



CREATE 2D EXPLAINED

**KEEP THE END IN MIND
WHEN DEVELOPING
YOUR APP**

**ABSTRACTION AND MANAGING THE
COMPLEXITY OF THE PROGRAM**

ROW 7 – YES OR NO WILL BE PART OF 2D TASK

| Reporting Category | Task | Scoring Criteria | Decision Rules | Scoring Notes |
|-------------------------------|-----------------------------|---|--|--|
| Row 7 Applying Abstraction | CODE SEGMENT IN RESPONSE 2D | <ul style="list-style-type: none"> Selected code segment is a student-developed abstraction. | <ul style="list-style-type: none"> Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; or the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). | <ul style="list-style-type: none"> The following are examples of abstractions (EK 5.3.1): <ul style="list-style-type: none"> Procedures Parameters Lists Application program interfaces (APIs) Libraries Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1) |

This must be a **student-developed abstraction** and must have a RECTANGLE around the entire abstraction. Do not choose any automated Code.org coding like an “onEvent.” It must be your own creation. When writing about it, you should include the type of algorithm this uses (is it procedural, does it have parameters, it is a list, etc.,) the name of your algorithm and what it is supposed to do when working properly

<https://apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf>

ROW 7 – YES OR NO WILL BE PART OF 2D TASK

| Reporting Category | Task | Scoring Criteria | Decision Rules | Scoring Notes |
|-------------------------------|-----------------------------|---|--|--|
| Row 7 Applying Abstraction | CODE SEGMENT IN RESPONSE 2D | <ul style="list-style-type: none"> Selected code segment is a student-developed abstraction. | <ul style="list-style-type: none"> Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; or the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). | <ul style="list-style-type: none"> The following are examples of abstractions (EK 5.3.1): <ul style="list-style-type: none"> o Procedures o Parameters o Lists o Application program interfaces (APIs) o Libraries Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1) |

When use the rectangle to identify your abstraction, make sure you also have where you called the function in your program. You can have two rectangles, **one that shows the function** instructions, procedures, or parameters and **one that shows where you called the function**

<https://apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf>

Mathisd3@duvalschools.org

64

ROW 7 – YES OR NO WILL BE PART OF 2D TASK

| Reporting Category | Task | Scoring Criteria | Decision Rules | Scoring Notes |
|-------------------------------|-----------------------------|---|--|--|
| Row 7 Applying Abstraction | CODE SEGMENT IN RESPONSE 2D | <ul style="list-style-type: none"> Selected code segment is a student-developed abstraction. | <ul style="list-style-type: none"> Responses that use existing abstractions to create a new abstraction, such as creating a list to represent a collection (e.g., a classroom, an inventory), would earn this point. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the response is an existing abstraction such as variables, existing control structures, event handlers, APIs; or the code segment consisting of the abstraction is not included in the written responses section or is not explicitly identified in the program code section; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). | <ul style="list-style-type: none"> The following are examples of abstractions (EK 5.3.1): <ul style="list-style-type: none"> Procedures Parameters Lists Application program interfaces (APIs) Libraries Lists and other collections can be treated as abstract data types (ADTs) in developing programs. (EK 5.5.1) |

The first sentence for this paragraph should be what type of abstraction it is (procedure, parameter, lists..)
The second sentence should describe how it works.
Using your snipping tool, capture your abstraction from the PDF file you created and insert it under these sentences.

<https://apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf>

Mathisd3@duvalschools.org

65

ROW 8 – YES OR NO WILL BE PART OF 2D TASK

| Reporting Category | Task | Scoring Criteria | Decision Rules | Scoring Notes |
|-------------------------------|-------------|--|--|---|
| Row 8 Applying Abstraction | RESPONSE 2D | <ul style="list-style-type: none"> Explains how the selected abstraction manages the complexity of the program. | <ul style="list-style-type: none"> Responses should not be penalized for explanations of abstractions that are not developed by the student. <p>Do NOT award a point if any one of the following is true:</p> <ul style="list-style-type: none"> the explanation does not apply to the selected abstraction; or the abstraction is not explicitly identified (i.e., the entire program is selected as an abstraction, without explicitly identifying the code segment containing the abstraction). | <ul style="list-style-type: none"> See Row 7 definitions and curriculum framework alignment. |

The words “**manages the complexity**” should be a part of your explanation.

Your sentence should state AND DESCRIBE how using this abstraction, makes your program easier to follow in the event of growing your application, or how much less coding you would have to do, or any other way that this refines your code.

Example: The iterative function (global score) manages the complexity of the program because it is constantly calling and checking the score from (some other) function as the app is used. It saves steps in coding and it

<https://apcentral.collegeboard.org/pdf/ap-csp-create-performance-task-scoring-guidelines-2019.pdf>

END
VIDEO NOTES #5

NOTES WILL BE AVAILABLE IN TEAMS