**Public Class CPU**

Properties ax and bx are integers. Both initialized as 0 in the constructor

Provide easy getters and setters for these properties

Implement a method to print the status of ax and bx

**Public Class Memory**

Property memoryContent is an integer array with size 10

In constructor, initialize the memoryContent with values equal to the indexes

Implement a toString method that returns a string of each memory address and its value

Implement a printMemoryStatus method that uses the toString method on this

Implement getters and setters for memoryContent

**Public Interface IExecutable**

Static IExecutable[] loadInstructionsFromFile(File file)

New File reader with parameter file

New buffered reader with parameter file reader

Number of instructions is the first line of the file to an int

An array of IExecutables with the size of the number of instructions is created

Using a for loop add the create instruction objects based on the instructions in the file and add them to the array

Close the buffered reader and return the array

**Abstract Class Instruction**

Declare properties String opcode, String arg1, and String arg2

Declare public constructor that takes the properties as parameters and sets them on the object

Declare getters and setters for the properties

**Public Class StoreInstruction**

Constructor calls the super constructor with the parameters of opcode, arg1, and arg2

Execute method takes arg1 as ax or bx and stores it in the memory location of arg2 then sets the memoryContent as the new array

**Public Class LoadInstruction**

Constructor calls the super constructor with the parameters of opcode, arg1, and arg2

Execute method takes arg1 as ax or bx and loads the memory location value of arg2 then sets the value of ax or bx on the CPU

**Public Class AddInstruction**

Constructor calls the super constructor with the parameters of opcode, arg1, and arg2

Execute method takes arg1 as ax, or bx. Arg2 is an int, ax, or bx. The method adds arg1 and arg2 and stores the sum in whatever arg1 is.

**Public Class SubInstruction**

Constructor calls the super constructor with the parameters of opcode, arg1, and arg2

Execute method takes arg1 as ax, or bx. Arg2 is an int, ax, or bx. The method subtracts arg2 from arg1 and stores the difference in whatever arg1 is.