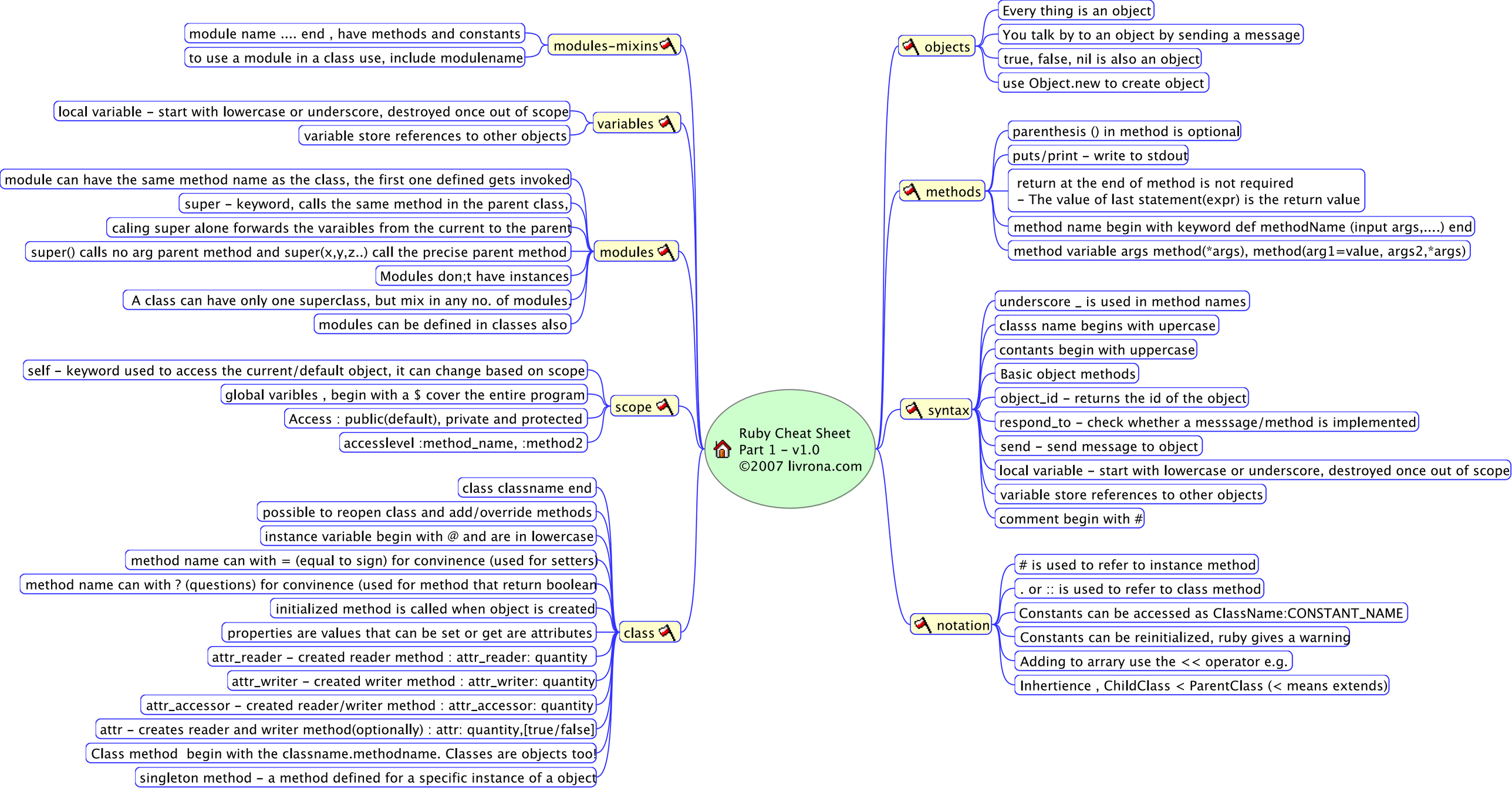
Ruby Cheat Sheet
Part 1 - v1.0
©2007 livrona.com
objects
Every thing is an object
You talk by to an object by sending a message
true, false, nil is also an object
use Object.new to create object
methods
parenthesis () in method is optional
puts/print - write to stdout
return at the end of method is not required
- The value of last statement(expr) is the return value
method name begin with keyword def methodName (input args,....) end
method variable args method(*args), method(arg1=value, args2,*args)
syntax
underscore _ is used in method names
classs name begins with upercase
contants begin with uppercase
Basic object methods
object_id - returns the id of the object
respond_to - check whether a messsage/method is implemented
send - send message to object
local variable - start with lowercase or underscore, destroyed once out of scope
variable store references to other objects
comment begin with #
notation
is used to refer to instance method
. or :: is used to refer to class method
Constants can be accessed as ClassName:CONSTANT_NAME
Constants can be reinitialized, ruby gives a warning
Adding to arrary use the << operator e.g.
Inhertience , ChildClass < ParentClass (< means extends)
modules-mixins
module name .... end , have methods and constants
to use a module in a class use, include modulename
variables
local variable - start with lowercase or underscore, destroyed once out of scope
variable store references to other objects
modules
module can have the same method name as the class, the first one defined gets invoked
super - keyword, calls the same method in the parent class,
caling super alone forwards the varaibles from the current to the parent
super() calls no arg parent method and super(x,y,z..) call the precise parent method
Modules don;t have instances
A class can have only one superclass, but mix in any no. of modules
modules can be defined in classes also
scope
self - keyword used to access the current/default object, it can change based on scope
global varibles , begin with a $ cover the entire program
Access : public(default), private and protected
accesslevel :method_name, :method2
class
class classname end
possible to reopen class and add/override methods
instance variable begin with @ and are in lowercase
method name can with = (equal to sign) for convinence (used for setters)
method name can with ? (questions) for convinence (used for method that return boolean
initialized method is called when object is created
properties are values that can be set or get are attributes
attr_reader - created reader method : attr_reader: quantity
attr_writer - created writer method : attr_writer: quantity
attr_accessor - created reader/writer method : attr_accessor: quantity
attr - creates reader and writer method(optionally) : attr: quantity,[true/false]
Class method begin with the classname.methodname. Classes are objects too!
singleton method - a method defined for a specific instance of a object