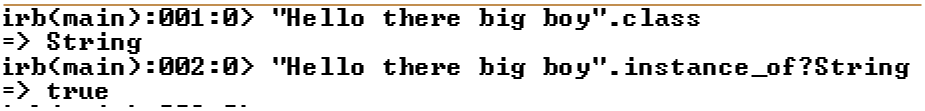
Subject: Exploring Ruby **Practical 2** **Code: COMP47530**
Student ID: 19200086 **Date: 29/09/2019**

1]

a)

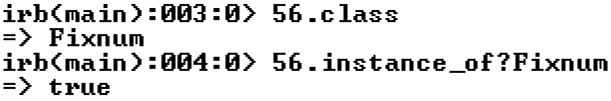


```
irb(main):001:0> "Hello there big boy".class
=> String
irb(main):002:0> "Hello there big boy".instance_of?String
=> true
```

.class returns -> class name of the instance

instance_of returns -> boolean

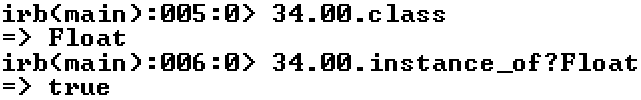
b)



```
irb(main):003:0> 56.class
=> Fixnum
irb(main):004:0> 56.instance_of?Fixnum
=> true
```

If number within 31 bits -> Fixnum, else, Bignum.

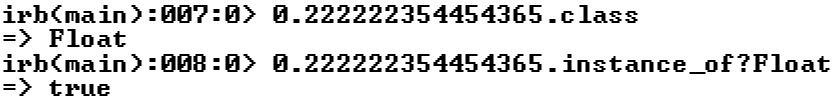
c)



```
irb(main):005:0> 34.00.class
=> Float
irb(main):006:0> 34.00.instance_of?Float
=> true
```

34.00 -> floating point literal, hence the results.

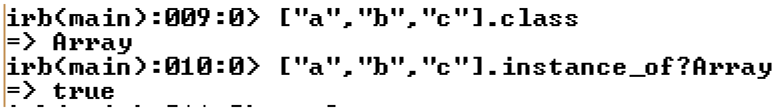
d)



```
irb(main):007:0> 0.222222354454365.class
=> Float
irb(main):008:0> 0.222222354454365.instance_of?Float
=> true
```

FLOAT::MIN < 0.222222354454365 < FLOAT::MAX, thus its class is Float.

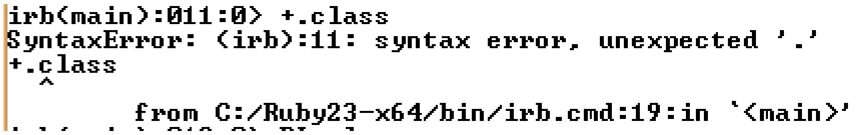
e)



```
irb(main):009:0> ["a","b","c"].class
=> Array
irb(main):010:0> ["a","b","c"].instance_of?Array
=> true
```

[“a”,”b”,”c”] -> Array containing String values, hence, Array class object.

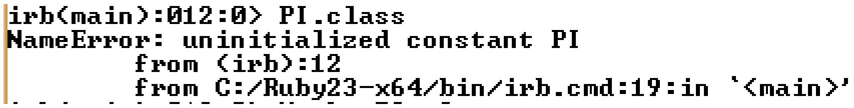
f)



```
irb(main):011:0> +.class
SyntaxError: (irb):11: syntax error, unexpected '.'
+.class
  ^
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

‘+’ -> operator, not instance. “.” cannot be used on operator.

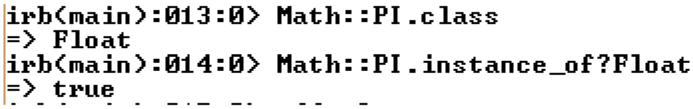
g)



```
irb(main):012:0> PI.class
NameError: uninitialized constant PI
        from (irb):12
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

PI -> treated as a constant. Unassigned before use, hence the failure.

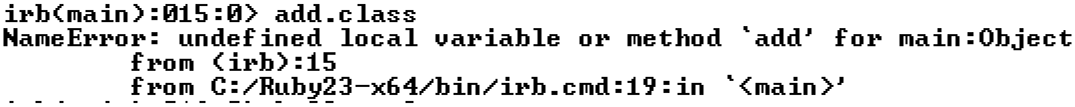
h)



```
irb(main):013:0> Math::PI.class
=> Float
irb(main):014:0> Math::PI.instance_of?Float
=> true
```

PI -> Float constant in Math module (Math::PI).

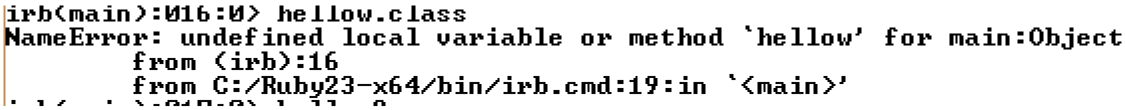
i)



```
irb(main):015:0> add.class
NameError: undefined local variable or method `add' for main:Object
        from (irb):15
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

Un-initialized local variable add, hence the failure.

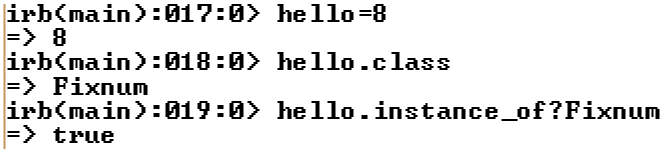
j)



```
irb(main):016:0> hellow.class
NameError: undefined local variable or method `hellow' for main:Object
        from (irb):16
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

hellow -> treated as locally scoped variable

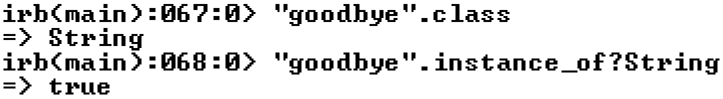
k)



```
irb(main):017:0> hello=8
=> 8
irb(main):018:0> hello.class
=> Fixnum
irb(main):019:0> hello.instance_of?Fixnum
=> true
```

Hello -> Variable holding an integer value. Therefore, hello.class -> Fixnum

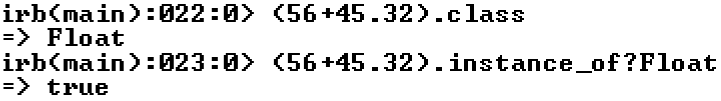
l)



```
irb(main):067:0> "goodbye".class
=> String
irb(main):068:0> "goodbye".instance_of?String
=> true
```

“Goodbye” -> Instance of String class, hence the results.

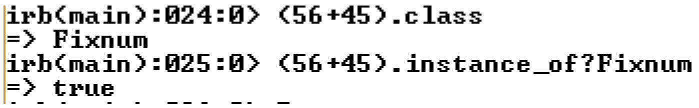
m)



```
irb(main):022:0> (56+45.32).class
=> Float
irb(main):023:0> (56+45.32).instance_of?Float
=> true
```

Int + float = float, hence, class=Float, result.instance_of?Float=true

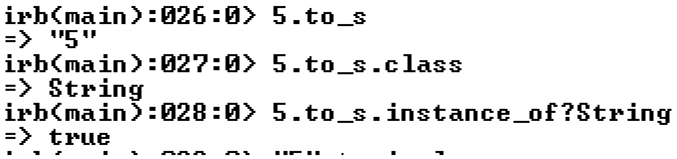
n)



```
irb(main):024:0> (56+45).class
=> Fixnum
irb(main):025:0> (56+45).instance_of?Fixnum
=> true
```

56 -> Fixnum, 45 -> Fixnum. Fixnum + Fixnum -> Fixnum. Hence the results.

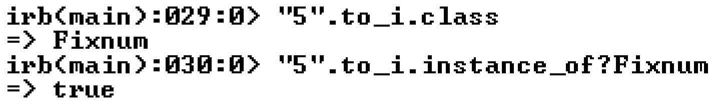
o)



```
irb(main):026:0> 5.to_s
=> "5"
irb(main):027:0> 5.to_s.class
=> String
irb(main):028:0> 5.to_s.instance_of?String
=> true
```

.to_s -> converts to String. Hence, 5.to_s.class = String

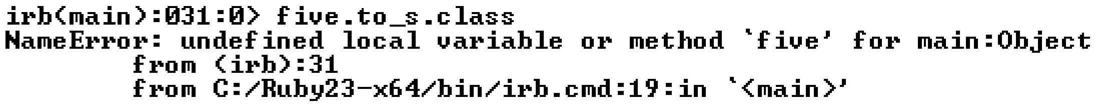
p)



```
irb(main):029:0> "5".to_i.class
=> Fixnum
irb(main):030:0> "5".to_i.instance_of?Fixnum
=> true
```

.to_i -> transforms to a number. Thus, “5”.to_i.class -> Fixnum

q)

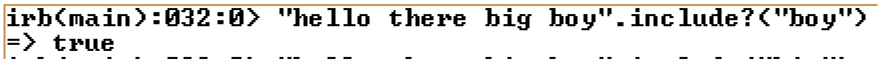


```
irb(main):031:0> five.to_s.class
NameError: undefined local variable or method `five' for main:Object
        from (irb):31
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

Five -> considered as a local variable. Undefined before use -> hence the error.

2]

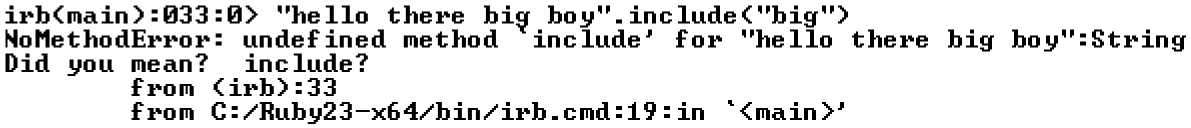
a)



```
irb(main):032:0> "hello there big boy".include?("boy")
=> true
```

Include? Returns true if the argument is present in the String, false otherwise.

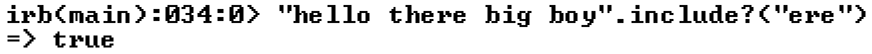
b)



```
irb(main):033:0> "hello there big boy".include("big")
NoMethodError: undefined method `include' for "hello there big boy":String
Did you mean?  include?
        from (irb):33
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

“Include?” is required.

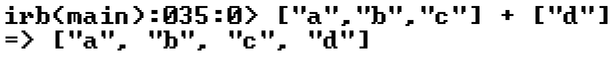
c)



```
irb(main):034:0> "hello there big boy".include?("ere")
=> true
```

Checks if the given string contains the string passed as an argument. Entire word is not matched.

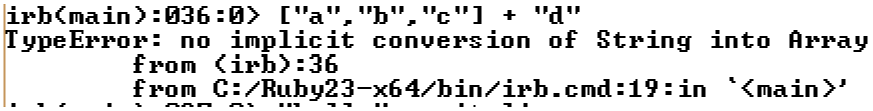
d)



```
irb(main):035:0> ["a","b","c"] + ["d"]
=> ["a", "b", "c", "d"]
```

Two arrays have been concatenated using the ‘+’ operator.

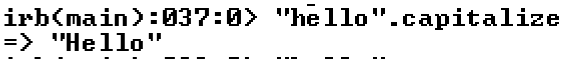
e)



```
irb(main):036:0> ["a","b","c"] + "d"
TypeError: no implicit conversion of String into Array
        from (irb):36
        from C:/Ruby23-x64/bin/irb.cmd:19:in `<main>'
```

“+” needs similar operands -> String “d” cannot be changed to an Array, hence the error.

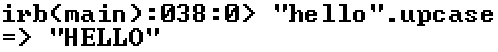
f)



```
irb(main):037:0> "hello".capitalize
=> "Hello"
```

Change the first letter to uppercase using capitalize method.

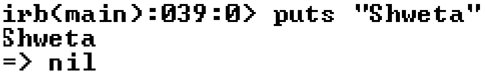
g)



```
irb(main):038:0> "hello".upcase
=> "HELLO"
```

Change the entire String to uppercase using upcase method.

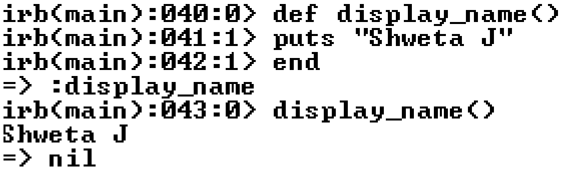
h)



```
irb(main):039:0> puts "Shweta"
Shweta
=> nil
```

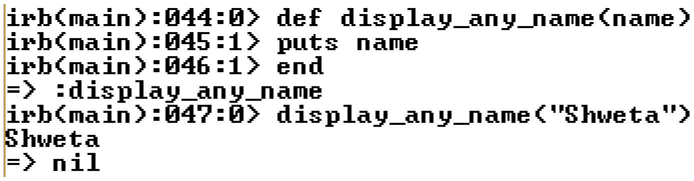
puts prints a string on the console.

i)



```
irb(main):040:0> def display_name()
irb(main):041:1> puts "Shweta J"
irb(main):042:1> end
=> :display_name
irb(main):043:0> display_name()
Shweta J
=> nil
```

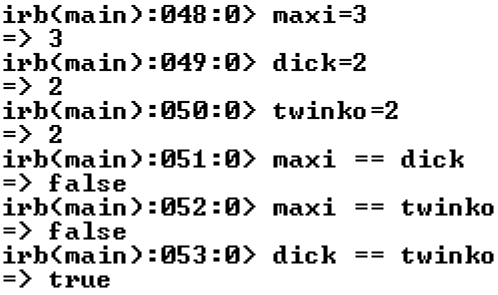
j)



```
irb(main):044:0> def display_any_name(name)
irb(main):045:1> puts name
irb(main):046:1> end
=> :display_any_name
irb(main):047:0> display_any_name("Shweta")
Shweta
=> nil
```

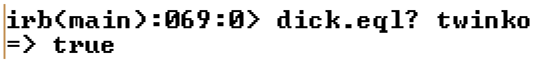
The function takes a “name” argument and prints on the console.

k)



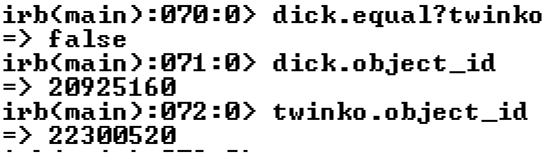
```
irb(main):048:0> maxi=3
=> 3
irb(main):049:0> dick=2
=> 2
irb(main):050:0> twinko=2
=> 2
irb(main):051:0> maxi == dick
=> false
irb(main):052:0> maxi == twinko
=> false
irb(main):053:0> dick == twinko
=> true
```

“==” -> returns true if numeric values are the same



```
irb(main):069:0> dick.eql? twinko
=> true
```

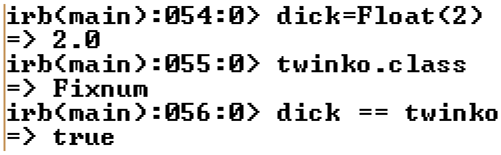
.eql? -> returns true if the same hash key is referred



```
irb(main):070:0> dick.equal?twinko
=> false
irb(main):071:0> dick.object_id
=> 20925160
irb(main):072:0> twinko.object_id
=> 22300520
```

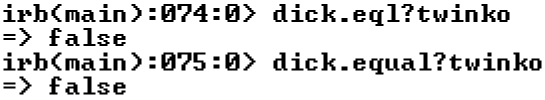
.equal? -> checks if the same object is referred, object_id is different, hence the result.

l)



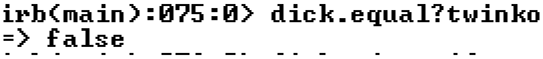
```
irb(main):054:0> dick=Float(2)
=> 2.0
irb(main):055:0> twinko.class
=> Fixnum
irb(main):056:0> dick == twinko
=> true
```

‘==’ -> no change

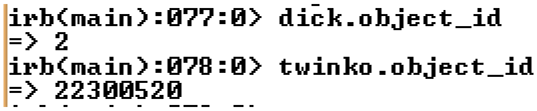


```
irb(main):074:0> dick.eql?twinko
=> false
irb(main):075:0> dick.equal?twinko
=> false
```

‘.eql?’ -> false, different result



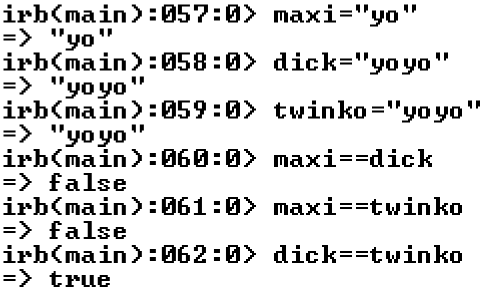
```
irb(main):075:0> dick.equal?twinko
=> false
```



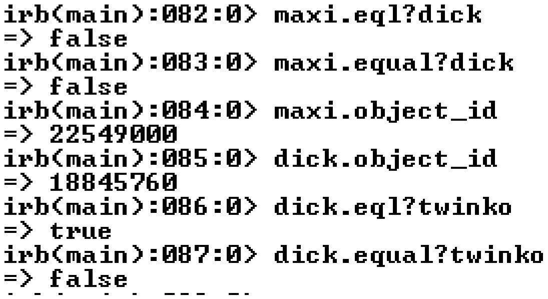
```
irb(main):077:0> dick.object_id
=> 2
irb(main):078:0> twinko.object_id
=> 22300520
```

‘.equal?’ -> false, same result (different object is referred)

m)



```
irb(main):057:0> maxi="yo"
=> "yo"
irb(main):058:0> dick="yoyo"
=> "yoyo"
irb(main):059:0> twinko="yoyo"
=> "yoyo"
irb(main):060:0> maxi==dick
=> false
irb(main):061:0> maxi==twinko
=> false
irb(main):062:0> dick==twinko
=> true
```

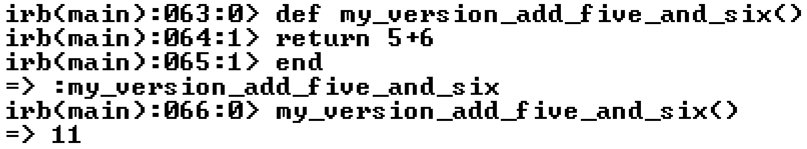


```
irb(main):082:0> maxi.eql?dick
=> false
irb(main):083:0> maxi.equal?dick
=> false
irb(main):084:0> maxi.object_id
=> 22549000
irb(main):085:0> dick.object_id
=> 18845760
irb(main):086:0> dick.eql?twinko
=> true
irb(main):087:0> dick.equal?twinko
=> false
```

No changes in the results with Strings.

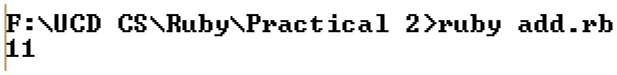
3] Predicate methods give a boolean result. E.g.: 5.even? => false

4]



```
irb(main):063:0> def my_version_add_five_and_six()
irb(main):064:1> return 5+6
irb(main):065:1> end
=> :my_version_add_five_and_six
irb(main):066:0> my_version_add_five_and_six()
=> 11
```

5]



```
F:\UCD CS\Ruby\Practical 2>ruby add.rb
11
```