2025 OOP Midterm Exam

Date: 2025/05/05 Name: Student ID:

Object-Oriented Programming (OOP) and Python Basics

1. **True or False**: In object-oriented programming, a class defines the blueprint for objects, specifying their attributes and methods, while an object is an instance of a class.
2. **True or False**: In Python, a subclass can only inherit from a single superclass, unlike languages like C++ that support multiple inheritance.
3. **True or False**: In Python, a class can have multiple `__init__` methods to initialize objects with different sets of arguments.
4. **True or False**: Encapsulation in Python is achieved by using private attributes (e.g., `_attribute` or `__attribute`) to hide implementation details and expose only a public interface.
5. **True or False**: Polymorphism in Python requires all classes to inherit from a common superclass to share method names.
6. **True or False**: In Python, an abstract class (defined using the `abc` module) cannot be instantiated directly and requires subclasses to implement its abstract methods.

Graph and Search

1. **True or False**: A graph consists of nodes and edges, and can be either directed (edges have direction) or undirected (edges are bidirectional).
2. **True or False**: Breadth-First Search (BFS) is guaranteed to find the shortest path in a weighted graph.

3. **True or False**: Depth-First Search (DFS) is an informed search algorithm that uses heuristic information to guide its exploration of a graph.
4. **True or False**: In Python, BFS is typically implemented using a queue (e.g., `collections.deque`), while DFS can be implemented using recursion or a stack.
5. **True or False**: DFS explores a graph by diving as deep as possible along one path before backtracking, while BFS explores all nodes at the current depth before moving to the next depth.
6. **True or False**: Only DFS, and not BFS, can be used to detect cycles in a graph.

Data Types in Python

1. **True or False**: In Python, lists are mutable, allowing elements to be added or removed, while tuples are immutable and cannot be modified after creation.
2. **True or False**: For the Python list `L = [1, 'x', 3, [4, 5]]`, the expression `L[4]` evaluates to the nested list `[4, 5]`.
3. **True or False**: Python lists have a built-in method to efficiently add elements to the beginning, similar to `collections.deque`'s `appendleft()`.
4. **True or False**: The Python `pop()` method, when called without arguments, removes and returns the last element of a list.
5. **True or False**: Python dictionaries allow duplicate keys, but maintain the insertion order of keys (since Python 3.7).
6. **True or False**: In Python, a set is an ordered collection that automatically removes duplicate elements.

Clean Code Principles

1. **True or False**: A function that modifies a global variable introduces a side effect, which can make the code harder to test and maintain.
2. **True or False**: The following code violates Clean Code principles due to unclear variable names:

```
x = ["Austin", "New York", "San Francisco"]
for i in x:
    print(i)
```

3. **True or False**: According to Clean Code principles, functions should have a single responsibility and avoid doing multiple unrelated tasks.
4. **True or False**: The following code follows Clean Code principles by using clear variable names:

```
import datetime
date_str = datetime.date.today().strftime("%Y-%m-%d")
```

5. **True or False**: Refactoring all duplicate code into a single function always improves maintainability, regardless of the context.
6. **True or False**: Functions with many arguments can increase complexity and lead to more test cases, violating Clean Code principles.

Python vs. C++

1. **True or False**: Python uses dynamic typing, where variable types are determined at runtime, while C++ uses static typing, requiring explicit type declarations.
2. **True or False**: Python provides built-in high-level data types like lists and dictionaries, while C++ relies on standard library types like `std::vector` and `std::map`.

3. **True or False**: C++ code is interpreted at runtime, while Python code is compiled into machine code before execution.
4. **True or False**: Python code can be compiled into machine code using tools like PyInstaller or Cython.
5. **True or False**: Tools like PyBind11 allow developers to create Python modules that call C++ code, enabling integration between the two languages.
6. **True or False**: In C++, memory management is automatic via garbage collection, while Python requires manual memory management (e.g., using `new` and `delete`).