CS 1331 - Polymorphism WS

NOTE: THIS IS NOT A PRACTICE EXAM: It is not meant to in any way reflect the contents or format of Exam 3. This is a practice worksheet and is not meant to be the sole preparation for the exam. Questions on this worksheet are meant to give students a better understanding of course concepts for homeworks as well as future exams.

Consider the hierarchy given below:

```
public class Animal {
	protected String name;
	protected String sound;
	protected String move;

	public Animal(String name, String sound, String move) {
		this.name = name;
		this.sound = sound;
		this.move = move;
	}

	public Animal() {
		this("animal", "sound", "move");
	}

	@Override
	public String toString() {
		return "Animal " + name + ", sound: " + sound + ", move:
		" + move;
	}

	public String myName() {
		return "I am an Animal";
	}

	public static String breath() {
		return "I am forever breathing";
	}
}
```

```
public class Cat extends Animal {
	public Cat() {
		super("Cat", "meow", "prowl");
	}

	@Override
	public String toString() {
		return "Cat " + name + ", sound: " + sound + ", move: " +
		move;
	}

	@Override
	public String myName() {
		return "I am a Cat named " + name;
	}
}


public class Dog extends Animal {
	public Dog() {
		super("Dog", "bark", "run");
	}

	@Override
	public String toString() {
		return "Dog " + name + ", sound: " + sound + ", move: " +
		move;
	}

	@Override
	public String myName() {
		return "I am a Dog named " + name;
	}

	public static String breath() {
		return "I am always out of breath";
	}

	public String bark() {
		return "I am dog, I " + sound;
	}
}
```

Determine if the block of code in each question will compile. If the code DOESN'T compile, write "Does not compile". If the code DOES compile, write "Compiles" along with what will be returned by the method call. If an error occurs at runtime, also write "Compiles. Error occurs". You can assume all the blocks of code are independent and do not affect one another.

1.

a)
```
Animal a = new Dog();
a.myName();
```

b)
```
Object d = new Dog();
d.myName();
```

c)
```
Object o2 = new Dog();
((Animal) o2).myName();
```

d)
```
Dog d = new Animal();
d.myName();
```

e)
```
Animal d = new Dog();
d.bark();
```

f)
```
Object o2 = new Dog();
((Cat) o2).myName();
```

Consider the classes below:

```
public class Person {
      int age;
      String name;
      public void where() {}
}


public class Student extends Person {
      double gpa;
      public void where() {}
      public void studentMethod(Student aStudent) {}
}


public class CollegeStudent extends Student {
      public void where() {}
}


public class HighSchoolStudent extends Student {

    public void where() {}

1.  @Override
2.  public boolean equals(HighSchoolStudent o) {
3.      if (o.equals(this)) {return true;}
4.      if (!(o instanceof HighSchoolStudent)) {return false;}
5.      Student other = (HighSchoolStudent) o;
6.      return other.school.equals(this.school) &&
        other.gpa == this.gpa;
    }
}
```

2.

a) For the following table of commands to be executed in a Driver class, state if each of these commands would compile and if they run, which class's method runs without crashing?

Command	***Compiles?***	***Runs?***
Student student1 = new Student();		
Student student2 = new Person();		
Person person1 = new Student();		
person1.where();		
person1.studentMethod(student1);		
((Student)person1).studentMethod(new CollegeStudent());		
((Student)person1).studentMethod(student1);		
(Student)person1.studentMethod(student1);		
CollegeStudent collegeStudent0 = new CollegeStudent();		
HighSchoolStudent castHS = (HighSchoolStudent) collegeStudent0;		
CollegeStudent castCS = (CollegeStudent) student1;		

b) When casting an Object, when are we guaranteed that the casting will compile? When will a casting not only compile but also run without crashing the program?

c) Look at the equals method defined in the HighSchoolStudent class. Three mistakes were made in the implementation. Some of them will be caught at compile time, some of them are logical bugs. List the line where mistakes are found, and provide a short description of what's incorrect.

Line ____:

Line ____:

Line ____:

Consider the hierarchy given below:

```
public class Animal {
      public void makeNoise() {
            System.out.println("Animal noise!");
      }

      public void eat() {
            System.out.println("Nom");
      }
}


public class Bird extends Animal {
      public void makeNoise() {
            System.out.println("Chirp chirp!");
      }

      public void fly() {
            System.out.println("I’m flying!");
      }
}


public class Fish extends Animal {

}
```

Determine if the block of code in each question will compile. If the code DOESN'T compile, write "Does not compile". If the code DOES compile, write "Compiles" along with what will be returned by the method call if there is a method call. If an error occurs at runtime, also write "Compiles. Error occurs". You can assume all the blocks of code are independent and do not affect one another.

3.

a)
```
Animal a = new Bird();
a.makeNoise();
```

b)
```
Animal a = new Bird();
a.fly();
```

c)
```
Animal a = new Bird();
a.eat();
```

d)
```
Bird b = new Bird();
Animal a = (Animal) b;
```

e)
```
Animal a = new Animal();
Fish f = (Fish) a;
```

f)
```
Animal a = new Fish();
Fish f = (Fish) a;
```

g)
```
Bird b = new Bird();
Fish f = (Fish) b;
```

h)
```
Fish f = new Fish();
Animal a = (Animal) f;
```

i)
```
Animal a = new Bird();
Bird b = (Bird) a;
```

j)
```
Animal a = new Animal();
Bird b = (Bird) a;
```

k)
```
Animal a = new Fish();
a = (Bird) a;
```

l)
```
Bird b = new Bird();
((Animal) b).makeNoise();
```

m)
```
Animal a = new Bird();
((Bird) a).fly();
```

n)
```
Animal a = new Bird();
(Bird) a.fly();
```

o)
```
Animal a = new Bird();
((Bird) a).eat();
```

p)
```
Animal a = new Bird();
((Bird) a).makeNoise();
```