

TOPIC 1.4: PLANNING PROGRAMS

Comprehensive Classroom Notes & Pedagogical Scaffold | Grade 6 ICT

Subject:	Information Communication Technology	Unit:	1. Computational Thinking & Programming
Level:	Grade 6 / Level 6	Date:	31 Oct 2026

1. Curriculum Focus & Prerequisites

SESSION LEARNING OBJECTIVES

- Understand how a clear plan can significantly improve a program.
- Describe how prototypes are useful when designing programs and collecting early user feedback.
- Create a structured project plan for a programming project using decomposition.
- Develop an effective interface with helpful prompts for a program..

What You Already Know (Previous Learning):

Variables store data: Variables are named containers used to store data in a program. The value inside a variable can change while the program runs.

The 'ask' and 'answer' blocks: In Scratch, the 'ask [] and wait' block prompts the user for text or number input and automatically stores the response in the built-in 'answer' variable.

Conditional Selection: Selection statements (IF THEN and IF THEN ELSE) execute specific instruction blocks depending on whether a condition evaluates to True or False.

Q1: What is decomposition in computer programming?

Answer: Decomposition is the process of breaking down a large, complex problem into smaller, manageable parts. This allows programmers to focus on and solve each part one by one rather than becoming overwhelmed.

Q2: Why is it important to plan a program before writing code?

Answer: Planning helps a programmer clearly outline what the program will do and how the screen interface will look for the user. Programs that are planned in advance are much more likely to work successfully and contain fewer logical bugs.

Q3: What is the user interface of a program?

Answer: The user interface is everything the user sees on the screen when interacting with a App or program, including sprites, backdrops, buttons, and text messages. It is vital because it clearly shows the user what actions they need to take.

Q4: What is a prompt and why is it used ?

Answer: A prompt is an on-screen message or question indicating that the program is paused and waiting for user input. It is used to ask for essential information from the user, such as user's name or their choice in a game.

Q5: What is a prototype and why do developers build them?

Answer: A prototype is a basic, working version of a program created for testing and gathering user feedback. It allows developers to identify mistakes and make quick improvements before spending time on the final product.

Q6: What does prioritising mean in project planning?

Answer: Prioritising means deciding which tasks or parts of a program to work on first. Programmers typically prioritise the most complex logic such as calculating game outcomes before adding visual decorations.

Q7: What does the term 'random' mean in programming?

Answer: Random means that the outcome of an event cannot be predicted in advance. It is used in games to provide fair, unpredictable events, like a computer choosing rock, paper, or scissors.

Q8: How does Scratch store a user's typed response when asking a question?

Answer: When Scratch runs the 'ask [] and wait' block, it pauses until the user types an answer. The typed text is automatically saved into the built-in 'answer' variable, which can then be transferred into custom variables.

Q9: What is the difference between front-end and back-end developers?

Answer: Front-end developers focus on designing the visual interface and layout that users see on screen. Back-end developers write the process steps which include sequence, selection and iteration logic, that work behind the scenes.

Q10: How many total outcome combinations exist in a two-player game of Rock, Paper, Scissors?

Answer: There are 9 total outcome combinations because each player has 3 choices ($3 \times 3 = 9$). The program uses conditional selection statements to evaluate whether the result is a win, loss, or draw.

Second Order Thinking Skills Questions**SO-Q1 [Evaluating User Experience]: Interface Design & Prompts**

Evaluate the difference between a game interface that immediately starts without instructions versus one that includes a welcome screen, rule summary, and an explicit prompt before asking for user input. Why is the second design superior?

Detailed Answer / Analysis:

An interface lacking prompts leaves the user confused about when to act, which keys to press, and what input formats are valid (e.g., typing 'rock' vs 'R'). A structured interface with welcoming instructions and explicit prompts sets clear expectations, prevents invalid input errors, and provides a smooth, accessible user experience.

SO-Q2 [Comparative Analysis]: Detailed Planning vs. Adaptive Planning

Some programmers write strict, highly detailed plans while others use flexible outlines and adapt their ideas while coding. Compare both approaches and explain why neither method is 'wrong'.

Detailed Answer / Analysis:

Detailed planning prevents logical mistakes, and gives a clear roadmap. Adaptive planning encourages creative exploration and spontaneous feature ideas. Both approaches are valid because real-world developers frequently start with a structured plan but adjust it as prototypes reveal new opportunities.

Key Vocabulary

Key Term	Textbook Definition & Classroom Context
Decomposition	Breaking down a complex problem into smaller parts that can be worked on one by one.
Interface	What the user sees on screen when they use a program (sprites, backdrops, buttons, text).
Prompt	An on-screen message or question showing that a program is waiting for user input.
Prototype	A basic, early version of something created for users to test and give feedback on.
Prioritising	Deciding which part of a program to build first, usually tackling the most complex logic first.
Random	An outcome that cannot be predicted in advance (used for fairness and unpredictability in games).