

職能單元代碼	INM4R1414
職能單元名稱	建立網路腳本(scripts)
領域類別	資訊科技/網路規劃與建置管理
職能單元級別	4
工作任務與行為指標	<p>一、開發演算法以解決某問題</p> <ol style="list-style-type: none"> 1. 與客戶【註 1】及關鍵利害關係人【註 2】討論以找出問題與相關的腳本(scripts)要求 2. 採用精簡的軟體開發週期來建立腳本(scripts) 3. 開發演算法來解決問題與符合用戶端要求 4. 開發演算法來納入預期可能會發生的情況 5. 開發一個保證有成果的演算法 6. 示範結構、排序、選擇、迭代的使用 <p>二、建立程式碼</p> <ol style="list-style-type: none"> 1. 選擇適當的腳本(scripts)語言【註 3】 2. 展現基本語法規則及最佳實務的了解與應用 3. 選擇與使用語言資料種類、運算式來建立明確精簡的程式碼 4. 運用選擇、迭代與排序技術來控制腳本(scripts)執行流量 5. 使用檔案排序的輸入與輸出技術來取出與儲存資訊 6. 取得與利用使用者輸入，讓腳本(scripts)運作 7. 運用內部文件原則建立程式碼 8. 建立腳本(scripts)時依循組織規範【註 4】來開發可維護的程式碼 9. 建立腳本(scripts)時遵守編碼標準【註 5】 <p>三、使用作業系統工具</p> <ol style="list-style-type: none"> 1. 運用搜尋與排序工具來從作業系統 (OS)【註 6】的登錄輸出中選擇資訊 2. 執行控制措施來確保在發生重大事件時，透過作業系統登錄機制建立並維持作業的登入 3. 以 OS 排程功能註冊並執行腳本(scripts) <p>四、測試程式碼並除錯</p> <ol style="list-style-type: none"> 1. 設計、記錄並進行簡易測試，以確認程式碼符合設計規格 2. 從腳本(scripts)中找出未涵蓋或涵蓋錯誤的區域 3. 採取行動以確保程式碼符合安全政策

	<p>4. 採取行動以確保程式碼有適當的許可操作</p> <p>5. 運用適合的腳本(scripts)語言偵錯技術，來偵測並解決語法、邏輯與設計起源的錯誤</p> <p>五、以文件記錄</p> <p>1. 建立技術等級文件</p> <p>2. 建立使用者等級文件</p>
職能內涵 (K=knowledge 知識)	<p>一、概略知識：</p> <p>1. 演算法設計</p> <p>2. 作業系統元件，例如指令列介面、登錄檔、程式排程公用程式、開發方法、工具與公用程式以及測試方法</p> <p>二、各種腳本(scripts)情境偵錯</p> <p>三、程式設計結構化控制架構：排序、選擇、迭代</p> <p>四、腳本(scripts)技術與語言語法</p>
職能內涵 (S=skills 技能)	<p>一、與外部人員討論問題與要求的溝通技能</p> <p>二、讀寫技能：</p> <p>1. 產生並評估技術文件</p> <p>2. 產生使用者與同儕文件</p> <p>三、問題解決與應變管理技能：</p> <p>1. 在測試期間去除程式中的語法及語意錯誤</p> <p>2. 針對問題開發演算解決方案</p> <p>四、技術技能：</p> <p>1. 建立腳本(scripts)來執行作業系統自動化的任務</p> <p>2. 執行腳本(scripts)</p> <p>3. 與使用者透過腳本(scripts)互動</p> <p>4. 操控與萃取檔案中所含的資訊</p> <p>5. 針對各種情境使用內建的腳本(scripts)選項</p>
評量設計參考	<p>一、評量之關鍵面向/能力證明之證據：</p> <p>1. 以下各項能力之證據：</p> <p>(1) 針對一個設定好的流程，去開發一個解決方案的演算敘述</p> <p>(2) 針對一個問題敘述設計、記錄、建構與測試一個小型單一目的 OS 公用應用程式</p> <p>(3) 建立腳本(scripts)程式，可以存取系統上儲存的檔案的資訊，並運用系統公用程式在這些檔案中排序或找出資</p>

	<p>訊</p> <p>(4) 驗證並記錄腳本(scripts)結果</p> <p>二、評量所需情境與特定資源：</p> <ol style="list-style-type: none"> 1. 評量必須確保取得以下項目： <ol style="list-style-type: none"> (1) 技術要求 (2) 軟體開發環境 (3) 軟體測試環境 (4) 需要時適當的學習與評量支援 2. 於適用時，實體資源宜包含針對有特殊需求人士所修改的設備。 <p>三、評量方法：</p> <ol style="list-style-type: none"> 1. 宜採用多種評量方式來評量實務技能與知識。以下範例適合本單元： <ol style="list-style-type: none"> (1) 口頭或書面提問有關所需的技能與知識，例如：演算法設計、結構化控制架構、基本腳本(scripts)開發方法與其應用 (2) 評估演算法設計 (3) 評估小型單一目的 OS 公用應用程式 (4) 評估完成的設計計畫與文件 <p>四、評量輔助資訊：</p> <p>建議視情況與其他單元進行產業界、工作場域與工作職責方面的全面性評量。</p> <p>評量流程與技巧必須在文化方面要合宜，並且適合受評者的溝通技能等級、語言與讀寫算能力與應進行之工作。</p> <p>原住民與其他非我國背景者可能需要額外支援。</p> <p>若採用實習評量時，則宜結合目標提問(targeted questioning)方式評量所需的知識。</p>
說明與補充事項	<p>【註 1】用戶端包含：</p> <ul style="list-style-type: none"> ● 外部組織 ● 資通訊科技(ICT)公司 ● 個人 ● 內部部門 ● 內部員工 ● 服務業

	<p>【註 2】利害關係人包含：</p> <ul style="list-style-type: none"> ● 開發團隊 ● 資訊科技(IT)管理者或代表 ● 網路工程師 ● 專案團隊 ● 系統管理者 ● 使用者 <p>【註 3】演算法包含：</p> <ul style="list-style-type: none"> ● 流程圖 ● 虛擬碼 ● 結構化英語 <p>【註 4】劇本式語言包含：</p> <ul style="list-style-type: none"> ● C, C++ ● JavaScript ● Linux shell 腳本(scripts) ● Perl ● Python ● VBScript ● VB.Net ● Windows PowerShell. <p>【註 5】組織規範包含：</p> <ul style="list-style-type: none"> ● 通訊方法 ● 電子郵件內容 ● 爭議解決 ● 文件程序與樣本 ● 下載資訊與存取特定網站 ● 財務控制機制 ● 開啟有附件的電子郵件 ● 電子郵件與網際網路存取的個人使用 ● 病毒風險 <p>【註 6】編碼標準包含：</p> <ul style="list-style-type: none"> ● GNU 編碼標準 ● Java 編碼標準 ● 組織標準
--	--

	<p>【註 7】作業系統包含：</p> <ul style="list-style-type: none">● Linux● Mac● Novell● Windows.
--	---