Question	Answer Choices	Misconceptions Targeted

The Worker class has several child classes: `Janitor`, `Designer`, and `Coder`. `Designer` has a child class of its own - `SoundDesigner`.
 	
This code is run in a driver class:
 	
`Worker[] lanCorp = new Worker[3];`
 	
`lanCorp[0] = new Janitor("Dennis"); //line 2`
 	
`lanCorp[1] = new Designer("Jake"); //line 3`
 	
`lanCorp[2] = new Employer("Jacob"); //line 4`
 	
`lanCorp[3] = new SoundDesigner(“Edward”); //line 5`
 	
Which line above causes a compiler error?
	A) Line 2	Selecting this option might mean that the respondent did not know that one can add objects of child classes to arrays of their parent classes.

	B) Line 3	Selecting this option might mean that the respondent did not know that one can add objects of child classes to arrays of their parent classes. It might also mean that they do not think one can add objects of different child classes to the same parent class array. (Because a Janitor object has already been added to the array by the end of line 2).
	C) Line 4	This answer is correct because Employer is not a child class of Worker. As it does not extend worker, it is also NOT a Worker, whereas Janitor and Designer are Workers.
	D) Line 5	The correspondent selecting this option likely means that they do not believe grandchild classes extend their grandparents AS WELL as their parents. SoundDesigner is a Worker the same way it is an Object.

Question	Answer Choices	Misconceptions Targeted
The `workDay()` method is one that exists in a resource class. An object array of the aforementioned resource class has been initialised in the driver class. `workDay()` includes a print statement in its code. `Janitor` is a child	E) There will be an out of bounds exception.	The respondent would have likely seen the error in the if statement. This code would have ended up in an out of bounds exception error if it had been implemented in an iterative implementation.

class of `Worker`.
 	
This code is run in a driver class:
 	
`public static void main(String[] args) {`
`  Worker[] lanCorp = new Worker[10];`
`  lanCorp[0] = new Janitor("Dennis");`
`  firmOutput(lanCorp);`
 	
`//the rest of main is not shown.`
`}`
 	
`private static int count = 0;`
`public static void firmOutput(Worker[] lanCorp){`
`  if(lanCorp.length < count){`
`  lanCorp[count].workDay();`
`  count++;`
`  firmOutput(lanCorp);`
`  }`
`  else{`
`  count = 0;`
`  }`
`}`
 	
What is the output of this program?
		However, as this is a recursive implementation, the program gets to the same solution with different methods. The respondent would have likely confounded the two types of solutions.

	F) Nothing will be outputted.	This answer is correct because the if statement does not pass so count is set to 0 with nothing printed and the program ends.
	G) The output of workDay() for all Worker objects in lanCorp will be printed.	The respondent would have done one of two things: 1. They wouldn't have noticed the error in the if statement and would have assumed that the code inside of the if statement would have been run, and the recursive firmOutput() method would have been called numerous times. 2. They assumed that because count is static, it cannot be set to 0, so the program must have had to run the if statement because there was no alternative. a. I know it sounds dumb but I did think this was true at one point.
	H) The code in the program will have caused a compiler error.	Selecting this option might mean that the respondent did not know that one can add objects of child classes to arrays of their parent classes and assumed that the fact that a Janitor object was being added to a Worker array would have caused a compiler error when it wouldn't have.