SUMMARY OF CHANGES

Task 1 : Implementing the GClock Replacement Policy

BasicBufferMgr.java:

Added the following data members to the class:
availableUnpinnedBuffers : To track the number of available unpinned buffers in the bufferpool.
gclock_index : A variable to track the running index to point to a buffer in the bufferpool. Logically, this tracks the current position of the clock head.
bufferpoolsize: A variable to store the size of the bufferpool.

Changes to the pin() and pinNew() methods of this class:
Added lines of code to insert an entry into the 'bufferMap' map. Once a buffer is pinned, the mapping between this buffer and its block number is captured in the map.

Also added code to print the bufferpool contents, pincounts of the buffers and the reference counter values of all the buffers in the bufferpool.

Changes to the chooseUnpinnedBuffer() method:
Added code to call Gclock() method when the relevant condition occurs.

Added the method GClock() :
Implemented the Gclock replacement policy in this method. The method returns the buffer that holds the block that will be evicted as per the GClock policy.

Added the following methods to print the 'debug' statements (which are required for evaluation)
BufferPrintStatus() : Prints the bufferpool contents : Basically lists the block that the buffer is pinned to. The index serves as the buffer number.
PinPrintStatus() : Prints the pin counts for all the buffers in the bufferpool. The index serves as the buffer number in the bufferpool.
ReferenceCountStatus() : Prints the reference counter values for all the buffers in the bufferpool. The index serves as the buffer number in the bufferpool.
UnpinnedBuffers() : Prints the number of available buffers in the bufferpool.

Buffer.java

Added the following data member to the class:
reference_counter : Variable that holds the value of the reference counter for a buffer.

Added the following methods in the class :

getReferenceCounter() : Returns the current value of the reference counter for a buffer.
decrementReferenceCounter() : Decrements the reference counter by 1 for a buffer.
resetReferenceCounter() : Resets the reference counter to the initial value that it must hold. (Taken as an input from the user)
getPinCount() : Returns the pincount for a buffer.

Startup.java

Added the following static data member to this class that holds the value for the number of times the Gclock head must spin.
'reference_counter_initialized' : An int variable

Added code to take the input from the user (for the number of times the Gclock head must spin).

Task 2 : Use Map to keep track of the buffer pool

BasicBufferMgr.java:

Added 'bufferMap' : A map to store the map between the buffers and the blocks allocated to the buffers as a data member to this class

Changes to the pin() and pinNew() methods of this class:
Added lines of code to insert an entry into the 'bufferMap' map. Once a buffer is pinned, the mapping between this buffer and its block number is captured in the map.

Changes to the findExistingBuffer() method:
Updated this method to now return the buffer for a given block by a simple lookup on the map rather than traversing the bufferpool.

Added code to remove the block to buffer mapping in the GClock() method, when the block is about to get evicted from the bufferpool.

Task 3:

RecoveryMgr.java
Modified setInt and setString methods to check if the buffer is dirty and save it (if dirty) and write an updateRecord log.

Buffer.java
Added two new methods to save the contents of the buffer into the block (saveBlock()) and vice versa (restoreBlock).

UpdateRecord.java
Added a new class for the updateRecord log type which implements the LogRecord interface.

LogRecord.java
Added macro to handle update type record.

LogRecordIterator.java
Added the case to create object of UpdateRecord class to handle a log record of that type.