



Quiz 04 Review

Announcements

- **EX05: River Simulation** due Monday at 11:59pm
-

```

1 from __future__ import annotations
2
3 class Node:
4     value: int
5     next: Node | None
6
7     def __init__(self, value: int, next: Node | None):
8         self.value = value
9         self.next = next
10
11     def __str__(self) -> str:
12         rest: str
13         if self.next is None:
14             rest = "None"
15         else:
16             rest = str(self.next)
17         return f"{self.value} -> {rest}"
18
19 pluto: Node = Node(22, None)
20 neptune: Node = Node(15, pluto)

```

5.1. Print the output.

```
1 print(neptune)
```

5.2. Print the output.

```
1 print(pluto.value)
```

5.3. Print the output.

```
1 print(neptune.next)
```

5.4. Print the output.

```
1 print(neptune.next.next)
```

Create a `Book` class that represents a book in a library system. Your class should include:

Attributes:

- `title (string)`: The book's title
- `author (string)`: The book's author
- `pages (integer)`: Total number of pages
- `is_checked_out (boolean)`: Whether the book is currently checked out

Methods:

- `__init__(self, title: str, author: str, pages: int)`: Constructor that initializes the book. Set `is_checked_out` to `False` by default.
- `check_out(self) -> None`: Marks the book as checked out
- `return_book(self) -> None`: Marks the book as available
- `get_reading_time(self) -> float`: Estimates reading time, assuming 250 words per page and 200 words per minute
- `__str__(self) -> str`: Returns a formatted string with the book's information (you choose the string representation!)