|  |  |
| --- | --- |
| Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Date: \_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

Lesson 1.1 Algorithms and Agile Development

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |  | 1 |  | 2 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 3 |  |  |  |  |  |  | 4 |  |  |  |  |  |  |  |  |  |  | 5 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | 6 |  | 7 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 8 |  |  |  |  |  |  |  |  |  |  |  |  | 9 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 10 |  |  |  |  |  |  |  |  | 11 |  |  |  |  |  |  |  |  |  |  | 12 |  |  |  |
|  |  |  |  |  |  |  |  | 13 |  |  |  |  |  |  |  |  |  | 14 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  | 15 |  |  |  |  |  |  |  |  |  |  |  |  |  | 16 |  |  |  |  |  |  |  |  |  |  | 17 |  |
|  | 18 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 19 |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  | 20 |  | 21 |  | 22 |  |  |  |  |  |  |  |  | 23 |  |  |  |  |  |  |  |
|  |  |  |  |  | 24 |  |  |  |  |  |  |  |  |  | 25 |  | 26 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 27 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  | 28 |  |  |  |  |  |  |  |  |  |  |  |  | 29 |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  | 30 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 31 |  |  | 32 |  |  |  | 33 |  |  |  | 34 |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  | 35 |  |  |  |  |  |  |  |  |  |  |  |  | 36 |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  | 37 |  |  |  |  |  |  | 38 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  | 39 |  |  |  |  |  | 40 |  |  |  |  |  |  | 41 |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | 42 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  | 43 |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |

|  |  |
| --- | --- |
| **Across****3.** Method A set of instructions grouped together to do something to or with an**4.** Algorithm A step-by-step procedure, like a recipe, but often involving**7.** Integrated Development Environment (IDE) Provides a developer with a way to create a program, run the program, and debug the program all within one**8.** Syntax Precise rules defining how the letters, words, and punctuation of a programming language are required to be used**13.** One-way Flag Variable Stores whether a condition has been true yet; reset before iteration and possibly raised during**14.** Input Data that is used by a**24.** Waterfall Design A method of software development that emphasizes completing each stage of the design process before beginning the next**26.** Encapsulation Keeping details (like data and procedures) together in one part of a program so that programmers working on other parts of the program don’t need to know about**28.** Conditional The Boolean expression in an if-structure that is evaluated to determine which branch of code to**29.** Walker Variable Stores one item from a list at a time during**31.** State Diagram Shows the ways that a program could progress from each state to other**34.** Memory The parts of computer that can store data or instruction, including volatile memory (lost when the computer is turned off) and non-volatile memory (retained when the computer is turned**35.** Script A set of instructions. In Scratch a script is a single stack of blocks belonging to one sprite or to the**36.** Aggregator Variable Stores a list of values built up over time and remembers separate individual**39.** Sprint Task List A list of small specific tasks to be completed during the current sprint in the scrum method of software**41.** Scrum The most common framework for Agile**42.** Turing Test One method of determining the strength of artificial intelligence, in which a human tries to decide if the intelligence at the other end of a text chat is**43.** Debugging The process of figuring out why code doesn’t behave as expected and eliminating bugs to make it work as | **Down****1.** Central Processing Unit (CPU, or core) The part of the computer that executes one instruction at a**2.** Boolean Expression Evaluates to either true or false; used in the conditional of an**5.** State Data that describes everything about how a program exists at a given time; a snapshot of a**6.** Event Handler A piece of code that responds to an**9.** Variable Roles Describe why a variable is being used in a**10.** Event A “trigger” from the user or from a program that causes a specific part of the program to**11.** Pseudocode An outline of the basic ideas behind how algorithms will**12.** Meta-information Information about a program or data file, including date created and author**15.** Sprite A graphics object that can be moved on top of or behind a**16.** Execute To run a program or a single**17.** Output The effect that a program has on its**18.** Iteration Repeating a step-by-step procedure several times; also used to refer to single execution of that repetitive**19.** Code Instructions in a**20.** Modular Code A group of instructive code that exists as an independent unit, which is easy to use with other code as**21.** Agile Design A method of software development that emphasizes frequent client input to produce better end**22.** If-structure Evaluates a Boolean expression and executes some code if the expression is true (and possibly other code if it is**23.** Most Recent Variable Stores user input or stores information about a program’s state that changes**25.** Product Backlog A prioritized list of user stories, showing both short- and long-term goals of a software development**27.** User Stories Plain-language description of a user’s need for the software under development, usually phrased as “(who) wants to (do what within the software) so that they can (do what in real**30.** Bug Part of a program that causes an error or undesired**32.** Accumulator Variable Maintains a running**33.** Stepper Variable A variable that counts in an arithmetic sequence, usually counting by ones starting at 0 or**37.** Method Call An instruction to use a**38.** Best-so-far Variable Keeps track of a record best or worst and updates when a new record best or worst has been**40.** Fixed Variable Stores a value that will not change over the course of the entire |