



Certified Ruby Programmer (CRBP)

Ten Sample Questions

These ten questions are written at the difficulty of the live exam. Six carry a figure, a data table, a configuration or a code listing; four are plain scenario questions, as on the live paper. Every one is a scenario a working Ruby programmer would meet, and none of them appears in the live question bank.

The live exam is multistage adaptive: how you score on one stage selects the difficulty of the next, so no two candidates sit the same paper. It is a performance-based exam: each form can carry up to 5 to 12 performance-based questions alongside the multiple-choice questions, which means you build, sequence or complete something rather than only choosing from a list. Answers and full explanations follow each question here; the live exam does not disclose item-level answers.

Question 1

Domain 1: Ruby Fundamentals and Syntax

CONFIG

Sarah Miller runs this short snippet in the Ruby interpreter.

```
x = 5
y = 2
puts x / y
```

Listing 1.1 Integer division.

What does the snippet print?

- ☐ A 2.5
- ☐ B 2
- ☐ C 3
- ☐ D 2.0

Correct answer: B

Ruby divides two integers using integer division, so 5 divided by 2 gives 2 and the remainder is dropped. A float result like 2.5 would appear only if one of the operands were itself a float.

Question 2

Domain 2: Data Types and Collections

DATA

James Carter maps each collection to how you reach a stored value, with one entry missing.

Collection	How to reach a value
Array	By numeric index
Hash	(to choose)
Range	Step through values
Symbol	Reused as a label

Table 2.1 Collections and access.

How do you reach a value in a hash?

- ☐ A By its position
- ☐ B By its index
- ☐ C By its length
- ☐ D By its key

Correct answer: D

A hash stores pairs and returns a value when you give it the matching key. An array is the collection reached by a numeric index, while a length or a position does not look up a hash value.

Question 3

Domain 3: Control Flow and Methods

CONFIG

Olivia Davis defines a method and then calls it once.

```
def add(a, b)
  sum = a + b
end
puts add(3, 4)
```

Listing 3.1 Implicit return.

What does the code print?

- ☐ A nil
- ☐ B An error
- ☐ C 7
- ☐ D 34

Correct answer: C

A Ruby method returns the value of its last expression, so add returns the sum and the code prints 7. No explicit return keyword is needed for this behavior in Ruby.

Question 4

Domain 4: Object-Oriented Programming and Modules

DATA

Liam Anderson lists the Ruby method visibility levels, with one description missing.

Visibility	Who can call it
public	Any caller
private	(to choose)
protected	Same class or subclass
module method	Called on the module

Table 4.1 Method visibility.

Who can call a private method?

- ☐ A Its own object only
- ☐ B Callers with a receiver
- ☐ C Only child classes
- ☐ D Any code anywhere

Correct answer: A

A private method can be called only within its own object and never with an explicit receiver. Public methods accept any caller, so an outside object using a receiver would not reach a private method.

Question 5

Domain 5: Blocks, Procs and the Standard Library

CONFIG

Emma Wilson passes a block to a method that yields twice.

```
def twice
  yield 5
  yield 5
end
sum = 0
twice { |n| sum += n }
puts sum
```

Listing 5.1 Yielding to a block.

What does the code print?

- ☐ A 5
- ☐ B 0
- ☐ C nil
- ☐ D 10

Correct answer: D

The method yields the value 5 to the block two times, and each call adds 5 to sum, so the final total printed is 10. The yield keyword hands control to the block that was passed in.

Question 6

Domain 6: Exceptions, Testing and Best Practices

CONFIG

David Brown wraps a risky calculation in a begin and rescue block.

```
result = begin
  10 / 0
rescue ZeroDivisionError
  -1
end
puts result
```

Listing 6.1 Rescuing an error.

What does the code print?

- ☐ A 0
- ☐ B nil
- ☐ C -1
- ☐ D An error stops it

Correct answer: C

Dividing by zero raises a `ZeroDivisionError`, and the rescue clause returns `-1`, which becomes the value of the whole block. That value is stored in `result` and printed instead of crashing the program.

Question 7

Domain 1: Ruby Fundamentals and Syntax

In Ruby a name written in all capital letters, such as `MAX_SIZE`, signals a value that should not change. What kind of name is this?

- ☐ A A constant
- ☐ B A local variable
- ☐ C A private method
- ☐ D A global variable

Correct answer: A

Ruby treats a name that begins with a capital letter as a constant, and all capitals is the common style for one. Reassigning it only produces a warning, which signals the value is meant to stay fixed.

Question 8

Domain 3: Control Flow and Methods

A variable created inside a method cannot be seen or used outside that method once it returns. What kind of scope does the variable have?

- ☐ A Global scope
- ☐ B Local scope

- ☐ C Class scope
- ☐ D Shared scope

Correct answer: B

A variable defined inside a method has local scope and exists only while that method runs. Once the method returns the name is gone, unlike a global or class level name that lives longer.

Question 9 Domain 4: Object-Oriented Programming and Modules

A method calls quack on any object that responds to it, without ever checking the class of that object. Which Ruby idea does this describe?

- ☐ A Strict typing
- ☐ B Static binding
- ☐ C Deep copying
- ☐ D Duck typing

Correct answer: D

Duck typing means Ruby cares only that an object responds to the message, not what class it belongs to. If the object responds to quack it is treated as usable, which keeps the code flexible.

Question 10 Domain 5: Blocks, Procs and the Standard Library

A method such as each walks through a collection and hands each element to a block in turn. What is the general name for a method that works this way?

- ☐ A An iterator
- ☐ B A constant
- ☐ C A rescue block
- ☐ D A getter

Correct answer: A

A method that steps through a collection and passes each item to a block is called an iterator. The each method is the common example, and it relies on yielding values to the block.