

RUBY LAB -2

Name: Vishwanth P

Register No: 21MIS1117

Assessment 2: Ruby programs using classes, objects, instance methods, and class methods

1. A program that has a method with and without argument
2. Method name ends with 2 different types of symbols other than “?” and “=”
3. Example program of Instance Method
4. Example program of Class Method
5. Bank Account System (Get Account details, check the balance before withdraw and update balance)
6. Student Grade book System
7. Employee Salary Management System

Question 1

```
def show_age(name, age)
  puts "Name: #{name}"
  puts "Age: #{age}"
end

puts "Program With Arguments"
print "Enter Name: "
name = gets.chomp
print "Enter Age: "
age = gets.chomp
show_age(name, age)

def show_age()
  puts "Program Without Arguments"
  print "Enter Name: "
  name = gets.chomp
  print "Enter Age: "
  age = gets.chomp
  puts "Name: #{name}"
  puts "Age: #{age}"
end

show_age()
```

Q1 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2.rb
Program With Arguments
Enter Name: Vishwa
Enter Age: 20
Name: Vishwa
Age: 20
Program Without Arguments
Enter Name: Vishwa
Enter Age: 20
Name: Vishwa
Age: 20
PS D:\7th Sem\F1 - Ruby\Lab> █
```

Question 2

```
cricketer = "Dhoni"

def batsman!(name)
  puts "#{name} is a batsman"
end

def greet_(name)
  puts "Hello #{name}"
end

greet_(cricketer)
batsman!(cricketer)
```

Q2 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_2.rb
Hello Dhoni
Dhoni is a batsman
█
```

Question 3

```
class VIT
  def initialize(name, regNo = nil, empid = nil)
    @name = name
    @regNo = regNo
    @empid = empid
  end

  def student
    "#{@name}, Reg No: #{@regNo}"
  end

  def faculty
    "#{@name}, Emp ID: #{@empid}"
  end
end

stud = VIT.new("vishwa", 1117)
emp = VIT.new("yogesh", nil, 123)

puts stud.student
puts emp.faculty
```

Q3 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_3.rb
vishwa, Reg No: 1117
yogesh, Emp ID: 123
```

Question 4

```
class VIT
  @@person = 0

  def initialize(name, role)
    @name = name
    @role = role
    @@person += 1
  end

  def self.total_persons
    "Total persons in college is #{@@person}"
  end
end

p1 = VIT.new("Vishwa", "Student")
p2 = VIT.new("Yogesh", "Faculty")
p3 = VIT.new("Bruce", "Student")
p4 = VIT.new("Raja", "Faculty")

puts VIT.total_persons
```

Q4 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_4.rb
Total persons in college is 4_
```

Question 5

```
class BankAccount
  def initialize(name, ib = 0)
    @name = name
    @balance = ib
  end

  def details
    "Acc_Details: #{@name}, Balance: #{@balance}"
  end

  def bal
    "Balance: #{@balance}"
  end

  def withdraw(amount)
    if amount > @balance
      "Insufficient funds."
    else
      @balance -= amount
      "Withdrew #{amount}. New balance: #{@balance}"
    end
  end

  def deposit(amount)
    @balance += amount
    "Deposited #{amount}. New balance: #{@balance}"
  end
end

account = BankAccount.new("Vishwa", 5000)
puts account.details
puts account.withdraw(500)
puts account.bal
puts account.deposit(200)
puts account.bal
```

Q5 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_5.rb
Acc_Details: Vishwa, Balance: 5000
Withdrew 500. New balance: 4500
Balance: 4500
Deposited 200. New balance: 4700
Balance: 4700
```

Question 6

```
class GradeBook
  def initialize(name)
    @name = name
    @grades = {}
  end

  def add_grade(subject, grade)
    @grades[subject] = grade
  end

  def calculate_grade(grade)
    case grade
    when 0..49
      'F'
    when 50..55
      'E'
    when 56..60
      'D'
    when 61..70
      'C'
    when 71..80
      'B'
    when 81..90
      'A'
    when 91..100
      'S'
    else
      'Invalid grade'
    end
  end
end
```

```

def display_grades
  puts "Grade Book for #{@name}:"
  @grades.each do |subject, grade|
    grade_type = calculate_grade(grade)
    puts "#{subject}: #{grade} - Grade: #{grade_type}"
  end
end
end

grade_book = GradeBook.new("Vishwa")

grade_book.add_grade("Ruby", 81)
grade_book.add_grade("Maths", 72)
grade_book.add_grade("TOC", 64)

grade_book.display_grades

```

Q6 Output

```

● PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_6.rb
Grade Book for Vishwa:
Ruby: 81 - Grade: A
Maths: 72 - Grade: B
TOC: 64 - Grade: C

```


Question 7

```
class Employee
  def initialize(name, basic, da_pct, hra_pct)
    @name = name
    @basic = basic
    @da = da(da_pct)
    @hra = hra(hra_pct)
  end

  def da(da_pct)
    (@basic * da_pct / 100.0)
  end

  def hra(hra_pct)
    (@basic * hra_pct / 100.0)
  end

  def total
    (@basic + @da + @hra)
  end

  def display
    puts "#{@name}: Basic: #{@basic}, DA: #{@da}, HRA: #{@hra}, Total: #{total}"
  end
end

dhoni = Employee.new("Vishwa", 50000, 20, 15)
kohli = Employee.new("Prakash", 60000, 25, 20)
dhoni.display
kohli.display
```

Q7 Output

```
PS D:\7th Sem\F1 - Ruby\Lab> ruby 21MIS1117_Lab_2_7.rb
Vishwa: Basic: 50000, DA: 10000.0, HRA: 7500.0, Total: 67500.0
Prakash: Basic: 60000, DA: 15000.0, HRA: 12000.0, Total: 87000.0
```