Data Structures – ITCS 2214  
Homework #2  
**Due via Moodle the day before our next class at 11:55pm**

**Name: Kheang Tang**

**Short Answer (5 pts each)**

1. In regards to fields and methods, what happens when one class inherits from another?

**When one class is inherits from another class, fields and methods can be reuse in subclass**

1. If a reference variable is polymorphic, what types of objects can it refer to?

**When reference variable is polymorphic, it can be refer to it own class object or an object of the subclass that inherit from it class**

1. What does the super reference refer to?

**Super reference refer to parent’s constructor to set up the data in the child class**

1. If a class overrides a method that it inherited, and an object is created from that child class, which version of the method is called when that object calls it (child or parent)?

**Parent class**

1. How does the “is-a” relationship relate to inheritance?

**Inheritance is –a relationship, example: Apple is-a fruit**

**Interpret / Write Code (8 pts each)**

1. public class Freshman extends Student

In the code above, which is the parent class?

**Student is the parent class**

1. Place the following classes in a logical hierarchy. Draw boxes with the class name to represent the class, and lines connecting the classes to represent inheritance. You will use this hierarchy to answer questions #3 – 5.

Animal, Dog, Cat, Husky, Bird, Cardinal

**Animal**

Dog

**Cat**

Bird

Husky

Cardinal

1. Write class headers for each of the six classes.

**Public class Animal{}**

**Public class Cat extends Animal{}**

**Public class Dog extends Animal{}**

**Public class Husky extends Dog{}**

**Public class Bird extends Animal{}**

**Public class Cardinal extends Bird{}**

1. Which class would be referenced if the super reference was used in the constructor for the Cat class?

**Animal class**

1. If an object was created from the Dog class, what types of objects could that reference variable point to?

**Dog class**

1. Write constructors for the Animal, Bird, and Cardinal classes. You must make all fields private, and assume that all of the information is initially passed to the Cardinal constructor.

Animal – typeOfMovement(Str), moveSpeed(double)

Bird – canFly(bool), wingSpan(int)

Cardinal – shadeOfRed(Str)

**Public class Animal{**

**Private String typeOfMovement;**

**Private double moveSpeed;**

**Public Animal(String typeOfMovement, Double moveSpeed)**

**{**

**this.typeOfMovement=typeOfMovement;**

**this.moveSpeed=moveSpeed;**

**}**

**}**

**Public class Bird extends Animal{**

**Private boolean canFly;**

**Private int wingSpan;**

**Public Bird( String typeOfMovement, Double moveSpeed, boolean canFly, int wingSpan)**

**{**

**super(typeOfMovement, moveSpeed)**

**this.canFly=canFly;**

**this.wingSpan=wingSpan;**

**}**

**}**

**Public class Cardinal extends Bird{**

**Private String shadeOfRed;**

**Public Cardinal(String typeOfMovement, Double moveSpeed, boolean canFly, int wingspan, String shadeOfRed)**

**{**

**super(typeOfMovement, moveSpeed, canFly, wingspan)**

**this.shadOfRed=shadeOfRed;**

**}**

**}**

1. Given the following toString() method of the Animal class, write a toString() method for the Dog class. Assume that all fields are private, and that the Dog class has the following fields: pureBlood(bool), furColor(Str), age(int).

public String toString()

{

return moveType + “ “ + moveSpeed;

}

**Public class Dog extends Animal{**

**Private boolean pureblood;**

**Private String furColor;**

**Private int age;**

**Public Dog(String typeOfMovement, Double moveSpeed, boolean pureblood, String furColor, int age)**

**{**

**super(typeOfMovement. moveSpeed)**

**this.pureblood=pureblood;**

**this.furColor=furColor;**

**this.age=age;**

**}**

**public String toString()**

**{**

**return super.toString() + “pureblood” +pureblood+ “furcolor”+furcolor+”age”+age;**

**}**