

21MIS1021

RUBY ASSESSMENT 6.1

VIMAL KUMAR S

1.

CODE:

```
def divide(x, y)
  raise "Divisor cannot be zero" if y == 0
  x / y
end

begin
  result = divide(10, 0)
  puts "Result: #{result}"
rescue => e
  puts "An error occurred: #{e.message}"
end
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021>ruby 1.rb
An error occurred: Divisor cannot be zero
```

2.

CODE:

```
max_retries = 3
retry_count = 0

begin
  # Perform some operation that might raise an exception
  # ...
  # If an exception occurs, raise it explicitly using `raise`
  raise "Some error occurred"
rescue => e
```

```

if retry_count < max_retries
  retry_count += 1

  puts "An error occurred: #{e.message}. Retrying (Attempt #{retry_count})..."

  retry
else
  puts "Maximum retries exceeded. Error: #{e.message}"
end
end

```

OUTPUT:

```

C:\Users\Dell\Desktop\21MIS1021>ruby 1.rb
An error occurred: Some error occurred. Retrying (Attempt 1)...
An error occurred: Some error occurred. Retrying (Attempt 2)...
An error occurred: Some error occurred. Retrying (Attempt 3)...
Maximum retries exceeded. Error: Some error occurred

```

3.

```

def open_file(file_path)
  file = nil

  begin
    file = File.open(file_path)

    # Perform some operations with the file
    # ...

    raise "Some error occurred"
  rescue => e
    puts "An error occurred: #{e.message}"
  ensure
    file.close if file
  end
end

```

OUTPUT:

```

C:\Users\Dell\Desktop\21MIS1021>ruby 1.rb

```

4.

```
begin
  puts 'no Exception raise'
rescue
  puts 'Finally Saved!'
else
  puts 'Else block execute because of no exception raise'
# using ensure statement
ensure
  puts 'ensure block execute'
end
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021>ruby 1.rb
no Exception raise
Else block execute because of no exception raise
ensure block execute
```

5.

```
def find_in_array(array, target)
  result = catch :found do
    array.each do |element|
      throw :found, element if element == target
    end
    nil # Return nil if the target is not found in the array
  end
  return result
end

arr = [1, 4, 7, 10, 15]
puts find_in_array(arr, 10)
puts find_in_array(arr, 20)
```

OUTPUT:

```
C:\Users\Dell\Desktop\21MIS1021>ruby 1.rb  
10
```