Computer Architecture
CS 325 - IND1
Department of Physics and Computer Science
Medgar Evers College
Exam 1: Part B

Instructions:

- **The exam requires completing a set of tasks by 12/18.**
- **The IAS computer architecture uses a bus composed of a 40-bit data bus, a 12-bit address bus, and a 3-bit control bus. Its main memory consists of 4,096 word-addressable locations, each storing a 40-bit word. Your task is to implement *Bus* and *Memory* classes for this IAS architecture and to create a C++ source file that updates a *Memory* object and its associated *Bus* object by executing control commands, in accordance with the specifications provided below.**
- **The classes must be defined in separate header files.**
- **All files can only include the libraries *iostream*, *fstream*, *string*, *sstream*, *stdexcept*, *set*, *cmath*, *iomanip*, 'Utility.h', 'MemCell.h', and 'Components.h'. Documentation of user-defined classes is provided in the directory.**
- **Cheating of any kind is prohibited and will not be tolerated.**
- **Violating and/or failing to follow any of the rules will result in an automatic zero (0) for the exam.**

TO ACKNOWLEDGE THAT YOU HAVE READ AND UNDERSTOOD THE INSTRUCTIONS ABOVE, PRINT YOUR NAME AND THE DATE ON YOUR SUBMISSIONS

Grading

Section	Maximum Points	Points Earned
1	3	
2	5	
3	2	
Total	10	

1. Create a header file that defines a class named *Bus* which publicly inherits from *BusCom* in '`Component.h`'. The class must

   - Provide all necessary special member functions, ensuring that the default constructor initializes the bus according to the IAS computer bus specifications.
   - Override the `read()`, `write()`, `length()`, `data()`, `addr()`, and `ctrl()` methods so that each behaves correctly for the IAS computer bus architecture.

   Hint: Make *Bus* a container class for a *Word* object.

2. Create a header file that defines a class named *Memory* which publicly inherits from *MemoryCom* in '`Component.h`'. The class must:

   - Provide all required special member functions, ensuring that the default constructor initializes the memory according to the IAS computer memory specifications.
   - Override the `process()` method so that it correctly executes each memory operation based on the control command:
     - 000 — full read
     - 001 — full write
     - 010 — left-half read
     - 011 — left-half write
     - 100 — right-half read
     - 101 — right-half write
   - Override the `manual()` method so that it returns a string describing all supported operations along with their associated control commands as illustrated in the previous task.

   Hint: Make a *Part* object for each component of the bus and the specified memory address.

3. Create a C++ source file that:

   - Instantiates a *Bus* object and a *Memory* object, and links them together.
   - Updates the *Bus* object and processes the *Memory* object to perform full, left-half, and right-half writes to different memory addresses using randomly generated data. After each operation, write the states of both the *Bus* and *Memory* objects to a file.
   - Update the *Bus* object and process the *Memory* object to perform full, left-half, and right-half reads from the memory address that was used in the full write in the previous step, ensuring that the bus data lines are cleared (set to zero) before each operation. After each operation, write the state of the *Bus* object to a file.