

occupation	Qualifications/Skills	Training principles
Programmers	Proficiency in Programming Languages: Mastery of programming languages such as Python, Java, C++, JavaScript, or others depending on the specific requirements of the job ความเชี่ยวชาญในภาษาการเขียนโปรแกรม: ความเชี่ยวชาญในภาษาการเขียนโปรแกรมเช่น Python, Java, C++, JavaScript หรืออื่นๆ ขึ้นอยู่กับข้อกำหนดเฉพาะของงาน	learn
		Practice
	Problem-Solving Skills: Ability to analyze problems, develop algorithms, and write efficient code to solve complex problems and implement software solutions ทักษะการแก้ปัญหา: ความสามารถในการวิเคราะห์ปัญหา พัฒนาอัลกอริทึม และเขียนโค้ดที่มีประสิทธิภาพเพื่อแก้ไขปัญหาที่ซับซ้อน และใช้โซลูชันซอฟต์แวร์	learn
		Practice
	Data Structures and Algorithms: Understanding of data structures and algorithms, including their efficiency and suitability for different programming tasks โครงสร้างข้อมูลและอัลกอริทึม: ความเข้าใจเกี่ยวกับโครงสร้างข้อมูลและอัลกอริทึม รวมถึงประสิทธิภาพและความเหมาะสมสำหรับงานการเขียนโปรแกรมต่างๆ	learn
		Practice
	Software Development Life Cycle (SDLC): Knowledge of software development methodologies such as Agile or Scrum, and familiarity with the entire software development life cycle วงจรชีวิตการพัฒนาซอฟต์แวร์ (SDLC): ความรู้เกี่ยวกับวิธีการพัฒนาซอฟต์แวร์ เช่น Agile หรือ Scrum และความคุ้นเคยกับวงจรชีวิตการพัฒนาซอฟต์แวร์ทั้งหมด	learn
		Practice
	Debugging and Testing: Skill in debugging code, identifying errors, and writing unit tests to ensure software reliability and maintainability การดีบั๊กและการทดสอบ: ทักษะในการดีบั๊กโค้ด การระบุข้อผิดพลาด และการเขียนการทดสอบหน่วยเพื่อให้มั่นใจถึงความน่าเชื่อถือและการบำรุงรักษาซอฟต์แวร์	learn
		Practice

	Version Control Systems: Experience with version control systems like Git, SVN, or Mercurial for collaborative development and code management ระบบควบคุมเวอร์ชัน: มีประสบการณ์กับระบบควบคุมเวอร์ชันเช่น Git, SVN หรือ Mercurial เพื่อการพัฒนา ร่วมกันและการจัดการโค้ด	learn
		Practice
	Object-Oriented Design: Proficiency in object-oriented programming principles and design patterns for developing scalable and maintainable software applications การออกแบบเชิงวัตถุ: ความเชี่ยวชาญในหลักการเขียนโปรแกรมเชิงวัตถุและรูปแบบการออกแบบสำหรับการพัฒนาแอปพลิเคชันซอฟต์แวร์ที่ปรับขนาดได้และบำรุงรักษาได้	learn
		Practice
	Web Development: Familiarity with web development technologies such as HTML, CSS, and frameworks like React, Angular, or Vue.js for front-end development, as well as back-end technologies like Node.js, Django, or Flask การพัฒนาเว็บ: ความคุ้นเคยกับเทคโนโลยีการพัฒนาเว็บ เช่น HTML, CSS และเฟรมเวิร์ก เช่น React, Angular หรือ Vue.js สำหรับการพัฒนาสถาปัตยกรรม เช่นเดียวกับเทคโนโลยีแบ็คเอนด์ เช่น Node.js, Django หรือ Flask	learn
		Practice
	Database Management: Knowledge of relational databases (e.g., MySQL, PostgreSQL) and NoSQL databases (e.g., MongoDB, Redis) for data storage and retrieval การจัดการฐานข้อมูล: ความรู้เกี่ยวกับฐานข้อมูลเชิงสัมพันธ์ (เช่น MySQL, PostgreSQL) และฐานข้อมูล NoSQL (เช่น MongoDB, Redis) สำหรับการจัดเก็บและการเรียกค้นข้อมูล	learn
		Practice
	Continuous Learning: Commitment to staying updated on new programming languages, frameworks, tools, and best practices through self-learning, online courses, or professional development opportunities การเรียนรู้อย่างต่อเนื่อง: มุ่งมั่นที่จะติดตามภาษาการเขียนโปรแกรม กรอบงาน เครื่องมือ และแนวปฏิบัติที่ดี	learn
		Practice

[illegible]