

RUBY LAB ASSESSMENT 3

21MIS1021 VIMAL KUMAR S

1. Write a ruby code using the following keywords yield, lambda and procs.

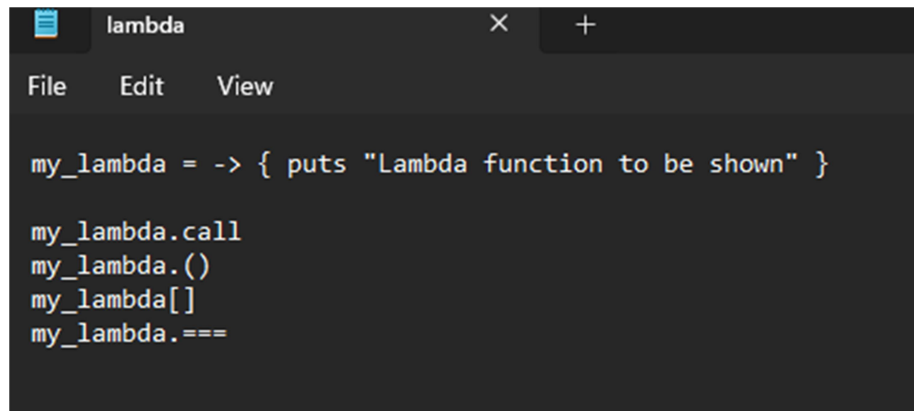
YIELD

```
def one_two_three
  yield 1
  yield 2
  yield 3
end

one_two_three { |number| puts number * 10 }
# 10, 20, 30
```

```
C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby yield.rb
10
20
30
```

LAMBDA



```
lambda

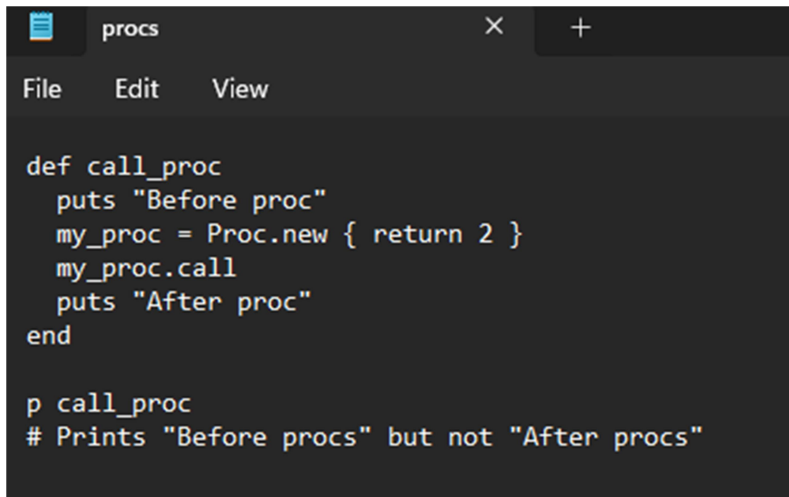
File Edit View

my_lambda = -> { puts "Lambda function to be shown" }

my_lambda.call
my_lambda.()
my_lambda[]
my_lambda.===
```

```
C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby lambda.rb
Lambda function to be shown
Lambda function to be shown
Lambda function to be shown
Lambda function to be shown
```

PROCS



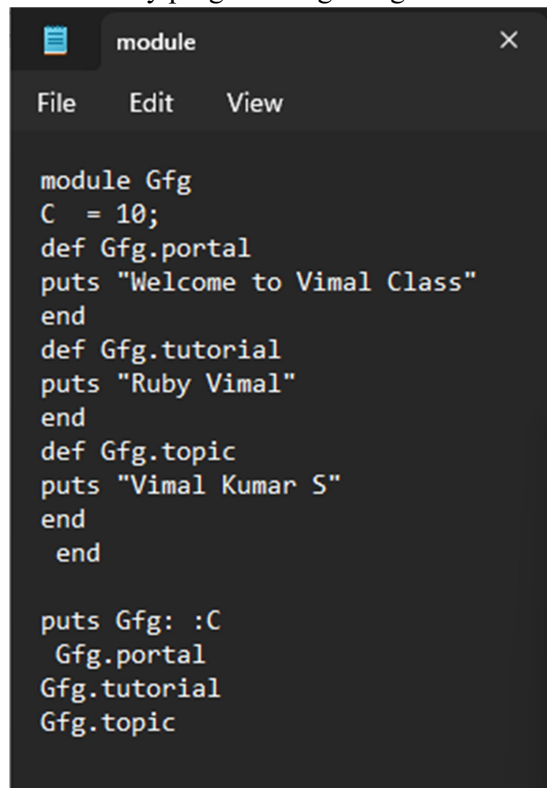
```
procs
File Edit View

def call_proc
  puts "Before proc"
  my_proc = Proc.new { return 2 }
  my_proc.call
  puts "After proc"
end

p call_proc
# Prints "Before procs" but not "After procs"
```

```
C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby procs.rb
Before proc
2
```

2. Write a ruby programming using Modules concept.

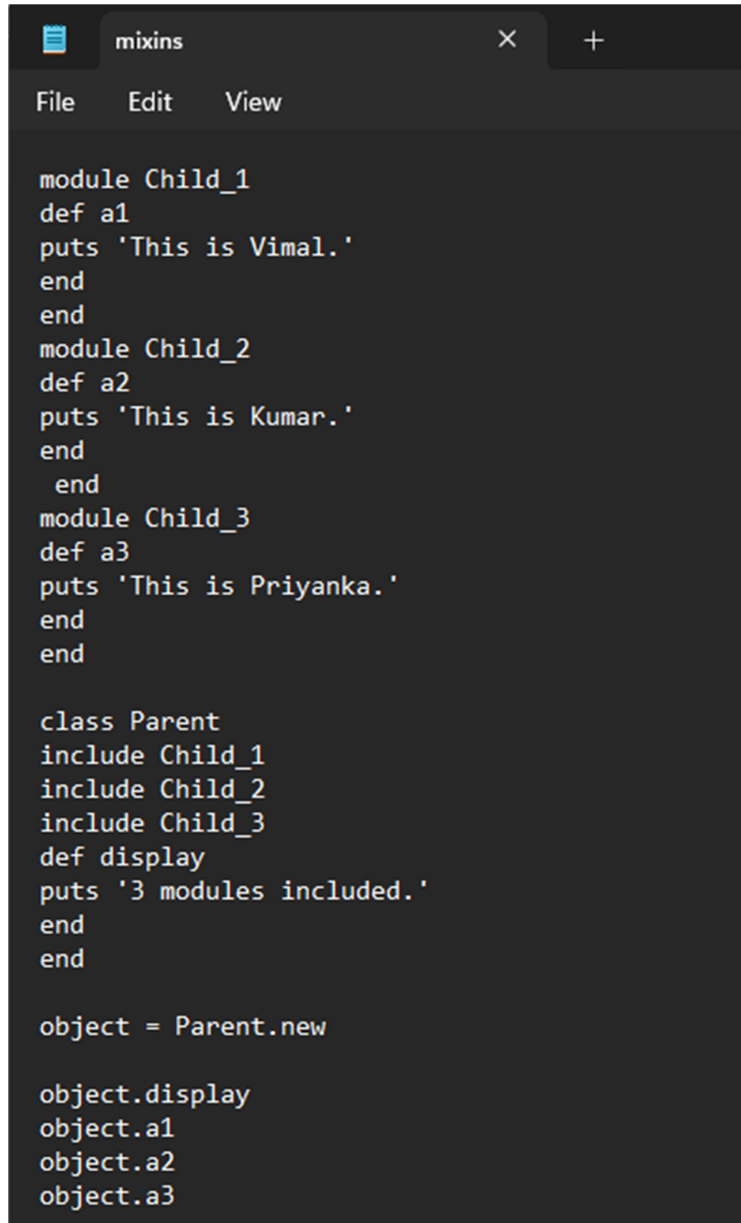


```
module
File Edit View

module Gfg
  C = 10;
  def Gfg.portal
    puts "Welcome to Vimal Class"
  end
  def Gfg.tutorial
    puts "Ruby Vimal"
  end
  def Gfg.topic
    puts "Vimal Kumar S"
  end
end

puts Gfg: :C
  Gfg.portal
  Gfg.tutorial
  Gfg.topic
```

3. Write a ruby programming using Mixins concept.

A screenshot of a code editor window titled 'mixins'. The editor shows a Ruby script that defines three modules (Child_1, Child_2, Child_3) and a class (Parent). Child_1 has a method 'a1' that prints 'This is Vimal.'. Child_2 has a method 'a2' that prints 'This is Kumar.'. Child_3 has a method 'a3' that prints 'This is Priyanka.'. The Parent class includes all three modules and has a 'display' method that prints '3 modules included.'. The script then creates a new instance of Parent, assigns it to 'object', and calls 'object.display', 'object.a1', 'object.a2', and 'object.a3' in sequence.

```
module Child_1
  def a1
    puts 'This is Vimal.'
  end
end

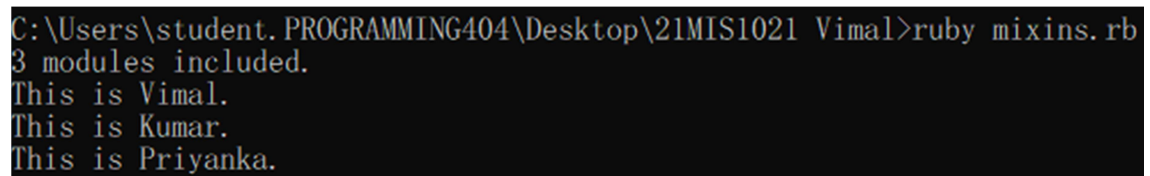
module Child_2
  def a2
    puts 'This is Kumar.'
  end
end

module Child_3
  def a3
    puts 'This is Priyanka.'
  end
end

class Parent
  include Child_1
  include Child_2
  include Child_3
  def display
    puts '3 modules included.'
  end
end

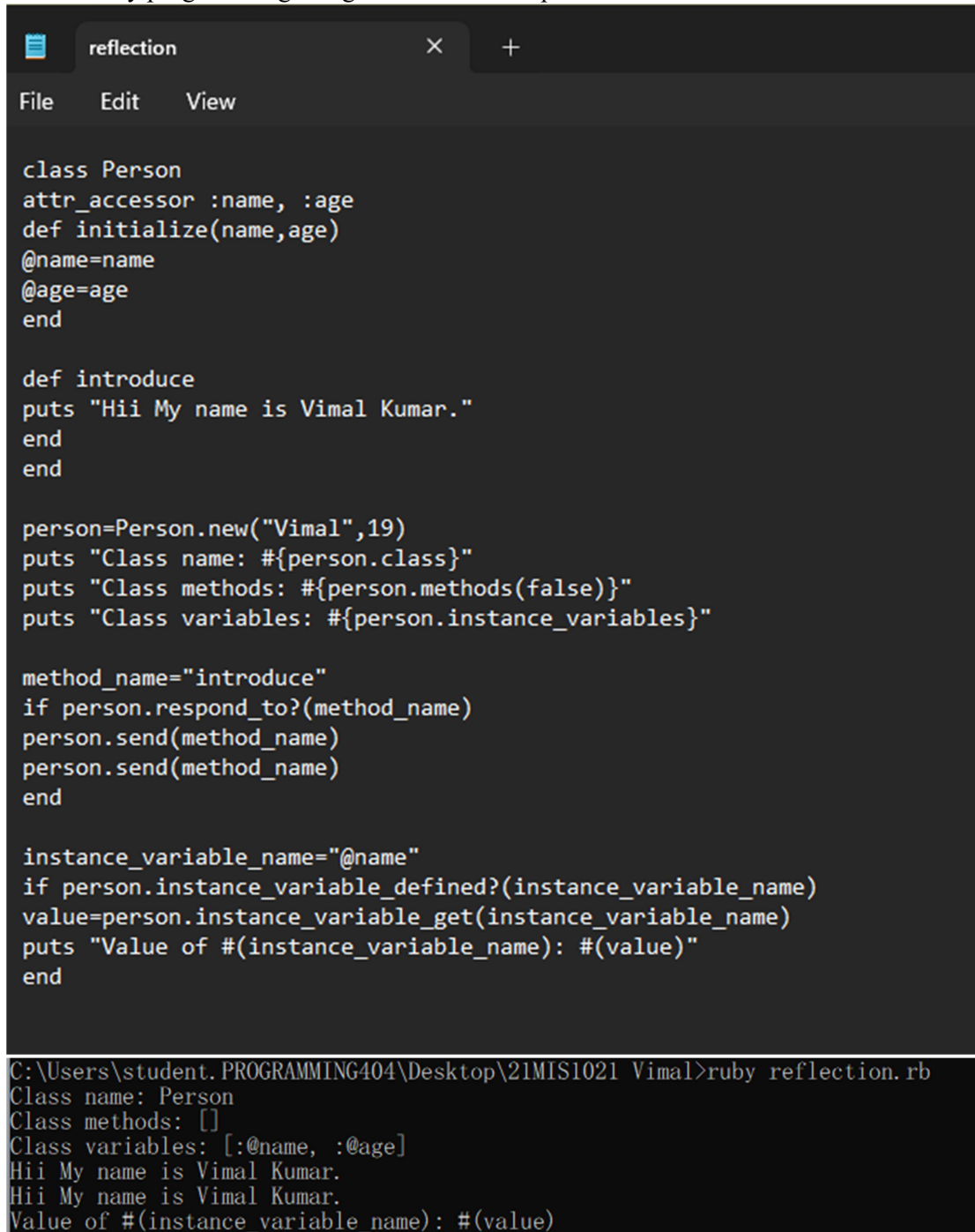
object = Parent.new

object.display
object.a1
object.a2
object.a3
```

A screenshot of a terminal window showing the execution of the 'mixins' script. The command 'C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby mixins.rb' is entered. The output shows '3 modules included.', followed by the three individual print statements: 'This is Vimal.', 'This is Kumar.', and 'This is Priyanka.'.

```
C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby mixins.rb
3 modules included.
This is Vimal.
This is Kumar.
This is Priyanka.
```

4. Write a ruby programming using Reflection concept.



The image shows a code editor window titled 'reflection' with a dark theme. The editor contains a Ruby script that defines a 'Person' class with attributes 'name' and 'age', an 'initialize' method, an 'introduce' method that prints a message, and reflection methods to inspect the class and instance. Below the editor, a terminal window shows the command to run the script and its output, which includes the class name, methods, variables, the introduction message, and the value of the instance variable '@name'.

```
class Person
  attr_accessor :name, :age
  def initialize(name,age)
    @name=name
    @age=age
  end

  def introduce
    puts "Hii My name is Vimal Kumar."
  end

  person=Person.new("Vimal",19)
  puts "Class name: #{person.class}"
  puts "Class methods: #{person.methods(false)}"
  puts "Class variables: #{person.instance_variables}"

  method_name="introduce"
  if person.respond_to?(method_name)
    person.send(method_name)
    person.send(method_name)
  end

  instance_variable_name="@name"
  if person.instance_variable_defined?(instance_variable_name)
    value=person.instance_variable_get(instance_variable_name)
    puts "Value of #{instance_variable_name}: #{value}"
  end
end
```

C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby reflection.rb
Class name: Person
Class methods: []
Class variables: [:@name, :@age]
Hii My name is Vimal Kumar.
Hii My name is Vimal Kumar.
Value of #{instance_variable_name}: #{value}

5. Write a ruby programming using Meta-programming concept

File Edit View

```
class MyClass
  def self.create_method(name)
    self.class_eval do
      define_method(name) do |arg|
        puts "Hello I am Vimal Priy..! , #{arg}!"
      end
    end
  end
end
```

```
obj=MyClass.new
MyClass.create_method(:greet)
obj.greet("World")
```

```
C:\Users\student.PROGRAMMING404\Desktop\21MIS1021 Vimal>ruby meta.rb
Hello I am Vimal Priy..! , #{arg}!
```

6. Create an array a=[1,2,3,4,5,6], and perform the following:
- Different ways to access the array elements

```
C:\> Start Command Prompt with Ruby - irb
irb(main):001:0> a=[1,2,3,4,5,6,7]
=> [1, 2, 3, 4, 5, 6, 7]
irb(main):002:0> a[2]
=> 3
irb(main):003:0> a[-4]
=> 4
irb(main):004:0> a[-4,5]
=> [4, 5, 6, 7]
irb(main):005:0> a.at(0)
=> 1
irb(main):006:0> a.first
=> 1
irb(main):007:0> a.take(3)
=> [1, 2, 3]
irb(main):008:0> a.drop(3)
=> [4, 5, 6, 7]
irb(main):009:0>
```

- Five different methods associated with array.

```

irb(main):012:0> a=[1,2,3,4,5,6]
=> [1, 2, 3, 4, 5, 6]
irb(main):013:0> a.last
=> 6
irb(main):014:0> a.length
=> 6
irb(main):015:0> a.first
=> 1
irb(main):016:0> a.reverse
=> [6, 5, 4, 3, 2, 1]
irb(main):017:0> a.join
=> "123456"
irb(main):018:0>

```

c. Different ways to add and delete an element of an array.

```

C:\ Start Command Prompt with Ruby - irb
irb(main):017:0> a.join
=> "123456"
irb(main):018:0> a=[1,2,3,4,5,6]
=> [1, 2, 3, 4, 5, 6]
irb(main):019:0> a.push(5.5)
=> [1, 2, 3, 4, 5, 6, 5.5]
irb(main):020:0> a.<<7
=> [1, 2, 3, 4, 5, 6, 5.5, 7]
irb(main):021:0> a.unshift(0)
=> [0, 1, 2, 3, 4, 5, 6, 5.5, 7]
irb(main):022:0> a.insert(4, 'Vimal')
=> [0, 1, 2, 3, "Vimal", 4, 5, 6, 5.5, 7]
irb(main):023:0> a.pop
=> 7
irb(main):024:0> a.shift
=> 0
irb(main):025:0>

```

d. Introduce two new arrays and perform intersection, concatenation, difference.

```

irb(main):025:0> a=[1,2,3,4,5]& b=[1,2,3]
=> [1, 2, 3]
irb(main):026:0> a=[4,5,6]+ b=[1,2,3]
=> [4, 5, 6, 1, 2, 3]
irb(main):027:0> a=[4,5,6]+ b=[3,4,6]
=> [4, 5, 6, 3, 4, 6]
irb(main):028:0>

```

- e. Perform a binary search using array a.

```
C:\> Start Command Prompt with Ruby - irb
irb(main):026:0> a=[4,5,6]+ b=[1,2,3]
=> [4, 5, 6, 1, 2, 3]
irb(main):027:0> a=[4,5,6]+ b=[3,4,6]
=> [4, 5, 6, 3, 4, 6]
irb(main):028:0> a=[1,2,3,4,5,6]
=> [1, 2, 3, 4, 5, 6]
irb(main):029:0> a.bsearch{|x|x>=4}
=> 4
irb(main):030:0> a.bsearch{|x|x>=5}
=> 5
irb(main):031:0> a.bsearch{|x|x>=7}
=> nil
irb(main):032:0> a.bsearch{|x|x>=0}
=> 1
irb(main):033:0> a.bsearch{|x|x>=1.5}
=> 2
irb(main):034:0>
```