CREATE 2C EXPLAINED

KEEP THE END IN MIND WHEN DEVELOPING YOUR APP

THE MOST IMPORTANT PART OF THE TASK THAT PROVES YOU HAVE LEARNED SOMETHING THIS YEAR

2C INCLUDES ROWS 4, 5, AND 6 ON THE GRADING RUBRIC

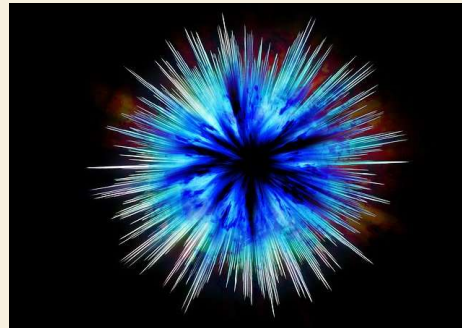| Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 4 Applying Algorithms	CODE SEGMENT IN RESPONSE 2C	• Selected code segment implements an algorithm.	Do NOT award a point if any one of the following is true: • the algorithm consists of a single instruction; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. (EU 4.1) • Algorithms make use of sequencing, selection or iteration. (EK 4.1.1A)
Row 5 Applying Algorithms	RESPONSE 2C	• Selected code segment implements an algorithm that uses mathematical or logical concepts. AND • Explains how the selected algorithm functions AND • Describes what the selected algorithm does in relation to the overall purpose of the program.	• The algorithm being described can utilize existing language functionality or library calls. • Response earns the point even if the algorithm was not newly developed. (I.e., a student's reimplementation of the algorithm to find the minimum value.) • Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms. Do NOT award a point if any one of the following is true: • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • the selected algorithm does not include mathematical or logical concepts; or • the response only describes what the selected algorithm does without explaining how it does it; or • the response does not explicitly address the program's purpose; or • the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• See Row 4 definitions and curriculum framework alignment. • Mathematical concepts include mathematical expressions using arithmetic operators and mathematical functions. (EK 5.5.1.D) • Logical concepts include Boolean algebra and compound expressions. (EK 5.5.1E and 5.5.1F) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C)
Row 6 Applying Algorithms	RESPONSE 2C	• Selected code segment implements an algorithm that includes at least two or more algorithms. AND • At least one of the included algorithms uses mathematical or logical concepts. AND • Explains how one of the included algorithms functions independently.	• Responses are still eligible to earn this row, even if they do not earn row 5. • The included algorithms can be sub-parts of the algorithm in row 5. Do NOT award a point if any one of the following is true: • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• See Row 4 and Row 5 definitions and curriculum framework alignment.

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

OVERVIEW OF 2C

"Capture and paste a program code segment that implements an algorithm (marked with an OVAL) and that is fundamental for your program to achieve its intended purpose. This code segment MUST be an algorithm YOU developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently of each other AS WELL AS in combination with others to form a new algorithm that helps to achieve the intended purpose of the program. (*Must not exceed 200 words.*)"

https://apcentral.collegeboard.org/pdf/ap-csp-student-task-directions.pdf page 11

Part 2C centers around you designing a program that uses three inter-related algorithms within your program. Knowing this, your planning of your program REALLY comes into play. Figure out the types of functions you understand best (logic, mathematical) and figure out a way to incorporate them into your project in order to solve your problem.

BREAKING DOWN 2C
STARTING WITH
THE BIG BANG FOR YOUR BUCKS

ROW 6 – YES OR NO
WILL BE PART OF 2C TASK

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 6 Applying Algorithms	RESPONSE 2C	• Selected code segment implements an algorithm that includes at least two or more algorithms. **AND** • At least one of the included algorithms uses mathematical or logical concepts. **AND** • Explains how one of the included algorithms functions independently.	• Responses are still eligible to earn this row, even if they do not earn row 5. • The included algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• See Row 4 and Row 5 definitions and curriculum framework alignment.

If you can totally kill it on row number 6, rows 4 and 5 will take care of themselves. This is the row many students do not earn the point because it is not complete. If you master this, Rows 4 and 5 will not be difficult to do.

UNDERSTANDING THE PROMPT

- "Capture and paste a program code segment that implements an algorithm (marked with an OVAL) and that is fundamental for your program to achieve its intended purpose. This code segment MUST be an algorithm YOU developed individually on your own, must include two or more algorithms, and must integrate mathematical and/or logical concepts. Describe how each algorithm within your selected algorithm functions independently of each other AS WELL AS in combination with others to form a new algorithm that helps to achieve the intended purpose of the program. (*Must not exceed 200 words.*)"

FUNDAMENTAL – your "main algorithm" or "selected algorithm," or "algorithm one" when you go to comment in your code and write about it in your write up.

This would be your primary algorithm that two or more other algorithms feed into

This would include Boolean (and, or, not, greater than, less than, etc.) or your if/then, if/else statements and anything that would involve making calculations of some sort.

"…Independently .. combination …" Think in terms of PARENT and child functions. The child functions can work on their own but when there is a big need for the child functions to work together, they need to be housed within the Parent function

GRUMPY CAT EXAMPLE

HTTPS://YOUTU.BE/PZTREGNQAK8

EXAMPLE OF PARENT AND CHILD ALGORITHMS

Logic

```
// Main algorithm, handles all purchasing behabior
function purchase(item) {
  if((item == "yarn") && (yarnCost <= score)){
    score = score - yarnCost;
    yarnCount = yarnCount + 1;
    multiplier = multiplier + 2;
  } else if ((item == "mouse") && (mouseCost <= score)){
    score = score - mouseCost;
    mouseCount = mouseCount + 1;
    multiplier = multiplier + 10;
  } else if ((item == "lasagna") && (lasagnaCost <= score)){
    score = score - lasagnaCost;
    lasagnaCount = lasagnaCount + 1;
    multiplier = multiplier * 2;
  }
  setPrices();
  setVisibleItems();
  updateScreen();
}
```

```
// Child algorithm 1
function setPrices(){
  // Update Cost and Count of All Store Items
  yarnCost = 20 + (yarnCount * 20);
  mouseCost = 100 + (mouseCount * 50);
  lasagnaCost = 500 + (lasagnaCount * 1000);

  //Update Yarn Text
  setProperty("buyYarn_button", "text",("BUY: " + yarnCost + " pts"));
  setProperty("yarnCount", "text", "You already own: " + yarnCount);

  //Update Mouse Text
  setProperty("buyMouse_button","text",("BUY: " + mouseCost + " pts"));
  setProperty("mouseCount", "text", "You already own: " + yarnCount);

  //Update Lasagna Text
  setProperty("buyLasagna_button","text",("BUY: " + lasagnaCost + " pts"));
  setProperty("lasagnaCount", "text", "You already own: " + lasagnaCount);
}
```

Math

```
// Child algorithm 2 and my abstraction
function setVisibleItems(){

  setProperty("buyYarn_button","background-color","crimson");
  //Change yarn button color if you can't afford it
  if(score < yarnCost){
    setProperty("buyYarn_button","background-color","gray");
  }

  setProperty("buyMouse_button","background-color","crimson");
  //Change mouse button color if you can't afford it
  if(score < mouseCost){
    setProperty("buyMouse_button","background-color","gray");
  }

  setProperty("buyLasagna_button","background-color","crimson");
  //Change lasagna button color if you can't afford it
  if(score < lasagnaCost){
    setProperty("buyLasagna_button","background-color","gray");
  }
}
```

Main algorithm "purchase" - when he goes to the pet store
Child 1 algorithm "set prices" – when he clicks yarn, mouse or Lasagna
Child 2 algorithm "setVisibleItems" – the resulting change due to ability to buy or not which is based on the purchase data.

ROW 4 – YES OR NO (PART OF 2C RESPONSE)

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 4 Applying Algorithms	CODE SEGMENT IN RESPONSE 2C	• Selected code segment implements an algorithm.	**Do NOT award a point if any one of the following is true:** • the algorithm consists of a single instruction; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• Algorithms are precise sequences of instructions for processes that can be executed by a computer and are implemented using programming languages. (EU 4.1) • Algorithms make use of sequencing, selection or iteration. (EK 4.1.1A)

This could be a sequencing algorithm with many steps that you can use over and over by just calling it, or the more preferred would be something with math and/or logic in it.

It must accomplish a purpose. The function is accomplishing the purpose.

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

WHEN EXPLAINING YOUR FUNCTION

- onEvent is not a FUNCTION you can use in your write up
  - You didn't develop the code for the "onEvent"
  - Your code doesn't start until and after the "onEvent" Choose something that comes AFTER the user has started your app.

ROW 5 – YES OR NO (PART OF 2C RESPONSE)

NOTE: If you accomplish row 5, you will get the point for rows 4 and 5.

The 3rd bullet is usually where students lose the point because they don't associate the function with the purpose.

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 5 Applying Algorithms	RESPONSE 2C	• Selected code segment implements an algorithm that uses mathematical or logical concepts. **AND** • Explains how the selected algorithm functions **AND** • Describes what the selected algorithm does in relation to the overall purpose of the program. All three must be fulfilled.	• The algorithm being described can utilize existing language functionality or library calls. • Response earns the point even if the algorithm was not newly developed. (I.e., a student's reimplementation of the algorithm to find the minimum value.) • Mathematical and logical concepts can be a part of the selected algorithm or part of either of the included algorithms. Do NOT award a point if any one of the following is true: • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • the selected algorithm does not include mathematical or logical concepts; • the response only describes what the selected algorithm does without explaining how it does it; or • the response does not explicitly address the program's purpose; or • the code segment consisting of the selected algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• See Row 4 definitions and curriculum framework alignment. • Mathematical concepts include mathematical expressions using arithmetic operators and mathematical functions. (EK 5.5.1.D) • Logical concepts include Boolean algebra and compound expressions. (EK 5.5.1E and 5.5.1F) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C) • Iteration is the repetition of part of an algorithm until a condition is met or for a specified number of times. (EK 4.1.1D) • Selection uses a Boolean condition to determine which of two parts of an algorithm is used. (EK 4.1.1C)

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

Mathisd3@duvalschools.org

ROW 6 – YES OR NO
WILL BE PART OF 2C TASK

Reporting Category	Task	Scoring Criteria	Decision Rules	Scoring Notes
Row 6 Applying Algorithms	RESPONSE 2C	• Selected code segment implements an algorithm that includes at least two or more algorithms. **AND** • At least one of the included algorithms uses mathematical or logical concepts. **AND** • Explains how one of the included algorithms functions independently. Must have all three.	• Responses are still eligible to earn this row, even if they do not earn row 5. • The included algorithms can be sub-parts of the algorithm in row 5. **Do NOT award a point if any one of the following is true:** • the selected algorithm consists of a single instruction; or • the selected algorithm consists solely of library calls to existing language functionality; or • neither of the included algorithms nor the selected algorithm that includes two or more algorithms uses mathematical or logical concepts; or • the code segment consisting of the algorithm is not included in the written responses section or is not explicitly identified in the program code section; or • the algorithm is not explicitly identified (i.e., the entire program is selected as an algorithm, without explicitly identifying the code segment containing the algorithm).	• See Row 4 and Row 5 definitions and curriculum framework alignment.

Pay attention to the things that will not work and will offer no point

ROW 6 – BE SPECIFIC

When you describe each of the three algorithms, you need to state

- The name of your algorithm
- The type of algorithm (Selection, Iteration or Sequence)
- Explain what it does
- Explain how it helps the overall program.

Sequence algorithm: tells the computer step by step how it should run. Think of it like a RECIPE
Example Making a Pizza:
base→ tomato sauce→ cheese → bake

Selection algorithm: this is a decision-making algorithm where there is a choice to make before proceeding
Example Making Pizza:
base→ tomato sauce→ topping? (if yes, add parsley, if not, skip) →cheese → bake

Iteration algorithm: it is a set of looping instructions as the program moves through
Example Making Pizza:
base→ tomato sauce→ topping? (if yes, add parsley and go back to topping, if not, skip) →cheese → bake

EXAMPLE OF PARENT AND CHILD ALGORITHMS

Logic

```
// Main algorithm, handles all purchasing behabior
function purchase(item) {
  if((item == "yarn") && (yarnCost <= score)){
    score = score - yarnCost;
    yarnCount = yarnCount + 1;
    multiplier = multiplier + 2;
  } else if ((item == "mouse") && (mouseCost <= score)){
    score = score - mouseCost;
    mouseCount = mouseCount + 1;
    multiplier = multiplier + 10;
  } else if ((item == "lasagna") && (lasagnaCost <= score)){
    score = score - lasagnaCost;
    lasagnaCount = lasagnaCount + 1;
    multiplier = multiplier * 2;
  }
  setPrices();
  setVisibleItems();
  updateScreen();
}
```

Math

```
// Child algorithm 1
function setPrices(){
  // Update Cost and Count of All Store Items
  yarnCost = 20 + (yarnCount * 20);
  mouseCost = 100 + (mouseCount * 50);
  lasagnaCost = 500 + (lasagnaCount * 1000);

  //Update Yarn Text
  setProperty("buyYarn_button", "text",("BUY: " + yarnCost + " pts"));
  setProperty("yarnCount", "text", "You already own: " + yarnCount);

  //Update Mouse Text
  setProperty("buyMouse_button","text",("BUY: " + mouseCost + " pts"));
  setProperty("mouseCount", "text", "You already own: " + yarnCount);

  //Update Lasagna Text
  setProperty("buyLasagna_button","text",("BUY: " + lasagnaCost + " pts"));
  setProperty("lasagnaCount", "text", "You already own: " + lasagnaCount);
}
```

```
// Child algorithm 2 and my abstraction
function setVisibleItems(){

  setProperty("buyYarn_button","background-color","crimson");
  //Change yarn button color if you can't afford it
  if(score < yarnCost){
    setProperty("buyYarn_button","background-color","gray");
  }

  setProperty("buyMouse_button","background-color","crimson");
  //Change mouse button color if you can't afford it
  if(score < mouseCost){
    setProperty("buyMouse_button","background-color","gray");
  }

  setProperty("buyLasagna_button","background-color","crimson");
  //Change lasagna button color if you can't afford it
  if(score < lasagnaCost){
    setProperty("buyLasagna_button","background-color","gray");
  }
}
```

Main algorithm "purchase" - when he goes to the pet store
Child 1 algorithm "set prices" – when he clicks yarn, mouse or Lasagna
Child 2 algorithm "setVisibleItems" – the resulting change due to ability to buy or not which is based on the purchase data.

END
VIDEO NOTES #4

NOTES WILL BE AVAILABLE IN TEAMS