**10802 CPP Final Exam**

|  |
| --- |
| **Contributor︰KWY** |
| **Subject：Animal Kingdom** |
| **Main testing concept：Inheritance**   |  |  | | --- | --- | | **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：**  Suppose that you are creating an animal kingdom with a variety of birds and mammals. The animals in the kingdom can do three action: make a sound, do the trick and introduce itself. To accomplish this task, you should complete the following requirements in “Animal.h”:   1. Define an Animal class with a string member name and four methods: makeASound() , doTrick(), introduced(), and myKind()which should return 0. All member function should be public. 2. The Animal class should has the following functions:    1. A constructor Animal(string inputName), which set the animal’s name.    2. A function of setName(string inputName), which modify the animal’s name.    3. A function of getName(), which return the animal’s name.    4. Overload operator <<,which output "My name is " + getName() + ". I am a " + introduced() +" of " + myKind() " and " + doTrick() + ", also I make sound like \"" + makeASound() + "\"". 3. Define a Bird class which is derived from Animal whose    1. doTrick() replies "I can fly".    2. myKind()replies "Bird".    3. a function history()prints "I'm a flying dinosaur actually. (fluttering!)\n". 4. Define a Mammal class which is derived from Animal whose    1. doTrick replies "I can jump".    2. myKind()replies "Mammal".    3. a function of run()prints "As a mammal, I can run. (sweating...)\n". 5. Define a Cat, Pig and Fox classes, respectively, which are derived from Mammal whose    1. makeASound() replies the sound of the animal.    2. introduced() replies its class name. 6. Define a Crow and Duck classes which are derived from Bird whose    1. makeASound()replies the sound of the animal.    2. introduced() replies its class name.   Note:   * Various animals make the following sounds, respectively:   Cat: “meow”  Pig: “oink”  Fox: “Ring-ding-ding-ding-dingeringeding!”  Crow: “cah”  Duck: “quack”  **Input：**  No inputs.  Note:   1. The main() function in your submission will be replaced when judging. 2. You can use the main() function in “Other Notes” to test your program. 3. All member function should be public.   **Output：**  You should output the result of executing the given main along with your implemented classes.  **Sample Input / Output :**   |  |  | | --- | --- | | **Sample Input** | **Sample Output** | |  | C  As a mammal, I can run. (sweating...)  I'm a flying dinosaur actually. (fluttering!)  My name is Gelatoni. I am a Cat of Mammal and I can jump, also I make sound like "meow"  My name is Jim. I am a Crow of Mammal and I can fly, also I make sound like "cah"  My name is Donald. I am Duck of Bird and I can fly, also I make sound like "quack" | |
| **■** **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:**  20 minutes |
| **LTE:**  1Sec |
| **Other notes:**  #include <iostream>  #include "Animal.h"  using namespace std;  int main() {  Cat cat("C");  Crow crow("Jim");  Duck duck("Donald");  cout << cat.getName() << endl;  cat.run();  duck.history();  cat.setName("Gelatoni");  Animal\* animals[] = { &cat,&crow,&duck };  for (int i = 0; i < 3; i++){  cout << \*animals[i] << endl;  }  return 0;  } |