







UNIT OVERVIEW: [Ruby Part 1](#)

| Lesson | Course | Exercises | Objectives | Progression Pathways | Time (min) |
|--------|--|-----------|--|--|------------|
| 1 | 1. Introduction to Ruby | 1 - 9 | <ul style="list-style-type: none"> Understand what Ruby is and its basic data types: strings, numbers and Booleans Understand and implement the 'puts' and 'print' commands Understand what methods are and implement the following string methods: .length, .reverse, .upcase, .downcase. Understand what variables are Recognise and implement basic math and the exponentiation (**) and modulo(%) operators | L2 data L1 algorithms L3 programming | 20 |
| 2 | 1. Introduction to Ruby | 10 - 16 | <ul style="list-style-type: none"> Understand the syntax of a single and multi-line comment Understand the conventions of naming variables and practise creating variables Practise multiple methods on a string | L4 algorithms L3 programming | 15 |
| 3 | 2. Putting the Form in Formatter | 1 - 7 | <ul style="list-style-type: none"> Plenary exercise synthesising Module 1 : Introduction to Ruby Understand and implement the gets and .chomp methods Understand using (!) to modify the value contained within a variable Understand and implement string interpolation | L3 programming | 20 |
| 4 | 3. Control Flow in Ruby | 1 - 8 | <ul style="list-style-type: none"> Understand the following: control flow, environment, whitespace and expressions Understand and implement the following operators : '==', '!=', '<', '>', '<=', '>=' Understand and implement if / else, elsif and unless statements | L4 programming L3 algorithms | 15 |
| 5 | 3. Control Flow in Ruby | 9 - 17 | <ul style="list-style-type: none"> Understand the AND (&&), OR () and NOT (!) operators Practise expressions with comparators and Boolean operators Practise implementing if, elsif and unless statements | L3 algorithms L4 data L5 programming | 20 |
| 6 | 4. Thith Meanth War! | 1 - 8 | <ul style="list-style-type: none"> Plenary activity synthesising Module 3 : Control Flow in Ruby Understand and implement the .gsub! and .include! methods | L3 programming | 15 |
| 7 | 5. Loops & Iterators | 1 - 7 | <ul style="list-style-type: none"> Recognise and implement the following assignment operators '+=', '-=', '*=' Understand and implement a while loop and until loop Understand and implement a for loop Practise a puts statement with string interpolation | L4/7 programming L8 programming | 15 |
| 8 | 5. Loops & Iterators | 8 - 18 | <ul style="list-style-type: none"> Recognise the loop iterator and implement it using the do and break keywords Understand and implement the next keyword in a for loop Understand and create an array Understand what the .each method and the .times iterator method | L5 algorithms L6 data L6/7 programming | 15 |

UNIT OVERVIEW: [Ruby Part 2](#)

| Lesson | Course | Exercises | Objectives | Progression Pathways | Time (min) |
|--------|---|-----------|--|--|------------|
| 9 | 6. Redacted! | 1 – 6 | <ul style="list-style-type: none"> Plenary activity synthesising Module 5: Loops & Iterators Practise puts command and get.chomp and an if / else statement Understand and implement the .split and the .each methods | | 15 |
| 10 | 7. Data Structures | 1 – 9 | <ul style="list-style-type: none"> Understand and practise creating multi-dimensional arrays Practise accessing an array element and creating arrays of strings Understand and implement a hash and practise using the Hash.new notation Understand how to add to the hash using bracket notation and accessing a hash value | L6 data  L6/8 programming  L4 algorithms  | 20 |
| 11 | 7. Data Structures | 10 – 16 | <ul style="list-style-type: none"> Recognise that iterators can be used to iterate arrays and hashes Practise using .each iterator to access each element of an array Practise creating and iterating over a multidimensional array Practise creating and iterating over a hash | L5 algorithms  L6/8 programming   | 20 |
| 12 | 8. Create a Histogram | 1 – 8 | <ul style="list-style-type: none"> Plenary activity synthesising Module 7: Data Structures Practise creating a hash with a default value Practise implementing the .each method and the sort_by and .reverse! keywords Practise iterating over the hash and using the .to_s method | L6 programming  | 20 |
| 13 | 9. Methods, Blocks, & Sorting | 1 – 8 | <ul style="list-style-type: none"> Understand how methods are useful and why they are used Understand the difference between a parameter and argument Practise passing an argument through a method Understand a splat argument and the return keyword | L7 programming    | 30 |
| 14 | 9. Methods, Blocks, & Sorting | 9 – 19 | <ul style="list-style-type: none"> Understand blocks and how they are different to methods Understand the importance of sorting and implement the .sort! method Understand and implement the combined comparison operator <=> Practise implementing a descending sort | L6 programming  | 30 |
| 15 | 10. Ordering Your Library | 1 – 6 | <ul style="list-style-type: none"> Plenary exercise synthesising Module 9: Methods, Blocks, & Sorting Create a method with parameters Implement .sort! into the method and call the method | L6 programming  | 20 |
| 16 | 11. Hashes and Symbols | 1 – 7 | <ul style="list-style-type: none"> Practise creating and iterating over a hash Understand the Nil value Understand a symbol in a hash Compare a symbol and a string as a key in a method Understand and implement the .object_id method | | 30 |

UNIT OVERVIEW: [Ruby Part 3](#)

| Lesson | Course | Exercises | Objectives | Progression Pathways | Time (min) |
|--------|--|-----------|---|--|------------|
| 17 | 11. Hashes and Symbols | 8 – 15 | <ul style="list-style-type: none"> Understand how to implement the following methods : <code>.to_sym</code>, <code>.to_s</code> and <code>.intern</code> Implement the <code>.select</code> method to select specific key/value pairs in a hash Understand the <code>.each_key</code> and <code>.each_value</code> methods Recognise that symbols are faster than strings | L4 algorithms   | 20 |
| 18 | 12. A Night at the Movies | 1 – 10 | <ul style="list-style-type: none"> Plenary activity synthesising Module 13: Hashes and Symbols Practise creating a hash and implement a case statement Practise implementing an if / else statement Practise iterating over a hash | L4/5 algorithms    | 20 |
| 19 | 13. The Zen of Ruby | 1 - 7 | <ul style="list-style-type: none"> Understand and implement a simpler if and a one line unless statement Understand and implement the ternary conditional statement and the conditional assignment operator | L7 data  L4/5 algorithms  L3 programming  | 20 |
| 20 | 13. The Zen of Ruby | 8 - 14 | <ul style="list-style-type: none"> Understand and implement the : <code>.respond_to</code>, <code>.upto</code>, <code>.downto</code> and <code>.next</code> methods Understand and implement the concatenation operator | L6 programming  | 20 |
| 21 | 13. The Zen of Ruby | 15 – 20 | <ul style="list-style-type: none"> Understand and implement refactoring Practise implementing a ternary statement and a conditional operator Practise implementing a method with an implicit return Practise refactoring a for loop using <code>.times</code> | | 20 |
| 22 | 14. The Refactor Factory | 1 – 7 | <ul style="list-style-type: none"> Plenary activity synthesising Module 13: The Zen of Ruby Practise implementing an implicit return and a concatenation operator Practise using the conditional assignment operator and a single line if and unless statement | L4/5 algorithms    | 20 |
| 23 | 15. Blocks, Procs, and Lambdas | 1 - 9 | <ul style="list-style-type: none"> Understand and implement the <code>.collect</code> method and the <code>yield</code> keyword Understand a Proc, its syntax and its use Practice creating a Proc | L7 programming    | 20 |
| 24 | 15. Blocks, Procs, and Lambdas | 10 – 21 | <ul style="list-style-type: none"> Recognise that a Proc can be called using <code>.call</code> Understand how to combine a symbol with a Proc Understand and practise creating a lambda Evaluate the difference between a lambda and Proc Implement the <code>.is_a?</code> method | L6 programming  | 30 |

UNIT OVERVIEW: [Ruby Part 4](#)

| Lesson | Course | Exercises | Objectives | Progression Pathways | Time (min) |
|--------|--|-----------|---|--|------------|
| 25 | 16. Object-Oriented Programming I | 1 – 10 | <ul style="list-style-type: none"> Understand a class and create a method within a class Understand an instance variable and practise creating one Recognise scope and the different scopes of variables Recognise the syntax of a class and global variables and practise creating them | L7 programming   | 30 |
| 26 | 16. Object-Oriented Programming I | 11 - 20 | <ul style="list-style-type: none"> Understand and implement inheritance Understand that inheritance can be overridden Understand and implement the super keyword Recognise the syntax for ending a method on a single line Understand that Ruby doesn't allow multiple inheritance | | 30 |
| 27 | 17. Virtual Computer | 1 – 8 | <ul style="list-style-type: none"> Plenary activity synthesising Module 16: Object-Oriented Programming I Practise creating a class and a method Create parameters of a method and an instance variable Recognise and implement the Time.now keyword Create a new method and call a class variable within it | L3 programming   L5 algorithms | 30 |
| 28 | 18. Object-Oriented Programming II | 1 – 8 | <ul style="list-style-type: none"> Evaluate the difference between a public and private method Practise creating private and public methods Understand the function of the following methods: attr_reader, attr_writer and attr_accessor Understand and create a module | L4 programming     | 20 |
| 29 | 18. Object-Oriented Programming II | 9 – 18 | <ul style="list-style-type: none"> Understand the scope resolution operator syntax Understand how to use require and include with modules Understand and implement a mixin Understand and evaluate the extend keyword Practise creating a method and mixin and creating a public and private method | L8 hardware     | 30 |
| 30 | 19. Banking on Ruby | 1 - 7 | <ul style="list-style-type: none"> Plenary activity synthesising Module 18: Object-Oriented Programming II Understand what an optional parameter is Practise creating a class with public and private methods Practise creating instances of a class | L5 programming   | 20 |