CSE 350 File Systems Project I

The first part of this project requires that you implement a class that will be used to simulate a disk drive. The disk drive will have *numberofblocks* many blocks where each block has *blocksize* many bytes. The interface for the class **Sdisk** should include :

```
Class Sdisk
{
public :
Sdisk(string diskname);
Sdisk(string diskname, int numberofblocks, int blocksize);
int getblock(int blocknumber, string& buffer);
int putblock(int blocknumber, string buffer);
private :
string diskname;        // file name of software-disk
int numberofblocks;     // number of blocks on disk
int blocksize;          // block size in bytes
};
```

The Sdisk is a file of characters which we will manipulate as a raw hard disk drive. There are two files associated with the Sdisk. The first file is diskname.spc which is a small file containing the disk architecture - numberofblocks and blocksize. The second file is diskname.dat which contains the data on the Sdisk.

An explanation of the member functions follows :

- **Sdisk(diskname)**

- **Sdisk(diskname, numberofblocks, blocksize)**

  These constructors incorporate the creation of the disk with the "formatting" of the device. If the Sdisk exists, only the string diskname must be passed to the constructor. If the Sdisk does not exist, then the second constructor accepts the integer values *numberofblocks*, *blocksize*, a string *diskname* and creates an Sdisk (software-disk). Each constructor function will first check if the file *diskname.spc* exists. If the file exists, it is opened and the numberofblocks and blocksize are read. Then the file diskname.dat is opened and and treated as an Sdisk with *numberofblocks* many blocks of size *blocksize*. If the file does not exist, the constructor will create the file along with *diskname.dat* which contains *numberofblocks*blocksize* many characters. This file is logically divided up into *numberofblocks* many blocks where each block has *blocksize* many characters.

The text file will have the following structure :

Bytes 0 - blocksize-1	Block 0
Bytes blocksize - 2*blocksize-1	Block 1
Bytes 2*blocksize - 3*blocksize-1	Block 2
.	.
.	.
.	.
Bytes ?-?	Block *numberofblocks*-1

- **getblock(blocknumber,buffer)**

  retrieves block *blocknumber* from the disk and stores the data in the string *buffer*. It returns an error code of 1 if successful and 0 otherwise.

- **putblock(blocknumber,buffer)**

  writes the string *buffer* to block *blocknumber*. It returns an error code of 1 if successful and 0 otherwise.

IMPLEMENTATION GUIDELINES : It is essential that your software satisfies the specifications. These will be the only functions (in your system) which physically access the Sdisk. **NOTE** that you must also write drivers to test and demonstrate your program.