

occupation	Qualifications/Skills	Training principles
Programmers	<p>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: ความเชี่ยวชาญในภาษาการเขียนโปรแกรม เช่น Python, Java, C++, JavaScript หรืออื่น ๆ ขึ้นอยู่กับข้อกำหนดเฉพาะของงาน</p>	learn
		Practice
	<p>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 และใช้ซอฟต์แวร์</p>	learn
		Practice
	<p>Data Structures and Algorithms: Understanding of data structures and algorithms, including their efficiency and suitability for different programming tasks โครงสร้างข้อมูลและอัลกอริทึม: ความเข้าใจเกี่ยวกับโครงสร้างข้อมูลและอัลกอริทึม รวมถึงประสิทธิภาพและความเหมาะสมสำหรับงานการเขียนโปรแกรมต่างๆ</p>	learn
		Practice
	<p>Software Development Life Cycle (SDLC): Knowledge of software development methodologies such as Agile or Scrum, and familiarity with the entire software development life cycle ชีวิตการพัฒนาซอฟต์แวร์ (SDLC): ความรู้เกี่ยวกับวิธีการพัฒนาซอฟต์แวร์ เช่น Agile หรือ Scrum และความคุ้นเคยกับวงจรชีวิตการพัฒนาซอฟต์แวร์ทั้งหมด</p>	learn
		Practice
	<p>Debugging and Testing: Skill in debugging code, identifying errors, and writing unit tests to ensure software reliability and maintainability การดีบักและการทดสอบ: ทักษะในการดีบักโค้ด การระบุข้อผิดพลาด และการเขียนการทดสอบหน่วยเพื่อให้มั่นใจถึงความน่าเชื่อถือและการบำรุงรักษาซอฟต์แวร์</p>	learn
		Practice

	<p>Version Control Systems: Experience with version control systems like Git, SVN, or Mercurial for collaborative development and code management</p> <p>ระบบควบคุมเวอร์ชัน: มีประสบการณ์กับระบบควบคุมเวอร์ชันเช่น Git, SVN หรือ Mercurial เพื่อการพัฒนาท่วมกันและการจัดการโค้ด</p>	learn
	<p>Object-Oriented Design: Proficiency in object-oriented programming principles and design patterns for developing scalable and maintainable software applications</p> <p>การออกแบบเชิงวัตถุ: ความเขี่ยงข่ายในหลักการเขียนโปรแกรมเชิงวัตถุและรูปแบบการออกแบบสำหรับการพัฒนาแอปพลิเคชันซอฟต์แวร์ที่ปรับขนาดได้และบ้ารุ่งรักษาระดับ</p>	learn
	<p>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</p> <p>การพัฒนาเว็บ: ความคุ้นเคยกับเทคโนโลยีการพัฒนาเว็บ เช่น HTML, CSS และเฟรมเวิร์ก เช่น React, Angular หรือ Vue.js สำหรับการพัฒนาส่วนหน้า เช่นเดียวกับเทคโนโลยีเบ็คเอนด์ เช่น Node.js, Django หรือ Flask</p>	learn
	<p>Database Management: Knowledge of relational databases (e.g., MySQL, PostgreSQL) and NoSQL databases (e.g., MongoDB, Redis) for data storage and retrieval</p> <p>การจัดการฐานข้อมูล: ความรู้เกี่ยวกับฐานข้อมูลเชิงสัมพันธ์ (เช่น MySQL, PostgreSQL) และฐานข้อมูล NoSQL (เช่น MongoDB, Redis) สำหรับการจัดเก็บและการเรียกดูข้อมูล</p>	learn
	<p>Continuous Learning: Commitment to staying updated on new programming languages, frameworks, tools, and best practices through self-learning, online courses, or professional development opportunities</p> <p>การเรียนรู้อย่างต่อเนื่อง: มุ่งมั่นที่จะติดตามภาษาการเขียนโปรแกรม กระบวนการ เครื่องมือ และแนวปฏิบัติที่ดี</p>	learn
		Practice

