Attachment A:

#ifndef MYARRAY\_H

#define MYARRAY\_H

#include <iostream>

#include <fstream>

#include <cstdlib>

using namespace std;

class MyArray

{

private:

int size;

char\* ptr;

public:

/\*Default constructor\*/

MyArray();

/\*regurlar constructor that accepts as a parameter a char array or

\*string literal example "Hello World"\*/

MyArray(const char\*);

/\*Copy constructor\*/

MyArray(const MyArray&);

/\*Destructor\*/

~MyArray();

/\*Returns the size of the array. \*/

int getSize()const;

/\*Overloaded operators\*/

const MyArray& operator=(const MyArray&);

/\*Returns true if the arrays are the same. Otherwise it returns false.\*/

bool operator==(const MyArray&) const;

/\*Returns true if the arrays are not equal, false otherwise.\*/

bool operator!=(const MyArray& right)const;

/\*Provides bounds checking for the []\*/

char& operator[](int);

friend istream& operator>>(istream&, MyArray&);

friend ostream& operator<<(ostream&, const MyArray&);

friend MyArray operator+(const MyArray&, const MyArray&);

friend MyArray operator+(const MyArray&, const char);

friend MyArray operator+(const char, const MyArray&);

};

#endif

Attachment B

class Time

{

private:

int hour;

int minute;

int second;

public:

Time( );

Time(int, int, int);

int getHour() const;

int getMinute() const;

int getSecond() const;

void printTime();

bool equals(const Time &otherTime);

};

Matching list

1. Base Class
2. Derived Class
3. Dynamically Allocated memory
4. Inheritance
5. Override of Function
6. Virtual Function
7. Fixed
8. Setprecision
9. Template
10. Polymorphic
11. Stack
12. Queue
13. Operator Overloading
14. Static member variable
15. Setw
16. Showpoint