

### Key Vocabulary

- **Evaluation:** Thinking about what we like or dislike about something and how it can be improved.
- **Success Criteria:** A defined set of standards or rules that tell us what a program should do in order to be successful.
- **User Interface and User Experience (UI and UX):** Everything a user sees and hears in a program, representing how the user feels when using the program (this does not include the underlying code).
- **Systematic Testing:** Planning a careful, step-by-step way to test a program instead of testing it randomly.

### Questions & Answers (Q&A)

#### Q1: What does evaluation mean in computer programming?

A: Evaluation means thinking about what we like or dislike about a program and identifying how it can be improved. It is an essential way to keep learning and make projects better.

#### Q2: What are success criteria, and why are they helpful?

A: Success criteria are defined rules or targets that tell us what a program must do to be successful. They focus our evaluations and make it easier to see exact improvements to make.

#### Q3: What is included in the user experience (UI and UX) of a program?

A: User interface and user experience includes everything a user sees and hears, as well as how they feel while using the software. It does not include the underlying programming code because users almost never see the code.

#### Q4: Why do developers release software updates for digital devices?

A: Developers release updates to fix errors in the code and improve the user experience. Updates also include important security improvements that keep devices safe and secure.

#### Q5: What happened to the famous online game that was released without proper testing?

A: The game which is not tested will be full of bugs, leading to worldwide user complaints and very bad reviews. The company had to refund everyone's money and lost millions of dollars.

#### Q6: What is systematic testing?

A: Systematic testing means planning a careful, organized way to test a program rather than testing it at random. For example, in a calculator app, this means testing all additions first, followed systematically by subtraction, multiplication, and division.

#### Q7: Why is it important to run program code multiple times during testing?

A: The more times you run a program successfully, the more confident you can be that it works correctly and contains no bugs. Testing multiple times ensures it gives the expected result consistently.

**Q8: What kinds of unexpected inputs should you test in a program like Rock, Paper, Scissors?**

A: You should test unexpected words (such as "bananas"), numbers, or single letters. This shows you how the program reacts when a user enters data it did not anticipate.

**Q9: Why should programmers feel positive when they discover a bug in their code?**

A: Finding a bug means the programmer now knows the error is there and can solve the problem. Correcting that mistake directly improves the program.

#### **4. Second-Order Thinking Skills Questions**

**Q1: A student tests a Rock, Paper, Scissors game once with "rock" and says, "It works, is this evaluation correct.".**

A: The conclusion is incorrect because running a test only once is not enough to guarantee a program is bug-free. A systematic approach requires testing multiple times (such as 5 times for rock, 5 for paper, and 5 for scissors) and using unexpected data to ensure the code always reacts correctly.

**Q2: Design a 3-point success criteria checklist to evaluate a digital greeting card created in Scratch.**

**A:**

1. The backdrops and costumes change in a logical order.
2. Text speech bubbles stay on the screen long enough for the user to read comfortably.
3. Sounds and visual effects play smoothly without cutting off abruptly.