Programming Learning Rubric (PLR) for Elementary School Students

Waseda University Global Software Engineering Laboratory

Category	Item	Stage 4	Stage 3	Stage 2	Stage 1
Attitude	Positivity	Learner voluntarily learns and challenges oneself without being instructed.	Learner spontaneously learns without being instructed.	Learner can tackle learning if instructed, but not voluntarily.	Learner does not tackle learning even when instructed.
	Interest	Self-evaluation indicates learner is greatly interested in programming and seems to have fun during programming activities.	Self-evaluation indicates learner is greatly interested in programming.	Self-evaluation indicates learner is interested in programming.	Self-evaluation indicates learner is not interested in programming.
	Toughness	Learner does not quit even if challenges arise while learning, investigates clues by oneself, and tries new methods.	Learner does not give up even if there is difficulty while learning and tries different solutions.	Learner tends to work on learning but give easily.	Learner quits too soon to start learning.
Programming Concepts	Sequence	Learner understands sequential execution, reads a program from the beginning to the end, writes a program of sequential execution, and thinks of a program to be executed sequentially by oneself.	Learner understands sequential execution, reads a program in order from the beginning to the end, and writes a program of sequential execution.	Learner can understand sequential and read a program in order from beginning to end.	Learner does not understand sequential.
	Loop	Learner understands loop, read loop programs, find loop by oneself, and incorporates loop into programs.	Learner understands loop, reads repeated programs and writes loop programs.	Learner can understand and read loop programs.	Learner does not understand loop.
	Conditional	Learner understands conditional branch, reads the program of the conditional branch, and incorporates conditional branch into programs.	Learner understands repetition, reads repeated programs, and writes repeated programs.	Learner can understand conditional branch and read conditional branching programs.	Learner does not understand conditional branching.
Construction of Computer	Construction of Computer	Learner understands the principles of computers and external devices, and connects them to solve problems when they arise.	Learner understands the basic principles of computers and the connection with the roles of external devices (mouse, printer, network equipment).	Learner can understand the basic principles of the computers (input-output processor, sensor, storage).	Learner does not understand the basic principles of computers.
Designing Programs	Subdivision	Learner correctly divides large problems into smaller ones that cannot be broken up any further.	Learner correctly divides large problems into multiple small problems.	Learner can divide large problems into two or more smaller problems.	Learner cannot divide the problem.
	Analysis	Learner considers the cause and effect relationship of events, which leads to abstract rules and principles from specific relationships, and writes upright thread.	Learner considers the relationship of cause and outcome of the event, and is aware of that specific relationship can be exported to stand thread.	Learner can notice that there is a relationship between the cause and effect of events.	Learner cannot notice the relationship that is in the event.
	Extraction	Learner extracts in accordance with the purpose with the minimum of operations.	Learner extracts in accordance with the purpose and takes the necessary action by oneself.	Learner can extract based on the purpose and the necessary action from the choices.	Learner cannot extract.
	Construction and Functionalization	Learner realizes the object considering the optimal combination of plurality of procedures and creates procedures with universality and reproducibility.	For learner to achieve the object, a plurality of procedures and sequential processing are iteratively combined by utilizing processes such conditional branch.	Learner can notice that there is a plurality of steps and intentionally rearrange to suit a given procedure on purpose.	Learner cannot build up an operation in combination with the procedure.
	Generalization	Learner uses the plurality of past resolved problems to find similarities and relationships, general rules, and principles in common, and uses these to resolve to other problems.	Learner compares the eye in front problem to resolve issues and solves problems by applying similarities and relationships.	Learner can use resolved events, such as Kizukeru, to realize that there are similarities and relationships between events.	Learner cannot find an association between events.
	Design Document	Learner plans and creates a design document with reference to FIG, sentences ideas, and procedures (story map, etc.) with an easy-to-understand format.	Learner uses ideas and procedures to plan and creates a design document with figures and texts (story map, etc.)	Learner can express ideas and procedures in the picture.	Learner cannot express ideas and procedures.
	Expression	Learner creates new original expressions making full use of various methods.	Learner imitates expressions of existing works and incorporates them into one's work.	Learner can create works with basic expression techniques by oneself.	Learner cannot create by oneself.
	Creativity	Learner stands from a global perspective and designs to achieve a purpose by associating the nature of the program, the position of the user, etc.	Learner designs things to achieve the goals according to one's sensitivity, considering the user's position.	Learner can design things to achieve the goals according to one's sensitivity.	Learner cannot design things to achieve the goals.
Creating Programs	Use of Programming Concepts	Learner creates a program, a solution, or creative expression problems, including sequential execution, events, loops, conditional branching, parallelism, variable.	Learner creates a program, a solution, or creative expression problems, including sequential execution, events, loop.	Learner can create a program or a solution or creative expression problems, including sequential execution, simple loop.	Learner cannot program.
	Logical Thinking	Learner uses the logic of such a set-proposition, Boolean (e.g., condition setting at the time of the branch).	Learner understands and uses a logical structure such as repetition and conditional branching.	Learner can specifically visualize operations (e.g., it is possible to combine the operation in the correct order).	Learner cannot program.
	Use of Software	Learner uses programming software to create a program that operates as intended.	Learner understands how to use the programming software to create some kind of program.	Learner can understand how to use some of the programming software.	Learner cannot use software.
	Programming Language	Learner programs using both visual and text-based languages.	Learner programs using only a text-based language.	Learner can program using only a visual language.	Learner cannot program.
	Expressing Data	Learner accurately represents all of the data using a number or other symbols (e.g., represented by raising and lowering of the Yes / No thumb, representing the color by numbers, representing the direction by an arrow).	Learner accurately represents some of the data using a number or other symbols.	Learner can see data represented using a number or other symbols.	Learner cannot express data.
	Use of Formula	Learner uses arithmetic operators and comparison operators.	Learner changes the value stored in the variable with an operator.	Learner can understand the program using the operator.	Learner cannot learn.
Reading, Editing, Evaluating Programs	Read	Learner reads the existing program and explains the contents.	Learner reads the existing program.	Learner can read some of the program.	Learner cannot read a program.
	Edit	Learner changes the existing program and the changes are reflected in one's program.	Learner reflects an existing program into one's program.	Learner can modify an existing program.	Learner cannot edit a program.
	Evaluate	Learner ensures that the program runs as intended and debugs.	Learner works the program as intended.	Learner, with the teacher, can verify the operation of the program.	Learner cannot evaluate a program.
Self-regulation	Plan	Learner plans proactively to achieve the intended purpose under the appropriate conditions.	Learner proactively plans to achieve the objective.	Learner, with the teacher, can make a plan to achieve an objective.	Learner cannot plan.
	Safety Considerations	Learner makes rules to work safely by oneself.	Learner makes rules to work safely by oneself.	Learner can be observed trying to work safely.	Learner does not consider safety.
	Announcing Own Idea	Learner presents in a way that is convincing of best thoughts design.	Learner emphasizes the best of ideas.	Learner can announce one's ideas and thoughts.	Learner does not work with others.
	Understanding Other's Idea	Learner understands the announcements of others and improves one's work in reference.	Learner understands what others want and evaluates the quality of the idea.	Learner can understand what others want to do.	Learner cannot understand others' ideas.
	Cooperation in Programming	Learner works with others and significantly contributes to the program and the team.	Learner works with others and helps contribute to the program and team.	Learner can program in partnership with others team members.	Learner cannot program with a team.

Cooperation with Others	Contribution in Group Work	When a learner works in a group, all three of the following activities are satisfied: 1. Learner uses one's abilities to achieve objectives and is independent. 2. Learner thinks about solutions to solve problems raised in the middle of the activities. 3. Learner considers the opinions of others analytically and compiles ideas of the group to achieve the purpose.	When a learner works in a group, two of the three following activities are satisfied: 1. Learner uses one's abilities to achieve objectives and is independent. 2. Learner thinks about solutions to solve problems raised in the middle of the activities. 3. Learner considers the opinions of others analytically and compiles ideas of the group to achieve the purpose.	When a learner works in a group, one of the three following activities are satisfied: 1. Learner uses one's abilities to achieve objectives and is independent. 2. Learner thinks about solutions to solve problems raised in the middle of the activities. 3. Learner considers the opinions of others analytically and compiles ideas of the group to achieve the purpose.	Learner does not understand.