

Csc600- Homework Ruby

Name – Mya Phyu

1)

```
4 # Loop:
5 i = 0
6 loop do
7   i += 1
8   puts i
9   break
10 end
11
12 # While:
13 x = 3
14 while x >= 1
15   puts "Hi"
16   x = x - 1
17 end
18
19 # For:
20 i = "yo"
21 for a in 1..3 do
22   puts i
23 end
24
25 # Until:
26 var = 7
27 until var == 11 do
28   puts var * 10
29   var = var + 1
30 end
31
32 # Each:
33 (0..9).each do |i|
34   puts i
35 end
36
37 # Collect:
38 a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
39 b = a.collect { |y| (5 * y) }
40 puts b
41
42 # Times:
43 7.times do |i|
44   puts i
45 end
46
47 # Upto:
48 4.upto(7) do |n|
49   puts n
50 end
51
52 # Downto:
53 7.downto(4) do |n|
54   puts n
55 end
56
57 # Step:
58 (0..60).step(10) do |i|
59   puts i
60 end
61
62 # Each_line:
63 "Welcome\nto\nRailway\nPortal".each_line do |i|
64   puts i
65 end
66
67 # Map:
68 array = [1, 2, 3]
69 puts array.map { |n| n * 2 }
70
71 # Collect (again):
72 a = [1, 2, 3, 4]
73 puts "collect a : #{a.collect { |x| x + 1 }}"
74
75 # Select:
76 c = [18, 22, 3, 3, 50, 6]
77 puts "select method : #{c.select { |num| num > 10 }}"
78
79 # Reject:
80 puts "reject method : #{c.reject { |num| num > 10 }}"
81
```

PROBLEMS	OUTPUT	DEBUG CONSOLE	TERMINAL
1	Hi		
	Hi		
	Hi		
	Yo		
	Yo		
	Yo		
	70		
	80		
	90		
	100		
	0		
	1		
	2		
	3		

```
4
5
6
7
8
9
5
10
15
20
25
30
35
40
45
```

```
50
0
1
2
3
4
5
6
4
5
6
7
7
6
5
4
0
10
20
30
40
50
60
Welcome
to
Railway
Portal
2
4
6
collect a : [2, 3, 4, 5]
```

```
select method : [18, 22, 50]
reject method : [3, 3, 6]
```

2)

```
84 class Array
85   def limited?(amin, amax)
86     self.all? { |x| x >= amin && x <= amax }
87   end
88
89   def sorted?
90     return 0 if self.empty?
91
92     increasing = self.each_cons(2).all? { |a, b| a <= b }
93     decreasing = self.each_cons(2).all? { |a, b| a >= b }
94
95     return 1 if increasing
96     return -1 if decreasing
97
98     0
99   end
100 end
101
102 # arr1 = [1, 2, 3, 4, 5]
103 # puts arr1.limited?(1, 5) # true
104
105 arr1 = [1, 2, 3, 4, 5]
106
107 arr2 = [5, 4, 3, 2, 1]
108 arr3 = [3, 2, 5, 4, 1]
109 arr4 = [10, 20, 30, 40]
110
111 puts arr1.limited?(1, 5)
112 puts arr2.limited?(1, 5)
113 puts arr3.limited?(1, 5)
114 puts arr4.limited?(10, 40)
115
116 puts arr1.sorted?
117 puts arr2.sorted?
118 puts arr3.sorted?
119 puts arr4.sorted?
120
```

```
true
true
true
true
1
-1
0
1
```

3)

```
class Triangle
  attr_accessor :a, :b, :c
  def initialize(a, b, c)
    @a = a
    @b = b
    @c = c
  end

  def perimeter
    return @a + @b + @c
  end

  def area
    s = self.perimeter / 2.0
    Math.sqrt(s * (s - @a) * (s - @b) * (s - @c))
  end

  def test
    sides = [@a, @b, @c].sort
    if sides[0] + sides[1] <= sides[2]
      puts "Not a triangle"
    elsif @a == @b && @b == @c
      puts "Equilateral Triangle"
    elsif @a == @b || @b == @c || @c == @a
      puts "Isosceles Triangle"
    elsif @a**2 + @b**2 == @c**2 || @a**2 + @c**2 == @b**2 || @b**2 + @c**2 == @a**2
      puts "Right Triangle"
    else
      puts "Scalene Triangle"
    end
  end
end

# Create a triangle
triangle = Triangle.new(3, 4, 5)

# Test the triangle type
triangle.test

# Compute perimeter and area
puts "Perimeter: #{triangle.perimeter}"
puts "Area: #{triangle.area}"
```

```
Right Triangle
Perimeter: 12
Area: 6.0
```

4)

```
class Sphere
  attr_accessor :radius
  def initialize(radius)
    @radius = radius
  end

  def area
    4 * Math::PI * @radius**2
  end

  def volume
    (4.0 / 3.0) * Math::PI * @radius**3
  end
end

class Ball < Sphere
  attr_accessor :color

  def initialize(radius, color)
    super(radius)
    @color = color
  end
end

class MyBall < Ball
  attr_accessor :owner

  def initialize(radius, color, owner)
    super(radius, color)
    @owner = owner
  end

  def show
    puts "Owner: #{@owner}"
    puts "Color: #{@color}"
    puts "Radius: #{@radius}"
    puts "Surface Area: #{area}"
    puts "Volume: #{volume}"
  end
end

my_ball = MyBall.new(5, "Red", "John")
my_ball.show
```

```
Owner: John
Color: Red
Radius: 5
Surface Area: 314.1592653589793
Volume: 523.5987755982989
```