|  |  |
| --- | --- |
| Edith Cowan University School of Science |  |

Workshop 10

Understanding Inheritance

**Related Objectives:**

* Learn about inheritance and its benefits
* Create a derived class
* Learn about restrictions imposed with inheritance
* Choose a class access specifier
* Override inherited access
* Override and overload parent class functions within a child class

**Activity:**

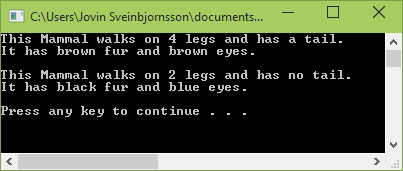
1. Use the attached HR System project. This defines two classes, HRSystem and Employee.
   1. The HRSystem class holds a vector of Employee objects. Modify the class so that it holds a vector of Employee object references and still works as intended.
   2. Create two child classes of Employee, Worker and Manager. Modify the isManager and isWorker methods so that they work based on class type rather than the value of the position variable. I.e. Both return false for an Employee object, isManager returns true from a Manager object and isWorker returns true from a Worker object. Modify the loading of Employee records so that for each record an object of the appropriate class is stored in the vector. Test to make sure the right names are printed for the list of workers and managers.
2. In the below tasks, using separate .CPP and .H files create a program which makes use of the classes provided. You will be given the Header file for each of the classes required, the task involves working with classes to eventually inherit down to a specialised class of animal called a “Liger”.

*Note: There are many ways to achieve the output, the solution file makes use of the ?: operator, this is a shorthand ‘if’ statement and can be used like so:*

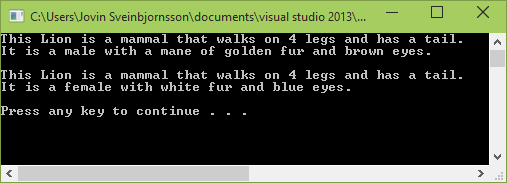
*<boolean value> ? <statement if true> : <statement if false>;*

**Using the below base class called Mammal, complete the following tasks**

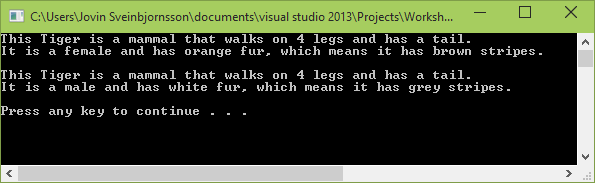
* 1. Write and compile the implementation for the given class in the attached file “Mammal.h”. You will be required to create the main() method that can provide the output as evidenced in the screenshot below.



* 1. Using the “Lion.h” header file, create the implementation of the Lion class and then modify your main() method so it can output the information in the screenshot below, in addition to the previous results.



* 1. Using the “Tiger.h” header file, create the implementation of the Tiger class and then modify your main() method so it can output the information in the screenshot below, in addition to the previous results.



* 1. Using the “Liger.h” header file, create the implementation of the Liger class so and then modify your main() method so it can output the information in the screenshot below, in addition to the previous results.

*Note: You will need to modify your other Classes in order to have appropriate constructors for this new Class. You will also need to choose an appropriate Class Access Specifier for each one as well.*

