is geïmplementeerd als een proces met een speciaal Java communicatie object. Communicatie vindt plaats door het aanroepen van methoden van het communicatie object van een agent (RMI). Geef het ontwerp van een *agent reference* voor dit systeem. 10pt
- 3a Op welke wijze zorgt de distributie van de DNS *name space* voor een afhankelijkheid tussen een naam en een locatie? 5pt
- 3b Leg het principe van *pointer caches* uit zoals toegepast in een hiërarchische *location service*. 5pt
- 4a Beschouw een verzameling processen logisch geordend in een ring. Een *token* circuleert met de klok mee. Als een proces het token ontvangt, mag het zijn kritieke sectie ingaan. Het token wordt doorgegeven aan de volgende bij het verlaten van de kritieke sectie, of anders wanneer het proces geen kritieke code had uit te voeren. Geef een simpele oplossing waarin het token circuleert alleen maar als er een proces is dat daadwerkelijk z'n kritieke sectie wil ingaan. 5pt
- 4b Voor je oplossing uit (a), geef een analyse van het totaal aantal boodschappen dat tussen alle paren van buurprocessen uitgewisseld wordt voordat een proces z'n kritieke sectie in mag. 10pt

Cijferbepaling: Het cijfer wordt bepaald door de punten die per onderdeel behaald zijn, bij elkaar op te tellen (totaal maximaal 45 punten), en vervolgens daar 5 extra punten bij te tellen. Er zijn dus totaal 50 punten te behalen.