**CPP Problem Design**

|  |
| --- |
| **Subject:** **Design Month Class** |
| **Contributor: 陳俊儒, 林承達, 廖宣瑋** |
| **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:**  Define a class called Month that is an abstract data type for a month. Your class will have one member variable of type int to represent a month (1 for January, 2 for February, and so forth).  Include all the following member functions:  1. a constructor to set the month using the first three letters in the name of the month as three arguments, a constructor to set the month using an integer as an argument (1 for January, 2 for February, and so forth), a default constructor.  2. an input function name inputInt that reads the month as an integer.  3. an input function name inputStr that reads the month as the first three letters in the name of the month.  4. an output function name outputInt that output the month as an integer.  5. an output function name outputStr that outputs the month as the first three letters in the name of the month.  6. a member function name nextMonth that returns the next month as a value of type Month. Embed your class definition in a test program.  Note that if month out of range, set the month to January.  **Input:**  Replace the main from main.cpp and enter the test data in input.txt.  **Output:**  See the Sample Output.  **Sample Input / Output：**   |  |  | | --- | --- | | Sample Input | Sample Output | | main1.in  sample.in | Month1 = 1 Jan  Month2 = 2 Feb  Month3 = 3 Mar  Month4 = 4 Apr  Month5 = 5 May  Month6 = 6 Jun | |
| **■Eazy,Only basic programming syntax and structure are required.**  **□ Medium,Multiple programming grammars and structures are required.**  **□ Hard,Need to use multiple program structures or more complex data types.** |
| **Expected solving time:**  30minutes |
| **Other notes:** |