1. OOPS Program

* Program

class Rectangle

def initialize(length, breadth)

@length = length

@breadth = breadth

end

def getLength

@length

end

def getBreadth

@breadth

end

def setLength=(length)

@length = length

end

def setBreadth=(breadth)

@breadth = breadth

end

private :getLength, :getBreadth

def perimeter

2 \* (getLength + getBreadth)

end

def printPerimeter

puts "Perimeter of Rectangle with #{@length} & #{@breadth} is #{2 \* (getLength + getBreadth)}."

end

end

class Square < Rectangle

def initialize(side)

@length = @breadth = side

end

def printPerimeter

puts "Perimeter of Square with #{@length} is #{perimeter}."

end

end

sq = Square.new(10)

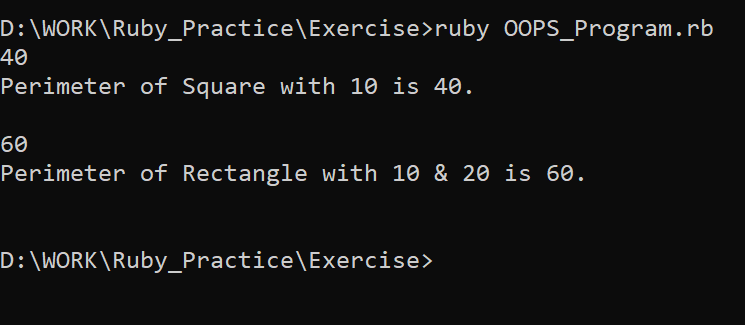
puts sq.perimeter

puts sq.printPerimeter

rc = Rectangle.new(10,20)

puts rc.perimeter

puts rc.printPerimeter

* Output

1. Operator Overloading Program

* Program

class ABC

attr\_accessor:num

def initialize(num)

@num = num

end

def +(other)

return @num + other

end

def \*(other)

return @num \* other

end

def \*\*(other)

return @num \*\* other

end

end

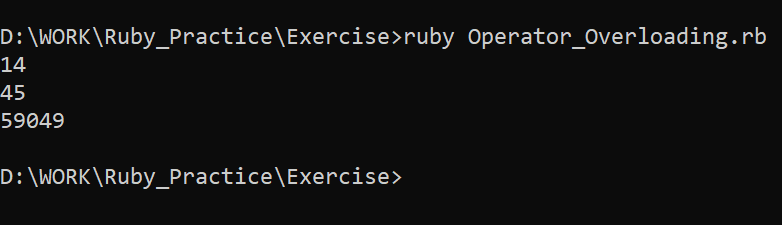
a = ABC.new(9)

puts a + 5

puts a \* 5

puts a \*\* 5

* Output



1. Attendance System Program

* Program

time = Time.new

puts "Attendance System by Kishan Aghera"

puts "Date: #{time.strftime("%d-%m-%Y")}"

puts "Are you a teacher?(y or n)"

choice = gets.chomp

if choice == 'n'

puts "Only teachers can use it."

else

data = {"001" => "n","002" => "n","003" => "n"}

puts "Enter your name: "

name = gets.chomp

count = 0

data.each { |key,value|

puts "Roll no: #{key}"

puts "Present?(y or n)"

ch = gets.chomp

if ch == 'y'

data[key] = "y"

count += 1

end

}

puts "\n\nReport"

puts "Date: #{time.strftime("%d-%m-%Y")}"

puts "Name of Teacher: #{name}"

puts "Total Number of Students : #{data.size}"

puts "Number of Students Present: #{count}"

puts "Number of Students Absent: #{data.size-count}"

puts "\nNumber\tPresent(y or n)"

data.each { |key, value|

puts "#{key}\t#{value}"

}

end

* Output

