**Standard Software Systems Development Questions**

Chioma Chance

Colorado State University Global

CSC505

Dr. Mujeye

01/11/2025

**Explanation for the Script:**

This program fulfills the assignment requirements by showcasing basic functionality in Python. It uses the print() function to output text to the console, reflecting a simple, beginner-level understanding of Python scripting. The program structure includes a main() function and a proper entry point with the if \_\_name\_\_ == "\_\_main\_\_": statement, which is considered good practice in Python programming.

**Updated Answers to the Development Questions:**

**1. Why does it take so long to get software finished?**  
It takes time to develop software because the process involves multiple stages, including planning, designing, coding, testing, and debugging. Each phase requires collaboration, problem-solving, and thorough validation to ensure the final product meets user requirements.

**2. Why are development costs so high?**  
Development costs are high due to factors like the need for skilled developers, expensive tools and technologies, and time spent on design, implementation, and quality assurance. Additionally, maintaining and updating software after release adds to the overall cost.

**3. Why can't we find all errors before we give the software to our customers?**  
Not all errors can be found due to the complexity of modern software and the variety of use cases. Testing can only cover a subset of possible scenarios, and unforeseen edge cases may arise when the software is used in real-world environments.

**4. Why do we spend so much time and effort maintaining existing programs?**  
Maintenance is necessary to fix bugs, adapt to new technologies, and meet changing user needs. As software evolves, its complexity increases, requiring significant effort to ensure it remains functional and secure.

**5. Why do we continue to have difficulty in measuring progress, as software is being developed and maintained?**  
Software progress is difficult to measure because it's not just about the number of lines of code written or tasks completed. Quality, functionality, and user satisfaction also play critical roles, making progress more subjective and harder to quantify.

**References**

1. Pressman, R. & Maxim, B. (2020). *Software Engineering: A Practitioner’s Approach* (9th ed.). McGraw-Hill Education.
2. DeMarco, T. (1982). *Controlling Software Projects: Management, Measurement, and Estimates*. Yourdon Press.
3. Sommerville, I. (2015). *Software Engineering* (10th ed.). Pearson Education.