**Comp2522 Quiz#2**

class Foo {

private Bar b;

private int x;

class Bar {

public void setX(int value){

x = value;//error?

}

}

}

1.Is there an error at the line indicated? Why or why not? <2 marks>

**No error – Inner classes have access to all it’s outer class members including private**

public static void main(String[] args){

Foo.Bar bar = new Bar();//error!

}

1. Examine the main() method above to create a Bar object from the definition provided in Q1 above. The compiler say the line in main is in error – Why? How do you fix it? Hint: Bar is an inner class, what must you have to create one? Think of Egg and Yolk. <1 marks>

**Must have an outer class object for the inner class object to belong to therefore you must first create the outer class object and then create the inner class object inside it.**

interface A {

void foo();

}

class B implements A{

public void foo(){}

}

class Main {

public static void main(String[] args){

A a = new B();//ERROR????

}

}

1. Is there an error in the above code? Why or why not? <1 marks>

**No error, you can make references from an Interface type.**

class Animal {

public void draw(){

System.out.println(“animal”);

}

}

class Dog {

public void draw(){

System.out.println(“dog”);

}

public void bark(){

System.out.println(“woof”);

}

}

class Cat {

public void draw(){

System.out.println(“cat”);

}

public void purr(){

System.out.println(“prrrr”);

}

}

class CatDog {}

//NO CHANGES TO CODE IN CLASS PLAY

public class Play {

public void guard(Dog d){

d.bark();

}

public void pet(Cat c){

c.purr();

}

public void make(Animal a){

a.draw();

}

public static void main(String[] args){

CatDog a = new CatDog();

Play p = new Play();

p.guard(a);

p.pet(a);

p.make(a);

}

}

output:

woof

prrrr

catdog

1. Modify the classes Animal, Dog, Cat as necessary and define **CatDog** to have Play function as desired. Note Cat is-NOT a Dog, a Dog is-NOT a Cat. HINT: Java does not allow multiple inheritance however it gives you something that allows a class to be many different types! Use that ability!! You cannot change class Play in any way. <6 marks>

Interface Dog { <1 mark>

Draw();

Bark();

}

Interface Cat { <1 mark>

Draw();

Purr();

}

Class CatDog extends Animal implements Dog, Cat{ <1 mark>

Public void draw(){ <1 mark>

…print(“CatDog”);

}

Public void bark(){ <1 mark>

…print(“woof”);

}

Public void purr(){ <1 mark>

….print(“prrr”);

}

}