CPSC 250 - Programming for Data Manipulation
Test 3 – **Solution Set**

Part 1 – Multiple Choice (10 points)

1. **b)** The base class's constructor is ignored
2. **d)** The derived method replaces the base version when called from an instance
3. **a)** To allow different object types to respond to the same method call
4. **b)** B speaks
5. **c)** To restrict direct access to some parts of an object

Part 2 – Find the Errors (10 points)

Question 1:

```
class Vehicle:
    def __init__(self, speed):
        speed = speed   # Error: should be self.speed = speed

class Bike(Vehicle):
    def __init__(self, speed, gear):
        self.gear = gear  # Error: base class constructor not called
```

Fix:

```
class Bike(Vehicle):
    def __init__(self, speed, gear):
        super().__init__(speed)
        self.gear = gear
```

```
class Hammer(Tool):
    def __init__(self, name, weight):
        Tool.__init__()    # Error: missing 'name' argument
        self.weight = weight
```

Fix:

```
class Hammer(Tool):
    def __init__(self, name, weight):
        super().__init__(name)
        self.weight = weight
```

Part 3 – Code Writing (30 points)

1. Book and Textbook Classes

```
class Book:
    def __init__(self, title, author):
        self.title = title
        self.author = author

class Textbook(Book):
    def __init__(self, title, author, subject):
        super().__init__(title, author)
        self.subject = subject

    def __str__(self):
        return f"{self.title} by {self.author} [Subject: {self.subject}]"
```

2. Movie Class with Comparison Operators

```
class Movie:
    def __init__(self, title, rating):
        self.title = title
        self.rating = rating

    def __eq__(self, other):
        return self.title == other.title

    def __lt__(self, other):
        return self.rating < other.rating
```

3. Mixin and Inheritance

```
class TimestampMixin:
    def timestamp(self):
        print("Logged at some time")

class Document:
    def __init__(self, filename):
        self.filename = filename
```

```
class PDFDocument(Document, TimestampMixin):
    def print_info(self):
        print(f"File: {self.filename}")
        self.timestamp()
```

Part 4 – Comment the Code (10 points)

```
class Animal:
    def __init__(self, species="unknown"):       # (1) Constructor sets species; uses a
        self._species = species                  # Stored as a protected attribute

    def get_species(self):                        # (2) Getter method to access species
        return self._species

    def speak(self):                              # (3) Base class placeholder for sound
        return "..."

class Dog(Animal):
    def speak(self):                              # (4) Overrides speak to return "Woof!
        return "Woof!"
```