3. Linked Lists :: Implementation :: Node Class :: Constructors

The *Node* class implements three constructors which are all straightforward:

```
// Creates a new Node storing no data and with mNext and mPrev set to null.
public Node() {
  this(null);
}

// Creates a new Node storing pData as the data and with mNext and mPrev set to
// null.
public Node(Integer pData) {
  setData(pData);
  setNext(null);
  setPrev(null);
}

// Creates a new Node storing pData as the data, mPrev initialized to pPrev, and
// mNext initialized to pNext.
public Node(Integer pData, Node pPrev, Node pNext) {
  setData(pData);
  setPrev(pPrev);
  setNext(pNext);
}
```

3. Linked Lists :: Implementation :: Node Class :: equals()

```
// Returns true if this Node and pNode are equal to each other where equal is
// defined as:
//
// 1. If pNode is null, returns false.
// 2. If mNode == pNode is true, returns true.
// 3. If the instance variables of this Node are equal to the instance variables of
//    pNode returns true.
// 4. Otherwise, returns false.
@Override
public boolean equals(Object pNode) {
  Node node = (Node)pNode;
  if (node == null) return false;
  if (this == node) return true;
  if (getData() == node.getData() && getNext() == node.getNext() &&
    getPrev() == node.getPrev()) return true;
  return false;
}
```

3. Linked Lists :: Implementation :: Node Class :: toString()

```
// Returns a string representation of this Node where we define the string
// representation to be the string representation of the data stored in this Node.
@Override
public String toString() {
  return "" + getData();
}
```