Python Programming Study Notes and Practice Questions

Days 5-6: Python and Java Programming

Day 5: Python Programming

Python Fundamentals - Study Notes

1. Python Basics

- **Features:** Interpreted, Object-oriented, High-level, Dynamic typing
- **Python Execution:** Source code → Bytecode → Python Virtual Machine
- **Memory Management:** Automatic garbage collection, reference counting

2. Data Types

```
# Primitive Types
integer = 42
floating = 3.14
string = "Hello World"
boolean = True

# Collections
list_data = [1, 2, 3, 4]
tuple_data = (1, 2, 3, 4)
dict_data = {"key": "value", "age": 25}
set_data = {1, 2, 3, 4}
```

3. Control Structures

```
# Conditional
if condition:
    statement
elif another_condition:
    statement
else:
    statement

# Loops
for item in iterable:
    statement
```

```
while condition:
    statement

# List Comprehension
squares = [x**2 for x in range(10)]
```

4. Functions

```
# Function Definition
def function_name(param1, param2="default"):
    """Docstring"""
    return result

# Lambda Functions
square = lambda x: x**2
filter(lambda x: x > 0, numbers)
map(lambda x: x*2, numbers)
```

5. Object-Oriented Programming

```
class ClassName:
    class_variable = "shared"

    def __init__(self, param):
        self.instance_variable = param

    def method(self):
        return self.instance_variable

    @staticmethod
    def static_method():
        return "static"

    @classmethod
    def class_method(cls):
        return cls.class_variable

# Inheritance
class ChildClass(ClassName):
    def __init__(self, param, extra):
        super().__init__(param)
        self.extra = extra
```

6. File Handling

```
# Reading Files
with open("file.txt", "r") as file:
    content = file.read()
    lines = file.readlines()
    line = file.readline()
```

```
# Writing Files
with open("file.txt", "w") as file:
    file.write("Hello World")
    file.writelines(["line1\n", "line2\n"])
```

7. Exception Handling

```
try:
    risky_operation()
except SpecificError as e:
    handle_specific_error(e)
except Exception as e:
    handle_general_error(e)
else:
    runs_if_no_exception()
finally:
    always_runs()
```

Python Practice Questions

MCQs 51-75

51. Python is:
    a) Compiled language b) Interpreted language c) Both d) Neither
52. Which is mutable in Python?
    a) Tuple b) String c) List d) Integer
53. What is the output of `print(type([]))`?
    a) `<class 'array'>` b) `<class 'list'>` c) `<class 'tuple'>` d) `<class 'dict'>`
54. List comprehension `[x for x in range(5)]` produces:
    a) [1,2,3,4,5] b) [0,1,2,3,4] c) [0,1,2,3,4,5] d) Error
55. Which method adds element to end of list?
    a) add() b) append() c) insert() d) extend()
56. `dict.get(key, default)` returns:
    a) Key if present b) Value if key present c) Default if key absent d) Both b and c
57. Python `__init__` method is:
    a) Constructor b) Destructor c) Class method d) Static method
58. Multiple inheritance in Python is:
    a) Not supported b) Supported c) Only with interfaces d) Causes errors
59. `with` statement is used for:
    a) Loops b) Conditions c) Resource management d) Functions
60. Lambda function `lambda x, y: x + y` is equivalent to:

```
a) def func(x, y): return x + y
b) def func(x, y): x + y
```

```
c) def func(): return x + y
d) None of the above
```

61. Which is NOT a Python built-in data type?
    a) list b) tuple c) array d) dict
62. `len("Hello")` returns:
    a) 4 b) 5 c) 6 d) Error
63. Python uses which method for string formatting?
    a) format() b) % c) f-strings d) All of the above
64. `range(1, 10, 2)` generates:
    a) [1,3,5,7,9] b) [1,3,5,7] c) [2,4,6,8] d) [1,2,3,4,5]
65. Which operator is used for floor division in Python?
    a) / b) // c) % d) **
66. Python list slicing `lst[1:4]` includes:
    a) Index 1 to 4 b) Index 1 to 3 c) Index 0 to 3 d) Index 2 to 4
67. `enumerate()` function returns:
    a) Only indices b) Only values c) Index-value pairs d) Length
68. Python `pass` statement:
    a) Terminates program b) Does nothing c) Passes parameters d) Causes error
69. List method `pop()` without arguments removes:
    a) First element b) Last element c) Random element d) All elements
70. Python `super()` is used for:
    a) Creating superclass b) Accessing parent class c) Multiple inheritance d) Class variables
71. Which is correct way to create empty dictionary?
    a) {} b) dict() c) Both a and b d) []
72. `"hello".upper()` returns:
    a) "HELLO" b) "Hello" c) "hello" d) Error
73. Python `__str__` method is used for:
    a) String representation b) String comparison c) String length d) String conversion
74. `break` statement in Python:
    a) Exits function b) Exits loop c) Exits program d) Skips iteration
75. Python `is` operator checks:
    a) Value equality b) Type equality c) Identity d) Membership

Day 6: Java Programming

Java Fundamentals - Study Notes

1. Java Basics

- **Features:** Platform independent, Object-oriented, Strongly typed
- **JVM Architecture:** Class Loader, Runtime Data Area, Execution Engine
- **Compilation:** Source code (.java) → Bytecode (.class) → JVM execution

2. Data Types and Variables

```
// Primitive Types
int number = 42;
double decimal = 3.14;
char character = 'A';
boolean flag = true;

// Reference Types
String text = "Hello World";
int[] array = {1, 2, 3, 4};
```

3. Object-Oriented Programming

```
// Class Definition
public class Person {
    private String name;
    private int age;

    // Constructor
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }

    // Getter/Setter
    public String getName() { return name; }
    public void setName(String name) { this.name = name; }

    // Method
    public void displayInfo() {
        System.out.println("Name: " + name + ", Age: " + age);
    }
}

// Inheritance
public class Student extends Person {
    private String studentId;

    public Student(String name, int age, String studentId) {
        super(name, age); // Call parent constructor
        this.studentId = studentId;
    }
```

```
    @Override
    public void displayInfo() {
        super.displayInfo();
        System.out.println("Student ID: " + studentId);
    }
}
```

4. Interfaces and Abstract Classes

```
// Interface
public interface Drawable {
    void draw(); // Abstract method

    default void display() { // Default method (Java 8+)
        System.out.println("Displaying...");
    }
}

// Abstract Class
public abstract class Shape {
    protected String color;

    public abstract double area(); // Abstract method

    public void setColor(String color) { // Concrete method
        this.color = color;
    }
}
```

5. Exception Handling

```
try {
    riskyOperation();
} catch (SpecificException e) {
    System.out.println("Specific error: " + e.getMessage());
} catch (Exception e) {
    System.out.println("General error: " + e.getMessage());
} finally {
    cleanupCode();
}

// Custom Exception
public class CustomException extends Exception {
    public CustomException(String message) {
        super(message);
    }
}
```

6. Collections Framework

```
// List
List<String> list = new ArrayList<>();
list.add("Item");
list.get(0);
list.size();

// Set
Set<String> set = new HashSet<>();
set.add("Unique");
set.contains("Unique");

// Map
Map<String, Integer> map = new HashMap<>();
map.put("key", 100);
map.get("key");
```

Java Practice Questions

MCQs 76-100

76. Java is:
    a) Platform dependent b) Platform independent c) Hardware specific d) OS specific
77. Which is NOT a Java primitive type?
    a) int b) double c) String d) boolean
78. `public static void main(String[] args)` - 'static' means:
    a) Method belongs to class b) Method belongs to object c) Method is final d) Method is private
79. Java constructor:
    a) Has return type b) Has no return type c) Returns object d) Returns void
80. Method overriding requires:
    a) Same class b) Inheritance c) Same package d) Static methods
81. `final` keyword prevents:
    a) Inheritance b) Method overriding c) Variable modification d) All of the above
82. Interface in Java:
    a) Can have constructors b) Can have instance variables c) Can have abstract methods d) Can be instantiated
83. Exception handling uses:
    a) try-catch b) try-finally c) try-catch-finally d) All of the above
84. `ArrayList` implements:
    a) List interface b) Set interface c) Map interface d) Queue interface
85. `==` operator in Java compares:
    a) Values only b) References c) Both d) Neither

86. Abstract class:
    a) Cannot be instantiated b) Can have concrete methods c) Can have abstract methods d) All of the above
87. `super` keyword is used to:
    a) Call parent constructor b) Access parent methods c) Access parent variables d) All of the above
88. Java packages:
    a) Organize classes b) Provide namespace c) Control access d) All of the above
89. `this` keyword refers to:
    a) Current object b) Parent object c) Class object d) Static context
90. Multithreading in Java is achieved through:
    a) Thread class b) Runnable interface c) Both d) Neither
91. `String` class in Java is:
    a) Mutable b) Immutable c) Both d) Neither
92. Access modifier `protected` allows access from:
    a) Same class only b) Same package and subclasses c) Everywhere d) Same package only
93. Static variables are:
    a) Instance specific b) Class specific c) Method specific d) Block specific
94. Java `equals()` method:
    a) Compares references b) Compares content c) Both d) Neither
95. `HashSet` allows:
    a) Duplicates b) Null values c) Ordered elements d) None of the above
96. Java `synchronized` keyword:
    a) Prevents thread interference b) Allows concurrent access c) Improves performance d) None of the above
97. `instanceof` operator:
    a) Checks object type b) Creates instance c) Compares values d) Checks inheritance
98. Java generics provide:
    a) Type safety b) Code reusability c) Performance improvement d) All of the above
99. `finally` block:
    a) Always executes b) Executes only if exception occurs c) Executes only if no exception d) May not execute
100. Java `clone()` method:
    a) Creates shallow copy b) Creates deep copy c) Depends on implementation d) Cannot be used

Additional Practice Questions for Networking and Advanced Topics

Networking Concepts (Questions 101-110)

101. OSI model has how many layers?
    a) 5 b) 6 c) 7 d) 8
102. TCP is:
    a) Connection-oriented b) Connectionless c) Both d) Neither
103. IP address 192.168.1.1 is:
    a) Public b) Private c) Multicast d) Broadcast
104. HTTP uses which port by default?
    a) 21 b) 80 c) 443 d) 25
105. DNS stands for:
    a) Domain Name System b) Data Network Service c) Digital Name Server d) Dynamic Network System
106. DHCP is used for:
    a) Domain resolution b) IP address assignment c) File transfer d) Email
107. Subnet mask 255.255.255.0 represents:
    a) /24 b) /16 c) /8 d) /32
108. TCP three-way handshake involves:
    a) SYN, ACK, FIN b) SYN, SYN-ACK, ACK c) SYN, ACK, RST d) SYN, FIN, ACK
109. Which protocol is connectionless?
    a) TCP b) UDP c) HTTP d) FTP
110. MAC address is:
    a) 32-bit b) 48-bit c) 64-bit d) 128-bit

Answer Keys

Python (Questions 51-75)

51. b) Interpreted language
52. c) List
53. b) `<class 'list'>`
54. b) [0,1,2,3,4]
55. b) append()
56. d) Both b and c
57. a) Constructor
58. b) Supported
59. c) Resource management

60. a) def func(x, y): return x + y
61. c) array
62. b) 5
63. d) All of the above
64. a) [1,3,5,7,9]
65. b) //
66. b) Index 1 to 3
67. c) Index-value pairs
68. b) Does nothing
69. b) Last element
70. b) Accessing parent class
71. c) Both a and b
72. a) "HELLO"
73. a) String representation
74. b) Exits loop
75. c) Identity

Java (Questions 76-100)

76. b) Platform independent
77. c) String
78. a) Method belongs to class
79. b) Has no return type
80. b) Inheritance
81. d) All of the above
82. c) Can have abstract methods
83. d) All of the above
84. a) List interface
85. b) References
86. d) All of the above
87. d) All of the above
88. d) All of the above
89. a) Current object
90. c) Both
91. b) Immutable
92. b) Same package and subclasses

93. b) Class specific
94. b) Compares content
95. b) Null values
96. a) Prevents thread interference
97. a) Checks object type
98. d) All of the above
99. a) Always executes
100. c) Depends on implementation

Networking (Questions 101-110)

101. c) 7
102. a) Connection-oriented
103. b) Private
104. b) 80
105. a) Domain Name System
106. b) IP address assignment
107. a) /24
108. b) SYN, SYN-ACK, ACK
109. b) UDP
110. b) 48-bit

Quick Reference - Programming Concepts

Time Complexities

- Array access: O(1)
- Binary search: O(log n)
- Sorting algorithms: O(n log n) for efficient ones
- Hash table operations: Average O(1)

Key Programming Principles

- **DRY:** Don't Repeat Yourself
- **SOLID:** Single Responsibility, Open/Closed, Liskov Substitution, Interface Segregation, Dependency Inversion
- **Encapsulation:** Data hiding
- **Inheritance:** Code reusability

- **Polymorphism:** One interface, multiple implementations

Memory Management

- **Stack:** Local variables, method calls
- **Heap:** Dynamic memory allocation
- **Garbage Collection:** Automatic memory cleanup

This comprehensive set of notes and practice questions covers all the essential topics needed for the BCA Data Analyst role examination. Focus on understanding concepts rather than memorizing, and practice solving similar problems to build confidence.