**CPP Problem Design Example**

|  |
| --- |
| **Subject: Operator overloading** |
| **Contributor: Wen-Kai Wang** |
| **Main testing concept:**   |  |  | | --- | --- | | **Basics** | **Functions** | | ■ C++ BASICS  □ FLOW OF CONTROL  □ FUNCTION BASICS  ■ PARAMETERS AND OVERLOADING  □ ARRAYS  ■ STRUCTURES AND CLASSES  □ CONSTRUCTORS AND OTHER TOOLS  ■ OPERATOR OVERLOADING, FRIENDS, AND REFERENCES  □ STRINGS  □ POINTERS AND DYNAMIC ARRAYS | □ SEPARATE COMPILATION AND NAMESPACES  □ STREAMS AND FILE I/O  □ RECURSION  □ INHERITANCE  □ POLYMORPHISM AND VIRTUAL FUNCTIONS  □ TEMPLATES  □ LINKED DATA STRUCTURES  □ EXCEPTION HANDLING  □ STANDARD TEMPLATE LIBRARY  □ PATTERNS AND UML | |
| **Description: Write a String Class (MixString.h/ MixString.cpp) and override the specified operator. The following unspecified “String”s refer to “MixString”.**  **++(prefix): Add a ‘+’ character in front of the original String.**  **++(postfix): Add a ‘+’ character after the original String.**  **[i]: Take the ith char (editable) in the String, if i is out of range(i could be less than zero), take the last char.**  **(n): Retrieve the String n consecutive times. For example, suppose String = “ABC”, return “ABCABCABC” if String(3) is called.**  **+, +=: Combine two Strings to form a new String. Suppose two Strings are a = "abc" and b = "12345". If c = a + b, then c should be "a1b2c345".**  **==: Determine if two Strings are identical.**  **>>: Support istream “>>” input.**  **<<: Support ostream “<<” output.**  **Provide get and set functions: .setString() and .getString()**  **How do we test your program: We will replace main.cpp that needs to successfully build and run (cannot change main.cpp at all). If successful, the TA will judge whether the program output is correct.**  **The MixString operator (+, +=, ==) needs to support MixString, std::string and character arrays.**  **Class should start with the definition provided below (below is an incomplete Class):**  **Class MixString**  **{**  **private:**  **std::string content;**  **public:**  **MixString(const std::string &str);**  **MixString(const MixString &mStr);**  **MixString(const char str[]);**  **void setString(const std::string &str);**  **std::string getString(void) const;**  **};**  **Input:**  Please see the sample  **Output:**  Please see the sample  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | sample.cpp sample.in | sample.out | |
| **□ Easy, Only basic programming syntax and structure are required.**  **■ Medium, Multiple programming grammars, and structures are required.**  **□ Hard, Need to use multiple program structures or complex data types.** |
| **Expected solving time:**  40 minutes |
| **Other notes:** |