教育部

創新典範資訊軟體人才培育計畫

（全程計畫：116 年 2 月至 120 年 1 月）

計畫簡要說明

壹、 計畫目標

貳、 推動架構及重點工作

參、 資訊軟體核心領域人才培育規劃參考

肆、 資訊相關系所定義

壹、計畫目標

一、 背景分析

(一)創新典範資訊軟體人才供需分析

1. 近年全球科技由單點創新轉向以 AI 為核心，結合雲原生、邊緣運算、高效能運算及系統軟體的整體數位創新。資策會 MIC 指出，AI 將推升 AI 基礎建設、雲邊協同運算及異質運算需求，未來產業競爭將更倚重系統軟體與運算架構整合能力。
2. AI PC、AI 手機與智慧終端普及，促使 AI 由雲端延伸至邊緣裝置，帶動軟硬體協同設計、作業系統及分散式系統能力的重要性。生成式 AI 加速程式開發、測試與資料分析，提高研發效率，但仍須仰賴人員判斷、整合與協作。未來人才除具備 AI 應用能力外，更須培養系統思維、跨域整合及問題解決能力。
3. 依據國發會 114-116 年針對人工智慧應用服務產業、通訊等重點產業人才供需調查及推估報告指出，產業欠缺人才職類占比前 5 大職類均與新興數位科技有關，以 AI 應用工程師、專業領域應用工程師、資料工程師、AI 與資料科學家、AI 專案經理等職缺最為缺乏，雖然資通訊學門畢業生人數每年維持在 1.8 萬至 1.9 萬人之間，但數據顯示畢業生投入勞動市場的比例顯著降低（投保人數由 112 年之 11,895 人大幅降至 113 年之 8,327 人），顯示軟體人才需求量在產業仍具大量缺口。
4. 綜觀全球發展，智慧化、系統化與永續已成為資通訊產業核心趨勢，人才培育亦須由應用導向，擴展至 AI 基礎技術、雲原生、分散式系統、系統軟體、高效能運算及軟硬體協同設計等關鍵能力。本計畫以「創新典範資訊軟體人才培育」為核心，建立兼具前瞻性、系統性及實務導向的人才培育機制，支撐國家數位競爭力。

(二)創新典範資訊軟體人才養成趨勢

1. 在生成式 AI 驅動的技術變革下，軟體工程師的角色正經歷從單純的「程式撰寫者」轉型為「問題定義者與系統整合者」的典範轉移。由於 AI 已能大幅自動化處理常規程式碼撰寫，未來的資訊人才必須具備更高層次的思維，專注於準確定義問題、拆解複雜需求並設計整體架構。
2. 生成式 AI 雖然能加速開發，但其缺乏對複雜軟體所需的邏輯理解、綜觀能力與精確性，因此人才養成的重點在於培養其對 AI 生成內容的判斷、驗證與修正能力，以確保系統的穩定性與安全性，進而支撐大規模的實務應用。
3. 為了厚植國家數位發展根基，人才培育正聚焦於雲原生、系統軟體、高效能運算及軟硬體協同設計等核心底層技術。運用「開源軟體發展模式」已成為培養

前瞻資訊人才的重要策略，透過參與國際開源社群接軌全球尖端技術，此模式不僅能緩解國內資通訊專業師資相對不足（僅占 6.07%）的困境，更能讓學生在「做中學」的過程中對國際專案做出實質貢獻，提升技術話語權。本計畫規劃透過開設主題式課群，引導師生成為國際開源專案的積極貢獻者，學習開源授權與代碼治理。

4. 未來軟體人才的養成趨向「問題導向學習（PBL）」與「跨域實踐」的緊密結合，在實際場域中解決真實問題，亦應具備敏捷（Agile）思維，能利用 AI 快速建立原型並驗證假設，將技術能力轉化為具產業價值的解決方案。這種趨勢不僅要求技術深度，更強調與領域專家溝通、理解場域痛點的能力，透過跨校、跨領域的實戰舞台，讓軟體開發成果能透過產學合作或技術移轉真正落地應用。

二、 願景與目標

為落實「國家希望工程-創新經濟、智慧國家」之國政願景及數位新社會之目標，因應產業人才需求趨勢與 AI 造成的典範轉移，本計畫聚焦於雲原生、系統軟體、高效能運算、軟硬體協同設計等資訊軟體核心領域，重點培育具備關鍵資訊軟體核心能力、以及能有效運用生成式 AI 並能與領域專家溝通協作的創新典範資訊軟體人才，以厚植國家數位發展與數位轉型基礎，提升國家數位韌性。本計畫具體推動模式如下：

(一)由資訊相關系所開設重點領域主題式課群，運用開源軟體發展模式，開發微服務/微系統解決實際場域重要問題，系統性培養學生整合及設計複雜軟體系統的核心能力。

(二)由資訊相關系所重構軟體開發核心課程，建立有效運用生成式 AI 之教學典範，透過跨域合作延伸至場域實踐，培養學生具備能與使用者互動、釐清需求與限制，並善用 AI 設計與實作以發展快速驗證解決方案之能力。

貳、推動架構及重點工作

本計畫整體推動架構如圖 1 所示，分為推動中心及示範系所兩大範疇。示範系所依推動目標分為 A/B 兩類。A 類計畫（核心領域前瞻人才培育系所）的目標為確保臺灣 ICT 產業的核心競爭力，補助資訊相關系所強化系統軟體核心能力的養成，培育能整合及設計複雜軟體系統的學生；B 類計畫（AI 跨域實踐人才培育系所）的目標為回應生成式 AI 對軟體開發的衝擊，補助資訊系所提出培育新世代軟體開發人才的教學典範，培養學生具備能與使用者互動、釐清需求與限制，並善用 AI 設計與實作以發展快速驗證解決方案之能力，加速應用軟體開發並延伸至場域實踐。推動中心各分項負責共通核心工作，提供教學及學習專業支援，協助示範系所順利執行計畫。

圖 1.創新典範資訊軟體人才培育計畫推動架構

一、 推動中心分項重點工作

推動中心除總計畫辦公室外，包括教學社群精進、創作資源共享、學習服務推動、價值創造推廣等 4 個分項，共同服務 A/B 兩類計畫，其架構如圖 2 所示，各分項負責推動之工作分述如下：

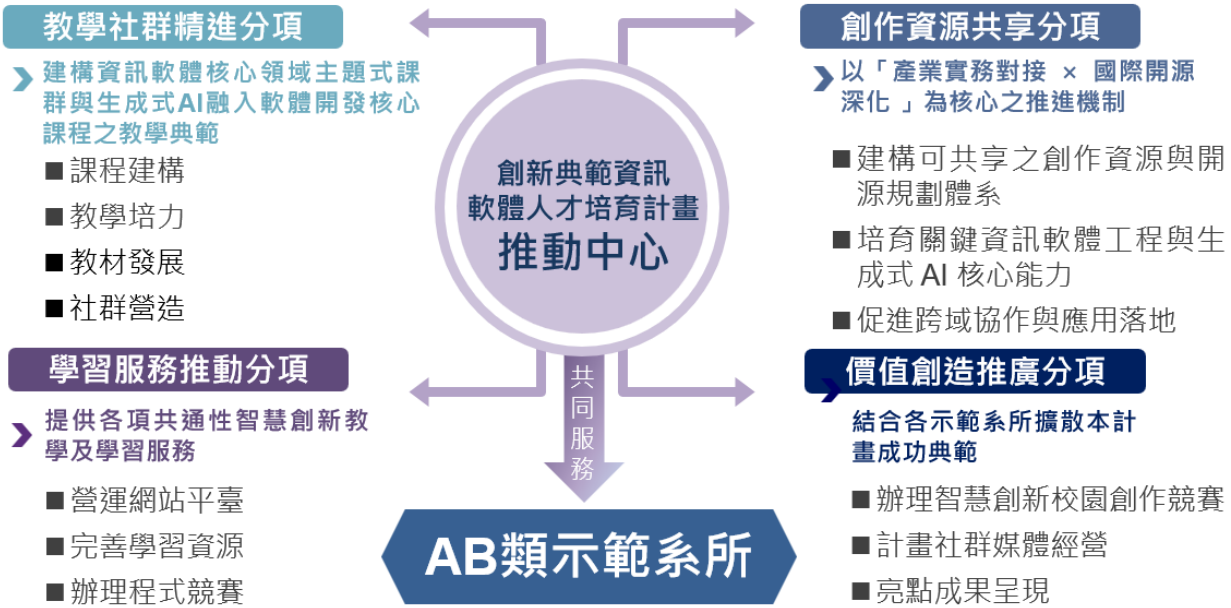

圖 2. 推動中心各分項推動工作

(一)教學社群精進分項：為協助示範系所規劃資訊軟體核心領域主題式課群與建立有效運用生成式 AI 進行軟體開發之教學典範，強化教師專業培力，推動創新教材研發及共享，全面強化軟體開發核心課程之教學品質與學習效能。執行策略包括課程建構、教學培力、教材發展、與社群營造等。為引導示範系所建構資訊軟體核心領域主題式課群與生成式 AI 融入軟體開發核心課程之教學典範，本分項將透過徵件說明會及觀摩交流等相關活動，促進經驗分享並傳達關鍵規劃重點。透過主題式課群、課程地圖與課程大綱之審查、教學設計的觀摩與分享、及實地訪視等方式，協助課群與課程規劃，進而推動課程之創新教學與優化。支持教師強化學生與生成式 AI 協同合作的軟體開發能力，並透過辦理教學培力活動與研習營，強化教師在生成式 AI 工具於軟體開發中之應用能力。營造教師社群，強化「教學共備」與「教材模組化」之深度，推動生成式 AI 融入課程之教學共備經驗與案例分享，並研議推動中心與示範學校共同開發教材之機制，整合軟體技術與場域應用，並接軌產業實務。

(二)創作資源共享分項：為協助示範系所，提供以「產業實務對接 × 國際開源深化」為核心之推進機制，建構可共享之創作資源與開源參與體系，協助示範系所培育關鍵資訊軟體工程與生成式 AI 核心能力，促進跨域協作與應用落地，執行策略包括透過媒合產業界真實需求案源與示範系所，引導示範系所以 PBL 為核心進行軟體研發，協助學生將技術能力轉化為具備產業價值的解決方案；引介國際指標性開源社

群與專案協作經驗，輔導示範系所參與全球化技術開發與開源治理，同時系統性普及開源授權（Licensing）與法律合規議題，協助示範系所在開發初期即納入合規規劃與代碼治理，提升我國軟體人才在開源生態系中的影響力與技術話語權；協同國家高速網路與計算中心或其它國內外企業，降低學生在實作生成式 AI 應用、微調大型模型及複雜系統開發時的技術門檻。本分項於新一期推動架構中，扮演「技術轉化」與「資源賦能」之關鍵支撐點，協助示範系所建構具實戰化與國際視野之軟體開發培育模式，將學術開發能量轉化為解決產業痛點之實質產出，並透過與國際開源生態接軌，提升整體計畫之技術影響力與國際競爭力。

(三)學習服務推動分項：本分項計畫旨在做為整體計畫之資訊平臺，提供各項共通性智慧創新教學及學習服務，包含營運網站平臺、完善程式學習資源、辦理程式設計競賽，普及數位創作風氣、精進學生的程式開發能力與強化數位創新技能，執行策略包括經營並維運計畫主網站、計畫徵件與管考平臺、教學資源平臺，及競賽官網等網站與平臺，以使計畫與競賽順利推動；除了豐富網站內容，同時落實資安防護，以打造便捷且安全的服務體驗。優化程式自學平臺，並辦理 ITSA 程式能力線上自我評量，透過豐富的教學輔助與學習資源，以提升教學品質並增進學生程式設計能力；為提升題庫品質，將提供關於演算法題目、程式能力線上自我評量與極客挑戰賽相關題目之 AI 協作解題教學影片與答題重點，以協助學生加速開發效率及突破學習盲點。辦理「ITSA 全國大專校院程式設計極客挑戰賽」，鼓勵學生透過生成式 AI 工具進行網頁設計，加強學生的程式設計能力與團隊合作能力，以培育新型態職能素養與技能之數位人才。

(四)價值創造推廣分項：本分項計畫以「競賽驅動創新」與「成果多元擴散」為核心，辦理競賽並透過社群媒體經營擴散計畫成果，執行策略包括透過辦理全國性指標「智慧創新暨跨域整合創作競賽」，建立跨校、跨領域的實戰舞臺；藉由產業熱門議題（如人工智慧、物聯網等）之題目設定，引導學生對接真實產業需求；於競賽期間規劃技術與實務訓練營，並於課程中導入生成式 AI 應用，邀請學術界及產業界之 AI 領域專家與資深架構師擔任講師，積極鼓勵學生將生成式 AI 輔助技術應用於軟體開發全週期，從根本提升學生在 AI 時代的實務接軌能力。持續維運計畫官

網之成果展示專區，建立主動式的亮點發掘機制，定期針對參賽團隊及示範系所進行實地訪談，透過技術專訪、文字報導與影音紀錄等形式，讓優質的教學經驗與開發模式能在不同校園間流通、參考與再創新，也提升計畫整體的社會能見度。主動發掘示範系所與優異團隊之亮點，提供多元的創業輔導資源與實習媒合機會，協助團隊深化作品的商業價值與實務應用，延續賽後的創作動能。綜合而言，本計畫將透過辦理跨校交流、營造教師社群、分享線上資源、舉辦學生競賽等方式，強化成果擴散機制，且不侷限於受補助學校，以擴大計畫執行成效，提升國家整體人才培育效益。

二、 示範系所推動機制

示範系所依照推動目標，分為兩類計畫，分述如下。

(一)「核心領域前瞻人才培育示範系所」（A類計畫）

此類計畫推動模式，係由資訊相關系所開設重點領域主題式課群，運用開源軟體發展模式，開發微服務/微系統解決實際場域重要問題，系統性培養學生整合及設計複雜軟體系統的核心能力。其核心機制如圖3，包括：

1. 擇定雲原生、系統軟體、高效能運算、軟硬體協同設計等資訊軟體核心領域一項或整合兩項為重點領域，發展其主題式課群，串聯重點領域核心知識或技能。系所可自訂重點領域內適當的主題式課群名稱。
2. 於課程中開發微服務/微系統，培育學生從「程式撰寫者」轉變為「問題定義者」與「系統整合者」，落實軟體系統開發及整合歷程。
3. 運用開源軟體發展模式，培育師生成為國際開源社群的積極貢獻者。
4. 建立重點領域知識地圖，產出主題式課群之教材（模組）或教案，並結合軟體專案開發及國際開源軟體社群參與經驗，推廣至其他學校。

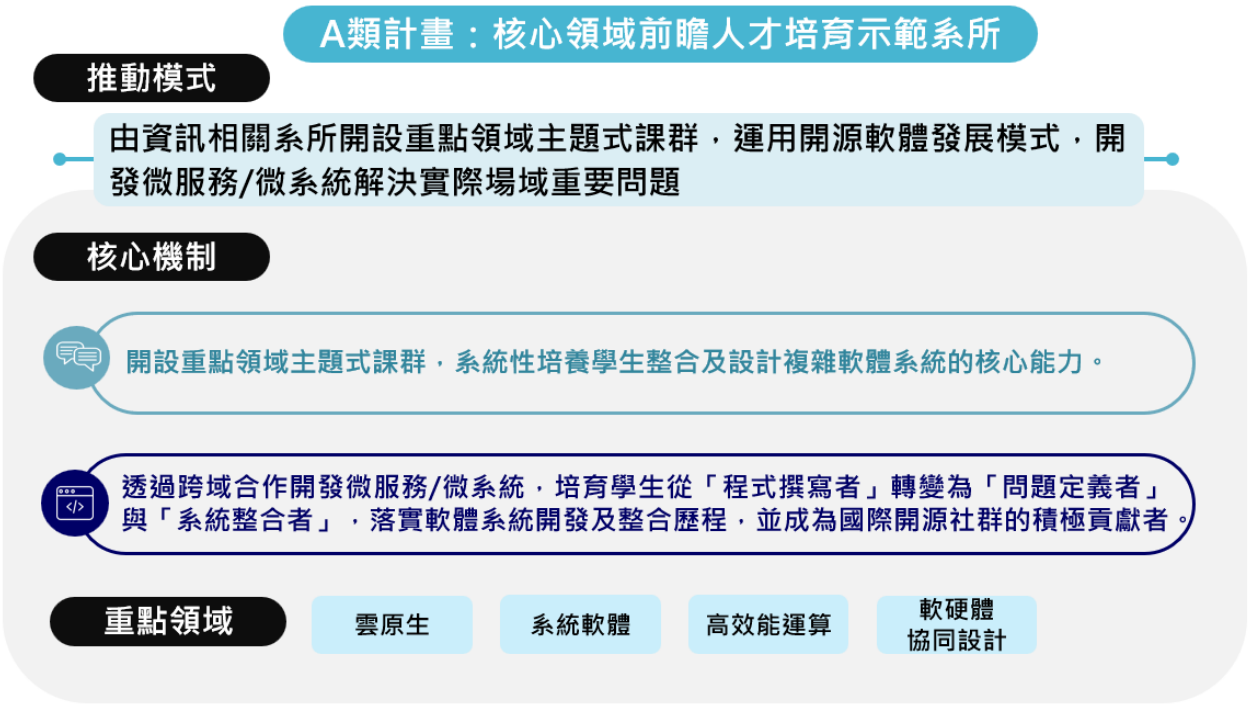

圖 3.核心領域前瞻人才培育示範系所（A 類計畫）

(二)「AI 跨域實踐人才培育示範系所」(B 類計畫)

此類計畫推動模式，係由資訊相關系所重構軟體開發核心課程，建立有效運用生成式 AI 之教學典範，透過跨域合作延伸至場域實踐，培養學生具備能與使用者互動、釐清需求與限制，並善用 AI 設計與實作以發展快速驗證解決方案之能力。其核心機制如圖 4，包括：

1. 因應生成式 AI，盤點及重構軟體開發核心課程。
2. 針對軟體開發核心課程，發展有效因應生成式 AI 開發模式之課程變革與教學精進作為。
3. 培育能充分運用 AI 並能跨域協作的新型人才，並將創作成果延伸至應用領域。
4. 示範系所須透過校內外教師社群進行教材、教案與教法之推廣，擴大其使用效益。

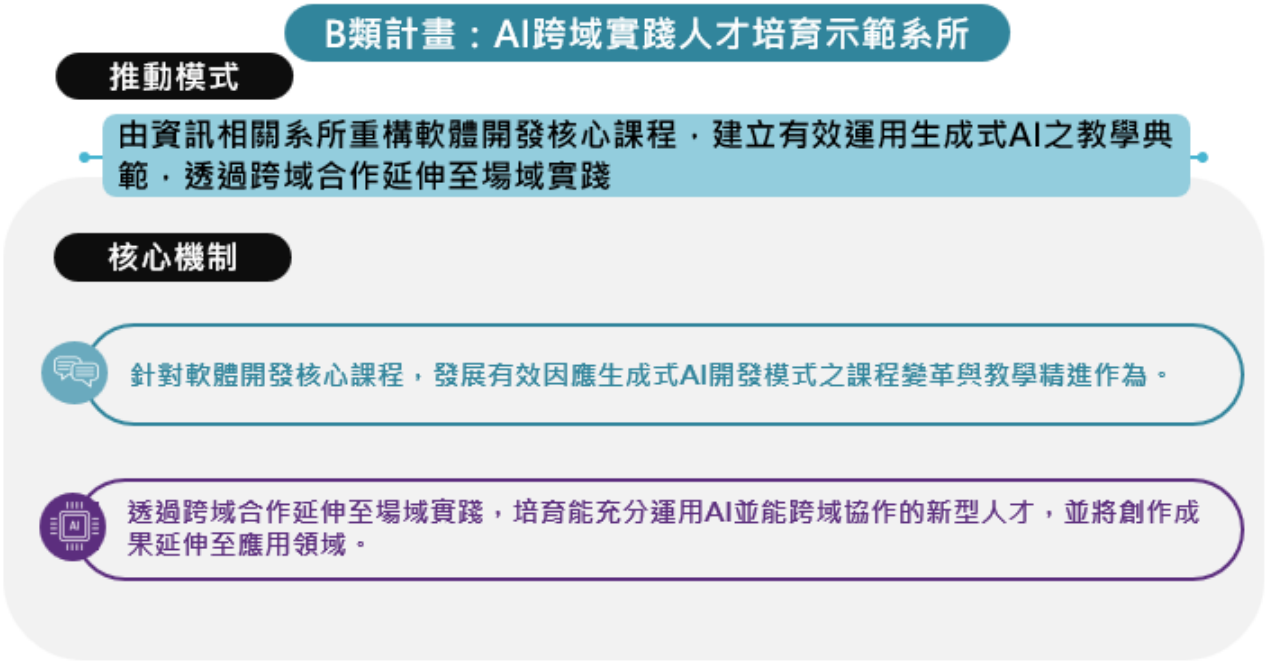

圖 4. AI 跨域實踐人才培育示範系所（B 類計畫）

參、資訊軟體核心領域人才培育規劃參考

下列說明僅供參考，提案單位可自行規劃合適之人才培育作法與內容。

一、 雲原生

(一)重要性及發展趨勢

隨著生成式 AI、數位轉型及雲端服務快速發展，軟體系統已由傳統桌面應用逐步轉向以雲端為核心的服務模式。現今試算表、文件編輯、影音串流、軟體開發工具及生成式 AI 等多採訂閱制提供服務，對系統持續可用性（Near Zero Downtime）、快速部署更新及大規模資料處理能力提出更高要求，使雲原生（Cloud Native）架構逐漸成為現代軟體開發的主流。雲原生是以容器化、微服務 DevOps、基礎設施即程式碼（IaC）與可觀測性等技術為核心，是開發當代分散式系統與高度彈性雲端服務不可或缺的基礎。透過自動化部署、彈性擴充及持續整合／持續部署（CI/CD），建構具高可用性、可維運性及快速迭代能力的分散式系統。國際雲原生計算基金會（Cloud Native Computing Foundation, CNCF）已建立完整的技術架構與知識體系，涵蓋基礎設施、執行環境、容器編排、系統監控、應用開發及平臺整合等面向，並已成為全球雲端技術發展的重要標準。近年來，雲原生技術已廣泛應用於全球資訊服務與企業數位轉型，國際大型雲端服務供應商及科技企業皆持續投入相關技術與人才培育，國內亦有多家科技領導企業與大專校院合作開設雲原生相關課程，培育符合產業需求之專業人才。

(二)核心能力需求

根據 ACM/IEEE/AAAI 提出的 CS 2023 Knowledge Model，和雲原生相關的 Konwledge Area 中，Data Management，Foundations of Programming Languages，Operating Systems，Software Engineering、Parallel and Distributed Computing 等五個面向構成了雲原生的基礎知識。其知識單元（Knowledge Unit, KU）構成直接與雲原生直接相關的核心知識，列舉如下:

1. 程式語言基礎：容器化與微服務架構多半依賴高度可平行化的運作模式，而不可變資料結構可以降低共享狀態衝突，提升服務的彈性與可靠性。雲端提供的 FaaS / serverless 平臺（如 AWS Lambda、Cloud Functions）也源自函數式思想設計小而獨立的運算邏輯。此外，雲原生系統大量依賴事件串流、非同步通訊、

以及處理 backpressure 能力的資料處理模式，事件驅動架構與反 Reactive Programming 正是這些模式的理論基礎。當代微服務透過 Kafka、NATS、RabbitMQ 等訊息系統進行解耦，而 Reactive Programming 的思維有助於設計非阻塞 I/O、主動資源管理、彈性錯誤處理的能力。

2. 作業系統：虛擬化技術是可說是雲端運算的基石。現代雲端環境從 IaaS 虛擬機（VM）到容器（Container），都依賴人工隔離機制、資源控制、命名空間與虛擬網路／虛擬儲存技術。理解虛擬化能協助學生掌握雲原生架構下可擴展、多租戶、動態調度與彈性伸縮的基礎。雲原生強調韌性（resilience）與高可用性，其核心就在於容錯機制。作業系統提供的監控、錯誤偵測、行程恢復、分散式鎖等概念，是建構雲端系統高可靠度的必要前提。在 Kubernetes 等雲原生平臺中，也有自我修復（self-healing）、自動重新排程（rescheduling）、ReplicaSet 保持多副本、以及滾動更新（rolling update）與回滾（rollback）等設計。這些設計的落實也都必須與底層 OS 提供的執行環境管理結合。
3. 平行與分散式運算：雲原生系統本質上就是一種分散式系統，通常由大量服務（可以看成 networked servers）組成，各服務透過 application layer 通訊協定溝通。理解分散式通訊的網路模型、延遲、非同步訊息、序列化格式（如 JSON、Protobuf）等基礎概念，有助於掌握這些服務如何協作、API Gateway 如何繞送分配訊息、以及 Kubernetes 如何透過 control plane 與 node 溝通。這些能力也直接關連到 cloud-native 下的 service discovery、負載平衡與 observability。雲端環境中各種分散式元件必須透過協調機制才能一致運作，其主要協調機制來出自於經典的分散式演算法例如 leader election、共識演算法（如 Raft、Paxos）。

(三)課程與教材發展參考規劃

根據上述整理的技術與知識，整理雲原生之課程與教材發展規劃，包含基礎、核心、進階課程（如圖 5 所示），引導學生從共通的基礎開始，接著建立專業核心能力，逐步引導學生發展進階能力。以下所列之課程為範例，因應新興科技的到來，系所宜根據系所特色與目標規劃將原本已開授之課程加以強化、開授新課程以涵蓋

新興科技、導入人工智慧作為設計或學習的工具。

圖 5. 雲原生領域之課程與教材發展規劃

二、 系統軟體

(一)重要性及發展趨勢

臺灣為半導體與資通訊（ICT）產品全球主要生產基地，資訊硬體產業產量為全球第一。而且，由於製造業占國內生產毛額（GDP）約 31%，因此 ICT 製造的發展牽動未來臺灣的發展。當前有許多資通訊產品均搭載系統軟體，以發揮硬體的特性、有效管理系統資源、支援應用軟體。根據資策會 MIC 於 2021 年底發布的趨勢觀察資料，半導體與資訊產業的高效能運算硬體發展可期，會驅動更多元與新型態的應用，其中相關的趨勢包含巨量數據的交換與分析、資訊安全的防護以及元宇宙相關單點應用逐漸成形。

資通訊產品技術進步快速，強調高效能運算、高度智慧化、快速聯網等功能升級以及資訊安全，因此系統的軟硬體越來越複雜。目前有許多產品已加入硬體加速器、高速網路、硬體資安機制，以支援高效率的資料分析與 AI 運算、快速連接邊緣與雲端服務、確保個人隱私與資料安全，而在這類系統中承上啟下的系統軟體便成為核心關鍵技術。如今包括本土的晶片設計公司與系統廠，以及知名的外商，為滿足日益提升的應用需求，皆在國內爭相聘用高階的系統軟體人才，以打造出符合當前與未來高速計算／智慧連網功能並且具競爭力的產品，深知系統軟體的運作優

劣與系統效能息息相關，關鍵在於系統軟體如何有效地分配系統資源、運用硬體加速器，以及系統開發人員如何分析、調校或優化系統效能，來滿足應用場景的需求。

(二)核心能力需求

系統軟體人才必須理解電腦系統的軟、硬體架構，並具備對系統所承載的應用軟體的工作特性的分析能力，才能妥善地安排運算任務的執行以及配置所需的資源，以發揮硬體的運算能力，達成預設的效能規格、滿足使用者的需求、乃至於超越競爭對手。本類型人才所強調的核心能力包含：

1. 軟硬體架構與技術：瞭解電腦系統的軟體與硬體架構以及軟硬體的互動，為系統軟體人才需要培養的核心技術能力。尤其是系統軟體涵蓋範圍甚廣，例如：作業系統、效能分析工具、編譯器、工具集（binary utilities）以及虛擬機器監視器（hypervisor）等。較為廣義的系統軟體定義則可涵蓋中介軟體與韌體，例如現在常被用來建造深度學習模型的軟體框架 TensorFlow 即可視為是中介軟體的一種，又例如量子電腦系統發展與運作所需的軟體，包括控制量子元件的底層韌體、量子程式編譯器、量子電路模擬器等，也都屬於系統軟體的範疇。
2. 系統相關的程式設計：程式設計能力也是重要的基礎技術能力之一，傳統的系統軟體大多由 C 語言撰寫而成，以提供快速的執行效率，如 Linux 作業系統。因此，為了能夠開發系統軟體的核心功能或是修改系統軟體提升系統效能，必須能理解較爲貼近硬體的程式設計語言以及相關的軟體開發工具。
3. 理解與分析特定應用的能力：為了針對特定應用系統進行最佳化，系統軟體人才必須要有分析特定應用的能力。然而應用領域何其多，不可能在短期間樣樣精通，因此真正要建立的是如何快速瞭解特定應用、知道其運作原理與流程，找出應用軟體的特性與需求，據以調整與修改系統軟體的實作，甚至可以提供硬體架構設計的修改建議，以期可以提升整體系統運行效率。
4. 理解與運用人工智慧的能力：近年人工智慧技術進步神速，系統軟體人才應該有能力善用人工智慧工具來輔助上述(1)(2)(3)項的知識技能，提升設計、運用、優化系統軟體的能力，解決日益複雜的系統問題。
5. 團隊合作與溝通協調：能跨領域溝通，清楚表達自己的想法，同時可以傾聽與整合不同背景組成的團隊成員意見（如：與硬體設計者與應用軟體開發者溝

通）。掌握不同領域的專業人士之間在合作時，可能遇到的各種問題，培養同理心的做事態度。

(三)課程與教材發展參考規劃

根據上述整理的技術與知識，整理系統軟體之課程與教材發展規劃，包含基礎、核心、與進階課程（如圖 6 所示），引導學生從共通的基礎開始，接著建立專業核心能力，逐步引導學生發展進階能力。以下所列之課程為範例，因應新興科技的到來，系所宜根據系所特色與目標規劃將原本已開授之課程加以強化、開授新課程以涵蓋新興科技、導入人工智慧作為設計或學習的工具。

圖 6 系統軟體領域之課程與教材發展規劃

三、 高效能運算

(一)重要性及發展趨勢

隨著近年來年生成式 AI 與大數據技術的爆發式成長，高效能運算正成為驅動全球科技競爭、經濟增長與科學發現的核心動力。高效能運算（High-Performance Computing, 簡稱 HPC），是指透過匯集大量的計算資源（處理器、記憶體、網路），以遠超一般電腦的速度，處理極其複雜的計算任務或龐大數據的技術。因此，只要人們持續不斷追求更大的計算力來用電腦解決更複雜的問題，那就無法缺少高效能運算的能力與技術。

高效能運算在基礎科學領域一直扮演著不可或缺的角色，科學家透過收集的大量實驗觀察數據和設計複雜精密的數學模型，利用電腦大量模擬計算得到電腦模擬的推測結果，再進一步分析數據驗證理論及調整實驗設計，加速科學研究與發展的

過程。尤其近年來人工智慧的興起更將人們對於計算的需求推升到一個史無前例的境界。眾所皆知現代的 AI 發展核心就是依靠深度學習這個關鍵技術，而深度學習模型的計算需求不斷隨著訓練資料量與模型參數量的增長而在快速成長。因此資料、模型、計算力，是 AI 發展不可或缺的三大支柱，而且隨著資料與模型朝向開源、公開的方向前進，計算力的短缺與計算成本的增加反而成為目前使用者在應用發展 AI 時遇到的主要瓶頸。而且隨著 AI 正在快速的滲透到各應用領域，影響的面相從科技擴展到經濟、法律、人文倫理等，高效能運算的重要性也已提升至戰略高度。它不僅是商業競爭的利器，更是國家在國防安全、公共衛生與永續發展上的競爭力指標。

(二)核心能力需求

整合上述背景，本類型人才所強調的核心能力包含：

1. 程式與系統的實作能力：高效能運算旨在透過有效利用系統資源提升程式效能，因此要從上至下，對軟體的計算以及硬體的運作都要十分熟悉。高效能運算的人才不僅要具備將應用程式安裝執行在客製化的電腦設備與環境的能力，更要具備能研究實作系統軟體提升運算效能的能力，因此必須具備比應用程式端更深的程式實作與軟體開發能力，包括平行程式、編譯器設計、Linux 核心程式函示庫開發優化等。
2. 效能量測與分析的知識技術：高效能運算的目標是提升運算效能，但這不僅需要具備提升效能的技術，更要先具備效能量測、分析的能力，才能掌握效能的瓶頸和問題的癥結點，採用有效的解決方法。因此高效能運算人才必須認識各種效能的量測指標（Throughput, Utilization, Latency, Bandwidth）、效能分析模型（如 Roofline Model, Amdahl’s Law, Operational Intensity 等）、效能量測工具（如 Benchamrk, Profiler, Performance Counter）等，並且具備實際對各種程式、系統、甚至硬體做效能量測分析的能力。
3. 叢集電腦系統操作與建置能力：高效能運算的計算環境是透過集結多台電腦透過大量平行運算所實現的，也就是所謂的叢集電腦或超級電腦。這些系統所使用的系統架構、系統軟體、管理方式、甚至程式執行開發環境都有別於一般電

腦。而且這些系統都是多人多工、資源共享的運算環境，所以在系統操作、函示庫安裝管理、資源調度上的操作與使用方式也不相同。因此高效能運算人才應具備建置這種系統的經驗，並掌握這種電腦的正確使用，從而認識高效能運算的軟硬體架構與運作原理。

4. 解決實務與跨領域問題的經驗：高效能運算必須結合應用領域才能展現效能提升的價值。如同傳統高效能運算被用來解決各種科學研究領域的計算問題，如今高效能運算透過 AI 的結合會被運用到更廣泛的問題領域，因此高效能運算的人才一定要去具備解決跨領域實務問題的核心能力。
5. 掌握趨勢的自學與探索能力：高效能運算是追求最新科技發展的技術，因此最新的技術與知識一定要靠自己透過網路搜尋、會議研討會參與、技術交流、甚至直接與產業互動，如實習、參訪等方式才能取得。因此學校一方面要培養學生的自學和探索能力，另一方面要在課程中安排及鼓勵學生透過校外的學習管道（如競賽、實習、研討會）去獲取新知。

(三)課程與教材發展參考規劃

根據上述整理的技術與知識，整理高效能運算之課程與教材發展規劃，包含基礎、核心、與進階課程（如圖 7 所示），引導學生從共通的基礎開始，接著建立專業核心能力，逐步引導學生發展進階能力。以下所列之課程為範例，因應新興科技的到來，系所宜根據系所特色與目標規劃將原本已開授之課程加以強化、開授新課程以涵蓋新興科技、導入人工智慧作為設計或學習的工具。

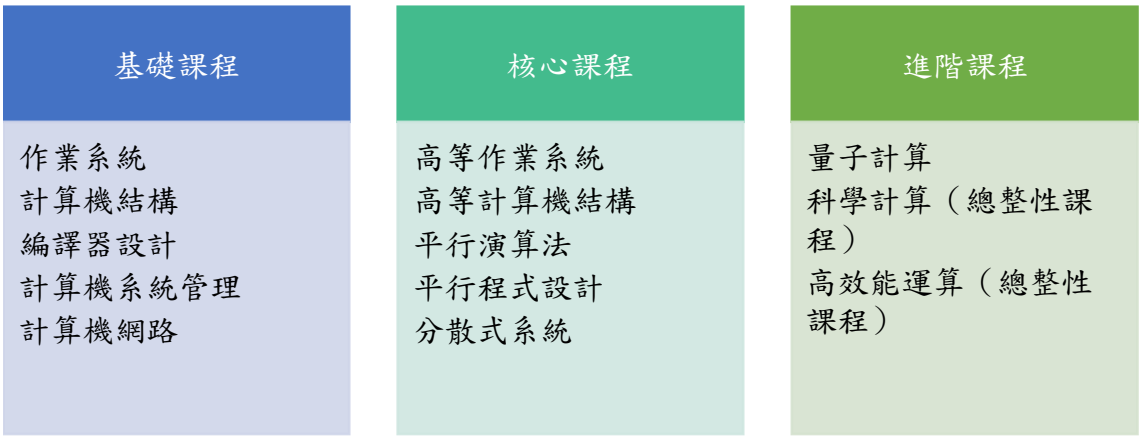

圖 7. 高效能運算領域之課程與教材發展規劃

四、 軟硬體協同設計

(一)重要性及發展趨勢

隨著 AI、5G、邊緣運算、物聯網、量子運算、新興非揮發性記憶體（如 ReRAM、PCM）與加速器架構快速發展，單純依靠軟體或硬體的設計已無法滿足現代系統在能源效率、運算性能、即時性與可靠度上的需求。軟硬體共同設計（Software–Hardware Codesign）因此成為新世代資訊科技的重要核心能力，廣泛應用於 AI 晶片、先進 SoC、嵌入式系統、自駕車平臺、機器人、雲端／邊緣協同系統乃至於量子前處理裝置等領域。

根據產業趨勢，Google TPU、NVIDIA CUDA、Apple Neural Engine、RISC-V 開源架構以及各類 AI 加速器持續推升軟硬體深度整合的人才需求；包含計算圖最佳化、模型壓縮、硬體友善演算法（hardware-friendly algorithms）、系統軟體最佳化、編譯器技術（Compiler/LLVM）、以及硬體架構共同調校（co-optimization）等能力皆屬企業急需之核心技能。

在能源受限的端側（edge）設備上，如何讓 AI 模型在小型裝置上運作，亦高度仰賴模型設計、硬體架構設計與系統管理三者的共同優化，突顯軟硬體共同設計的重要性。國際開放硬體（Open Hardware）、開源軟體（Open Source Software）與開源編譯器（如 MLIR、TVM）的蓬勃發展，更加速此領域技術的進一步落地。

(二)核心能力需求

以下列舉軟硬體協同設計人才所需要培養的專業核心技術、特定應用相關知識與團隊合作能力：

1. 軟硬體整合與分析能力：包含具備數位邏輯、計算機結構、嵌入式系統、SoC 架構基礎；了解軟體層（演算法、程式語言、系統軟體）與硬體層（微架構、記憶體、匯流排、加速器）的互動關係；能分析系統瓶頸（performance bottleneck）、能源效率（energy efficiency）與記憶體行為（memory behavior）。
2. 硬體友善演算法設計能力：包含具備模型壓縮、量化（quantization）、剪枝（pruning）與硬體友善運算（如 Winograd、tiling、reuse-aware algorithms）等技術；了解 AI workload 與硬體架構（CPU/GPU/AI-Chip/NPU）的最佳化策略。
3. 編譯器、系統軟體與工具鏈能力：包含熟悉編譯器前端／後端（Frontend/

Backend）、IR、MLIR、TVM、LLVM 等工具鏈；能分析程式在硬體上的執行行為並進行最佳化；了解驅動程式、韌體（Firmware）、RTOS、Virtualization 等系統層軟體設計概念。

4. 硬體實作與驗證能力：包含能使用 HDL（Verilog/VHDL/SystemVerilog）或 HLS 進行架構原型開發；熟悉 FPGA、模擬器（Simulator）/Emulator、形式驗證（Formal Verification）與效能測試方法。
5. 跨領域系統整合與團隊協作能力：包含能與演算法開發者、資工系統軟體開發者、電機硬體設計者協作；理解系統需求並轉化為軟硬體規格；熟悉敏捷開發（Agile）與跨功能團隊溝通方式。

(三)課程與教材發展參考規劃

根據上述整理的技術與知識，整理軟硬體協同設計之課程與教材發展規劃，包含基礎、核心、與進階課程（如圖 8 所示），引導學生從共通的基礎開始，接著建

基礎課程	核心課程	進階課程
計算機結構 作業系統 系統程式 硬體描述語言 FPGA 與硬體加速器設計	高等計算機結構 嵌入式系統 編譯器設計（Compiler/LLVM/MLIR） 系統效能評估與分析（Performance Modeling and Profiling） GPU 程式設計（CUDA、OpenCL）	軟硬體協同設計實務（總整性課程） 高階合成 (HLS) 與加速器原型建構（總整性課程） AI Accelerator Architecture Hardware-aware Machine Learning

立專業核心能力，逐步引導學生發展進階能力。以下所列之課程為範例，因應新興科技的到來，系所宜根據系所特色與目標規劃將原本已開授之課程加以強化、開授新課程以涵蓋新興科技、導入人工智慧作為設計或學習的工具。

圖 8. 軟硬體協同設計領域之課程與教材發展規劃

肆、資訊相關系所定義

一、 以電算機之工程、網路、軟體開發、系統設計為主教學單位（系所或學位學程）。

二、 畢業學分中之必修課程包含「物件導向程式設計」、「計算機組織」、「計算機結構」、「作業系統」等課程（以上其中之一）的系所。

三、 如符合前述 2 點之一即定義為資訊相關系所，請參考下表所列之系所名稱。（本表僅供參考，未將所有資訊相關系所列出，請依上述 2 點之定義進行判別。）

學類	資訊相關學系所
電腦運用學類 資料庫、網路設計及管理學類 軟體及應用的開發與分析學類 其他資訊通訊科技學類	資訊工程學系所 資訊科學與工程學系所 資訊管理學系所 資訊與通訊學系所 資訊科技通訊學系所 資訊應用學系所 資訊傳播工程學系所 資訊安全研究所 資訊與系統應用研究所 資訊網路與多媒體研究所 資訊電機不分系 電腦與通訊學系 軟體工程學系所 電子工程學系所 電機工程學系所 通訊工程學系所 網路工程研究所 醫學資訊學系所