SE 116 – LAB#2 2014 – 2015 SPRING

Aim: Practice on declaration and definition of classes, understanding the concept of encapsulation, using constructors and destructors.

PART 1.

Assume you are generating a character for a medieval themed computer game. Your character is a Swordsman. He/she has sword and wears armor. Create a **Swordsman** class with the following attributes: ***name (as string), swordID (as int), armorID (as int), swordType (as string), armorType (as string).***

a) Your class attributes (data members) must be **private**. The class methods (member functions) you are going to use to access these attributes must be **public**.

b) Define appropriate **set** functions for *name*, *swordID* and *armorID*.
There is a condition: *swordID* and *armorID* attributes accept values within the range [1, 3]. For those ID values, you need to assign the corresponding type as following:

swordID	corresponding *swordType*
1	ShortSword
2	LongSword
3	Greatsword

armored	corresponding *armorType*
1	LeatherArmor
2	ChainArmor
3	PlateArmor

Example: If the user sets the swordID as 2, the program must set the corresponding swordType to "LongSword" automatically.
If the swordID/armorID value is attempted to set with any value out of the range [1, 3], it must be replaced by 0, which means that the character is unarmed/unarmored.

c) Define appropriate **get** functions for *name, swordType* and *armorType* attributes.

d) Declare *checkStatus()* function and define it outside of the class declaration. If the swordsman is unarmed and/or unarmored this function will ask user to set a valid type of weapon and/or armor. If the swordsman has both weapon and armor, then the function will display the status of the swordsman as shown in the following sample output:
Example:

```
Name: Arthur
Sword Type: GreatSword
Armor Type: PlateArmor
```

e) In **main** function, instantiate (create) two objects from Swordsman class. After entering name, select sword and armor for the objects and finally display the status of the objects.

PART 2.

a) Add **constructor** and **destructor** functions to your class. You are supposed to create two types of constructors as default and parameterized.

- Swordsman()   // default constructor
- Swordsman(string name, int swordID, int armorID)   // parameterized constructor
- ~Swordsman() // destructor

When any constructor is called, the following information will be displayed on screen:
"Here comes a new challenger!"

When the destructor is called, the following information will be displayed on screen:
"Arthur has just been killed!" ("Arthur" is the name of the corresponding object.)

b) In main function, create two Swordsman objects, one with the parameterized constructor and one with the default constructor. For the object instantiated using default constructor, user is going to select name, sword and armor later. The other object is going to be set with appropriate name, sword and armor at construction time. Finally, display the status of both objects.

c) In main function, define two Swordsman pointers and make them point to different Swordsman objects constructed dynamically using operator **new**. For the object instantiated using default constructor, user is going to select name, sword and armor later. The other object is going to be set with appropriate name, sword and armor at construction time. Print the status of both objects and then **delete** them via their pointers.