**EE2310 C++程式設計HW 8 (Chapter 11 OOP) due: 6/11/2020**

**Part1, True/False是非題(20%, 1% for each)**

1. True/False: When a class declares an entire class as its friend, the friendship status is reciprocal. That is, each class's member functions have free access to the other's private members.
2. True/False: By default, when an object is assigned to another, each member of one object is copied to its counterpart in the other object.
3. True/False: When a class contains a pointer to dynamically allocated memory, it a good idea to have the class overload the assignment operator.
4. True/False: If you overload the prefix ++ operator, the postfix ++ operator is automatically overloaded.
5. True/False: A public data member may be declared a friend of a private function.
6. True/False: C++ permits you to overload the sizeof operator.
7. True/False: A static member variable can be used when there are no objects of the class in existence.
8. True/False: When you overload the << operator, you must also overload the >> operator.
9. True/False: You can overload the conditional operator to make it function as an unconditional operator.
10. True/False: In an inheritance situation, you can't pass arguments to a base class constructor.
11. True/False: A derived class may become a base class, if another class is derived from it.
12. True/False: A member function of a derived class may not have the same name as a member function of a base class.
13. True/False: A derived class may not have any classes derived from it.
14. True/False: Some C++ operators cannot be overloaded by the programmer.
15. True/False: The this pointer is a special built-in pointer that is automatically passed as a hidden argument to all instance member functions.
16. True/False: In C++, if you overload the < operator, you must also overload the > operator.
17. True/False: When you overload an operator, you cannot change the number of operands taken by the operator.
18. True/False: The this pointer is automatically passed to static member functions of a class.
19. True/False: A constructor that takes a single parameter of a type other than its class is called a convert constructor.
20. True/False: The base class access specification can be viewed as a filter that base class members must pass through when becoming inherited members of a derived class.

**Part2 Choice選擇題(30%, 1% for each)**

1. Each object of a class has its own copy of the class's \_\_\_\_\_\_\_\_.

A) static member variables

B) instance member variables

C) static member functions

D) All of the above

E) None of the above

1. A(n) \_\_\_\_\_\_\_\_ member variable may be accessed before any objects of the class have been declared.

A) private

B) public

C) inline

D) static

E) None of the above

1. C++ requires that a copy constructor's parameter be a(n) \_\_\_\_\_\_\_\_.

A) integer data type

B) floating-point data type

C) pointer variable

D) reference to an object

E) None of the above

1. C++ allows you to overload \_\_\_\_\_\_\_\_.

A) compiler errors

B) preprocessor directives

C) operators and functions

D) undefined variables

E) None of the above

1. When a class contains a pointer to dynamically allocated memory, it is a good idea to equip the class with a \_\_\_\_\_\_\_\_.

A) dynamically allocated constructor

B) copy constructor

C) static constructor and an overloaded comparison operator

D) inline constructor

E) None of the above

1. A good reason for overloading an operator is to enable it to \_\_\_\_\_\_\_\_.

A) outperform its C language counterparts

B) be used with types defined by the programmer

C) operate on more operands than in its standard definition

D) operate on no operands

E) None of the above

1. The process of having a class contain an instance of another class is known as \_\_\_\_\_\_\_\_.

A) object overloading

B) operator overloading

C) object composition

D) dynamic composition

E) None of the above

1. If you do not furnish a(n) \_\_\_\_\_\_\_\_, an automatic memberwise copy will be performed when one object is assigned to another object.

A) overloaded constructor function

B) overloaded assignment operator

C) default constructor function

D) overloaded copy operator

E) None of the above

1. It is a good idea to make a copy constructor's parameters \_\_\_\_\_\_\_\_ by specifying the \_\_\_\_\_\_\_\_ keyword in the parameter list.

A) inline, inline

B) static, static

C) constant, const

D) global, glob

E) None of the above

1. \_\_\_\_\_\_\_\_ allows us to create new classes based on existing classes.

A) Polymorphism

B) Inheritance

C) Function overloading

D) The copy constructor

E) None of the above

1. When you derive a class from an existing class, you \_\_\_\_\_\_\_\_.

A) can add new data, but cannot add new functions

B) can add new functions, but cannot add new data

C) can add both new data and new functions

D) must add both new data and new functions

E) None of the above

1. \_\_\_\_\_\_\_\_ members of a base class are never accessible to a derived class.

A) Public

B) Private

C) Protected

D) A, B, and C

E) None of the above

1. The \_\_\_\_\_\_\_\_ class constructor is called before the \_\_\_\_\_\_\_\_ class constructor.

A) base, derived

B) derived, base

C) public, private

D) private, public

E) None of the above

1. Arguments are passed to the base class destructor function by the \_\_\_\_\_\_\_\_ class \_\_\_\_\_\_\_\_ function.

A) derived, constructor

B) derived, destructor

C) base, constructor

D) base, destructor

E) None of the above

1. The statement class Car:private Vehicle allows the \_\_\_\_\_\_\_\_ members of the Car class to access \_\_\_\_\_\_\_\_ members of the Vehicle class.

A) private, private

B) public, private

C) protected, private

D) public, protected

E) None of the above

1. A member function that is declared \_\_\_\_\_\_\_\_ cannot use the this pointer.

A) private

B) public

C) static

D) inline

E) None of the above

1. A(n) \_\_\_\_\_\_\_\_ is a special function that is called whenever a new object is created and initialized with data from another object of the same class.

A) destructor

B) static function

C) copy constructor

D) assignment function

E) None of the above

1. When you redefine the way a standard operator works when it is used with class objects, you have \_\_\_\_\_\_\_\_ the operator.

A) reassigned

B) reformatted

C) overloaded

D) overwhelmed

E) None of the above

1. The \_\_\_\_\_\_\_\_ is a special built-in pointer that is available to a class's instance member functions.

A) overloaded -> operator

B) this pointer

C) &constructor pointer

D) ~destructor \*ptr

E) None of the above

1. Object composition is useful for creating a \_\_\_\_\_\_\_\_ relationship between classes.

A) friend

B) static

C) has-a

D) conditional

E) None of the above

1. A(n) \_\_\_\_\_\_\_\_ operator can work with programmer-defined data types.

A) inline

B) unconditional

C) overloaded

D) undefined

E) None of the above

1. When overloading the operator ++, \_\_\_\_\_\_\_\_\_\_\_\_\_\_ is used to distinguish preincrement from postincrement.

A) a parameterless data type

B) the keyword void

C) a dummy integer parameter

D) the placehoder [ ]

E) None of the above

1. To dereference an object pointer and access one of the object's members, use the \_\_\_\_\_\_\_\_.

A) -> operator

B) <> operator

C) dot operator

D) & operator

E) None of the above

1. In an inheritance situation, the new class that you create from an existing class is known as the \_\_\_\_\_\_\_\_.

A) derived class

B) inheritee

C) child class

D) both A and C

E) None of the above

1. The base class's \_\_\_\_\_\_\_\_ affects the way its members are inherited by the derived class.

A) name

B) return data type

C) access specification

D) both A and B

E) None of the above

1. Protected members of a base class are like \_\_\_\_\_\_\_\_, with the exception that they may be accessed by derived classes.

A) constructor functions

B) static members

C) private members

D) public members

E) None of the above

1. The \_\_\_\_\_\_\_\_ class destructor is called before the \_\_\_\_\_\_\_\_ class destructor.

A) base, derived

B) derived, base

C) public, private

D) private, public

E) None of the above

1. \_\_\_\_\_\_\_\_ is commonly used to extend a class, or to give it additional capabilities.

A) Inheritance

B) Privacy

C) The constructor

D) The destructor

E) None of the above

1. In the statement class Car:protected Vehicle**,** which is the derived class?

A) Car

B) Vehicle

C) protected

D) The answer cannot be determined from the information given

E) None of the above

1. To overload the + operator, you would write a function called \_\_\_\_\_\_\_\_.

A) overload +

B) operator +

C) function +

D) operator.overload(+)

E) None of the above

**Part3 Coding程式題(50%)**

**1. Check Writing (10%)**

Design a class Numbers that can be used to translate whole dollar amounts in the range 0 through 9999 into an English description of the number. For example, the number 713would be translated into the string *seven hundred thirteen*, and 8203 would be translated into *eight thousand two hundred three*.

The class should have a single integer member variable

int number;

and a collection of static string members that specify how to translate key dollar amounts into the desired format. For example, you might use static strings such as

string lessThan20[ ] =

{"zero", "one", …, "eighteen", "nineteen" };

string hundred = "hundred";

string thousand = "thousand";

The class should have a constructor that accepts a nonnegative integer and uses it to initialize the Numbers object. It should have a member function print() that prints the English description of the Numbers object. Demonstrate the class by writing a main program that asks the user to enter a number in the proper range and then prints out its English description.

**2. Day of the Year (10%)**

Assuming that a year has 365 days, write a class named DayOfYear that takes an integer representing a day of the year and translates it to a string consisting of the month followed by day of the month. For example,

Day 2 would be *January 2*

Day 32 would be *February 1*

Day 365 would be *December 31*.

The constructor for the class should take as parameter an integer representing the day of the year, and the class should have a member function print() that prints the day in the month-day format. The class should have an integer member variable to represent the day, and should have static member variables of type string to assist in the translation from the integer format to the month–day format.

Test your class by inputting various integers representing days and printing out their representation in the month–day format.

**3. Palindrome Testing (10%)**

A palindrome is a string that reads the same backward as forward. For example, the words *mom*, *dad*, *madam* and *radar* are all palindromes. Write a class Pstring that is derived from the STL string class. The Pstring class adds a member function

bool isPalindrome( )

that determines whether the string is a palindrome. Include a constructor that takes an STL string object as parameter and passes it to the string base class constructor. Test your class by having a main program that asks the user to enter a string. The program uses the string to initialize a Pstring object and then calls isPalindrome() to determine whether the string entered is a palindrome.

You may find it useful to use the subscript operator [] of the string class: if str is a string object and k is an integer, then str[k] returns the character at position k in the string.

**4. Corporate Sales (10%)**

A corporation has six divisions, each responsible for sales to different geographic locations. Design a DivSales class that keeps sales data for a division, with the following members:

**•** An array with four elements for holding four quarters of sales figures for the division

**•** A private static variable for holding the total corporate sales for all divisions for the entire year.

**•** A member function that takes four arguments, each assumed to be the sales for a quarter. The value of the arguments should be copied into the array that holds the sales data. The total of the four arguments should be added to the static variable that holds the total yearly corporate sales.

**•** A function that takes an integer argument within the range of 0 to 3. The argument is to be used as a subscript into the division quarterly sales array. The function should return the value of the array element with that subscript.

Write a program that creates an array of six DivSales objects. The program should ask the user to enter the sales for four quarters for each division. After the data is entered, the program should display a table showing the division sales for each quarter. The program should then display the total corporate sales for the year.

**5. Rational Arithmetic** **(10%)**

A *rational number* is a quotient of two integers. For example, 12/5, 12/–4, –3/4, and 4/6 are all rational numbers. A rational number is said to be in *reduced form* if its denominator is positive and its numerator and denominator have no common divisor other than 1. For example, the reduced forms of the rational numbers given above are 12/5, –3/1, –3/4, and 2/3.

Write a class called Rational with a constructor Rational(int, int) that takes two integers, a numerator and a denominator, and stores those two values in reduced form in corresponding private members. The class should have a private member function void reduce() that is used to accomplish the transformation to reduced form. The class should have an **overloaded** **insertion operator <<** that will be used for output of objects of the class. The class also should have **overloaded operators +, -, \***, and **/** to be used for addition, subtraction, multiplication, and division. Test the class by reading and processing from the keyboard (or from a file) a series of rational expressions such as

2 / 3 + 2 / 8

2 / 3 \* – 2 / 8

2 / 3 – 2/ 8

2 / 3 / 2 / 8

To facilitate parsing of the input, you may assume that numbers and arithmetic operators are separated by whitespace.