EE2310 C++程式設計 HW5 (Ch 8) Due 5/16/2019

**A. True/False是非題: 15%**

是非題:

1. True/False: The amount of memory used by an array depends solely on the number of elements the array can hold.

F

2. True/False: If a C++ program contains the following array definition

int score[10];

the following statement would store 100 in the first array element

score[1] = 100;

F

3. True/False: The following statement initializes all five elements of the number array to 1.

int number[5] = {1};

F

4. True/False: To assign the entire contents of one array to another, you can use the assignment operator.

F

5. True/False: When you pass an array as an argument to a function, the function can modify the contents of the array.

T

6. True/False: A one-dimensional array can only store elements of a single data type, but a two-dimensional array can hold data of two different data types.

F

7. True/False: An element of a two-dimensional array is referenced by the array name and two subscripts, first the element row number and then the element column number.

T

8. True/False: When you create a vector it is unnecessary to specify how many elements it will hold because it will expand in size as you add new values to it.

T

9. True/False: The following statement adds a new element to a the department vector at index 25.

department.push\_back(25);

F

10. True/False: Assuming employee is an array of objects with a public member function named setHourlyWage, the following statement correctly calls this method for employee[2].

employee.setHourlyWage[2](20.00);

F

11. True/False: The amount of memory used by an array depends upon the array's data type and how many elements in the array currently have data stored in them.

F

12. True/False: Each individual element of an array can be accessed by the array name and an element number, called a subscript.

T

13. True/False: An individual array element can be processed or passed to a function just like a regular C++ variable.

T

14. True/False: After carrying out the following two statements, sales will have been created as a one-dimensional array that can hold 20 double values.

typedef salesArray double[20];

salesArray sales;

F

15. True/False: In C++ If you attempt to store more data in an array than it can hold, the compiler will issue an error.

F

**B. Choice選擇題: 15%**

1. To access an array element, use the array name and the element's \_\_\_\_\_\_\_\_.

A) name

B) data type

C) subscript

D) value

E) size declarator

C

2. The statement

int grades[ ] = { 100, 90, 99, 80 };

is an example of \_\_\_\_\_\_\_\_.

A) default arguments

B) an illegal array declaration

C) an illegal array initialization

D) implicit array sizing

E) data encapsulation

D

3. By using the same \_\_\_\_\_\_\_\_ you can build relationships between data stored in two or more arrays.

A) array name

B) data

C) subscript

D) arguments

E) data type

C

4. To step through a one-dimensional array, accessing the elements one by one, it would be most appropriate to use \_\_\_\_\_\_\_\_.

A) an infinite loop

B) a sentinel controlled loop

C) a for loop

D) a nested loop

E) none of above

C

5. When an array is passed to a function, it is actually \_\_\_\_\_\_\_\_ the array that is/are passed.

A) the starting memory address of

B) a copy of all the values in

C) the value stored in the first element of

D) the data type and size of

E) the data type and name of

A

6. An array of 10 integers named myArray can have its contents displayed with which of the following statements?

A) cout << myArray;

B) cout << myArray[];

C) cout << myArray[10];

D) cout << myArray[0-9];

E) None of the above

E

7. If you leave out the size declarator in an array declaration \_\_\_\_\_\_\_\_.

A) the array will contain no elements

B) the array size defaults to 100 elements

C) the value of each array element is set to a default value of 0

D) you must furnish an initialization list

E) the array cannot be created

D

8. The statement

typedef int oneDArray[20];

does which of the following?

A) creates an array of 20 integers

B) makes oneDArray a copy of another 20-integer array

C) makes oneDArray an alias for a data type that holds 20 integers

D) creates a one-dimensional integer array with all elements initialized to 20

E) None of the above

C

9. An array can store a group of values, but the values must be \_\_\_\_\_\_\_\_.

A) all the same data type.

B) declared at the time the array is created.

C) constants.

D) numeric, not characters or strings.

E) None of the above.

A

10. You can assign the contents of one array to another by using \_\_\_\_\_\_\_\_.

A) the assignment operator

B) the equality operator

C) both array names

D) A and C together

E) None of the above

E

11. When an array is passed to a function, it is actually \_\_\_\_\_\_\_\_ the array that is/are passed.

A) a copy of all the values in

B) the value stored in the first element of

C) the starting memory address of

D) the data type and size of

E) None of the above

C

12. To add up all the values in a two-dimensional array it would be best to use \_\_\_\_\_\_\_\_.

A) one for loop

B) two separate for loops

C) a nested for loop

D) no loop

E) one sentinel controlled loop

C

13. If the array defined as int myArray[20][10] is being passed to a function named displayArray, along with information on the number of rows and number of columns, which of the following function **calls** is correct?

A) displayArray(myArray, 20, 10);

B) displayArray(myArray[ ][ ], 20, 10);

C) displayArray(int myArray, 20, 10);

D) displayArray(myArray[20][10]);

E) None of the above

A

14. When a relationship is established between two or more arrays by using the same subscript to relate entries between the arrays, the arrays are called \_\_\_\_\_\_\_\_ arrays.

A) linked

B) indexed

C) parallel

D) brother

E) paired

C

15. If employee is an array of objects with a public member function named setHoursWorked, which of the following statements correctly calls that function for the employee object in array element 5?

A) employee.setHoursWorked[5] = 40;

B) employee[5].setHoursWorked = 40;

C) employee.setHoursWorked[5](40);

D) employee[5].setHoursWorked(40);

E) setHoursWorked(employee[5], 40);

D

**C. Programming程式題:70%**

**1. Roman Numeral Converter**

Write a program that displays the Roman numeral equivalent of any decimal number between 1 and 20 that the user enters. The roman numerals should be stored in an array of strings and the decimal number that the user enters should be used to locate the array element holding the Roman numeral equivalent. The program should have a loop that allows the user to continue entering numbers until an end sentinel of 0 is entered.

Input validation: Do not accept scores less than 0 or greater than 20.

**2. Monkey Business**

A local zoo wants to keep track of how many pounds of food each of its three monkeys eats each day during a typical week. Write a program that stores this information in a two-dimensional 3 × 7 array, where each row represents a different monkey and each column represents a different day of the week. The program should first have the user input the data for each monkey. Then it should create a report that includes the following information:

• Average amount of food eaten per day by the whole family of monkeys.

• The least amount of food eaten during the week by any one monkey.

• The greatest amount of food eaten during the week by any one monkey.

Input Validation: Do not accept negative numbers for pounds of food eaten.

**3. Lottery**

Write a program that simulates a lottery. The program should have an array of 5 integers named winningDigits, with a randomly generated number in the range of 0 through 9 for each element in the array. The program should ask the user to enter 5 digits and should store them in a second integer array named player or quit to quit the game. The program must compare the corresponding elements in the two arrays and count how many digits match. For example, the following shows the winningDigits array and the player array with sample numbers stored in each. There are two matching digits, elements 2 and 4.



Once the user has entered a set of numbers, the program should display the winning digits and the player’s digits and tell how many digits matched. Then the program repeats the game.

Input validation: Do not accept player inputs less than 0 or greater than 9.

**4. Rainfall Statistics**

Create a **Stats class** whose member data includes a string for storing the county (city) name, a string for storing staring year- month: yyyymm (e.g., 201901, 199812, 201003, etc.) an array capable of storing 30 double data values for consecutive monthly rainfall, and whose member functions include total, average, lowest, and highest functions for returning information about the data to the client program. The total and average functions return the total and average rainfalls of the rainfall values stored in the data member, respectively. The lowest and highest functions return the month-year and rainfall of the lowest and highest rainfall that fell in those months.

In addition to these functions, the Stats class should have a Boolean storeValue function that accepts a double value for rainfall amount from the client program and stores it in the array. Input validation: Do not accept rainfall amounts less than 0. It is the job of this function to keep track of how many values are currently in the array, so it will know where to put the next value it receives and will know how many values there are to process when it is carrying out its other functions. It is also the job of this function to make sure that no more than 30 values are accepted. If the storeValue function is able to successfully store the value sent to it, it should return true to the client program. However, if the client program tries to store a thirty-first value, the function should not store the value and should return false to the client program. The class should also have a displayReport function to print a summary rainfall report similar to the following:

January 2010 –April 2012 Rain Report for xxxxxxx County

Total rainfall in this period: 23.19 inches

Average monthly rainfall: 1.93 inches

The least rain fell in January, 2011 with 0.24 inches

The most rain fell in May, 2010 with 4.29 inches

The client program should create (using a constructor with initial values for starting yyyymm and county name) and use a Stats object to carry out the rainfall analysis. Then generate a report. Notice that the Stats object does no I/O. All input and output is done by the client program.

**5. Drink Machine Simulator**

Create a class that simulates and manages a soft drink machine. Information on each drink type should be stored in a structure that has data members to hold the drink name, the drink price, and the number of drinks of that type currently in the machine. The class should have an array of five of these structures, initialized with the following data.



The class should have two public member functions, displayChoices (which displays a menu of drink names and prices) and buyDrink (which handles a sale). The class should also have at least two private member functions, inputMoney, which is called by buyDrink to accept, validate, and return (to buyDrink) the amount of money input, and dailyReport, which is called by the destructor to report how many of each drink type remain in the machine at the end of the day and how much money was collected. You may want to use additional functions to make the program more modular.

The client program that uses the class should have a main processing loop which calls the displayChoices class member function and allows the patron to either pick a drink or quit the program. If the patron selects a drink, the buyDrink class member function is called to handle the actual sale. This function should be passed the patron’s drink choice.

Here is what the buyDrink function should do:



Input Validation: Only accept valid menu choices. Do not deliver a beverage if the money inserted is less than the price of the selected drink.

**6. Bin Manager Class**

Design and write an object-oriented program for managing inventory bins in a warehouse. To do this you will use two classes: InvBin and BinManager. The InvBin class holds information about a single bin. The BinManager class will own and manage an array of InvBin objects. Here is a skeleton of what the InvBin and BinManager class declarations should look like:





**Client Program**

Once you have created these two classes, write a menu-driven client program that uses a BinManager object to manage its warehouse bins. It should initialize it to use 9 of the bins, holding the following item descriptions and quantities. The bin index where the item will be stored is also show here.

The modular client program should have functions to display a menu, get and validate the user’s choice, and carry out the necessary activities to handle that choice. This includes adding items to a bin, removing items from a bin, and displaying a report of all bins. Think about what calls the displayReport client function will need to make to the BinManager object to create this report. When the user chooses the “Quit” option from the menu, the program should call its displayReport function one last time to display the final bin information. All I/O should be done in the client class. The BinManager class only accepts information, keeps the array of InvBin objects up to date, and returns information to the client program.

Input Validation in the BinManager class: The class functions should not accept numbers less than 1 for the number of parts being added or removed from a bin. They should also not allow the user to remove more items from a bin than it currently holds.

**7. Tic-Tac-Toe Game**

Write a modular program that allows two players to play a game of tic-tac-toe. Use a two dimensional char array with 3 rows and 3 columns as the game board. Each element of the array should be initialized with an asterisk (\*). The program should display the initial board configuration and then start a loop that does the following:

* Allow player 1 to select a location on the board for an X by entering a row and column number. Then redisplay the board with an X replacing the \* in the chosen location.
* If there is no winner yet and the board is not yet full, allow player 2 to select a location on the board for an O by entering a row and column number. Then redisplay the board with an O replacing the \* in the chosen location.

The loop should continue until a player has won or a tie has occurred, then display a message indicating who won, or reporting that a tie occurred.

* Player 1 wins when there are three Xs in a row, a column, or a diagonal on the game board.
* Player 2 wins when there are three Os in a row, a column, or a diagonal on the game board.
* A tie occurs when all of the locations on the board are full, but there is no winner.

Input Validation: Only allow legal moves to be entered. The row must be 1, 2, or 3. The column must be 1, 2, or 3. The (row, column) position entered must currently be empty (i.e., still have an asterisk in it).