Montgomery College, CMSC 203
Worksheet 1
Module 16

Objectives
- Inheritance
- Superclass constructor
- Overriding methods

Concept Questions
1) The relationship between a superclass and an inherited class is called an _____ relationship

Answer: IS A

2) If a class inherits from another class it inherits:

A) Methods
B) Variables
C) Both A and B
D) None of the above

Answer: C

3) A _____ is a general class, and a _____ is a more specialized class which inherits from the general class.

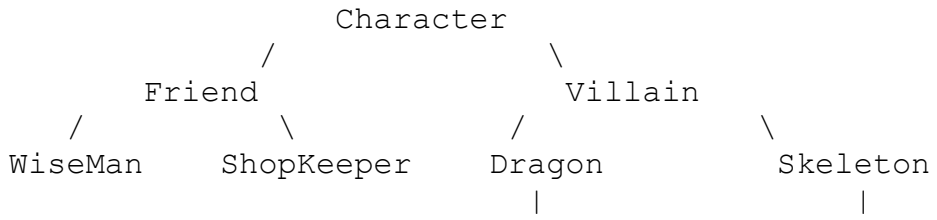
Answer: Superclass, subclass

4) Write a header of a class `Circle` which inherits a superclass `Shape`.

Answer: `public class Circle extends Shape`

5) If a superclass method has this access modifier it CANNOT be accessed by a child class:
A) `private`
B) `public`
C) `protected`
D) Both B and C
Answer: A

6) Consider the following inheritance hierarchy that is used in a video game.



`FlyingDragon` `EliteSkeleton`

Which of the following declarations and initializations will NOT cause a compiler error?

A) `Character c = new FlyingDragon();`
B) `FlyingDragon f = new Character();`
C) `Dragon d = new Villain();`
D) `Villain v = new Skeleton();`
E) `Dragon d = new ShopKeeper();`

Answer: A and D

7) Suppose that `Horse` is a subclass of `Animal`, and neither class is abstract. Which of the following is an invalid declaration and initialization?

A) `Horse h = new Horse();`
B) `Horse h = new Animal();`
C) `Animal a = new Animal();`
D) `Animal a = new Horse();`
E) all of the above are valid

Answer: B

8) Let Dog be a subclass of Animal, and suppose Animal has a method called speak() that is overridden in the Dog class. Consider the following code.

```
Animal spot = new Dog();
spot.speak();
```

Which of the following is true?

A) This code will result in a compile-time error.
B) This code will result in a run-time error.
C) The speak method defined in the Animal class will be called.
D) The speak method defined in the Dog class will be called.
E) The speak method will not be called at all.

Answer: D

9) When a subclass is instantiated, the superclass default constructor is executed first (TRUE/FALSE)

Answer: True

10) If the superclass does not contain a default constructor:
A) That class cannot be a superclass

B) A default constructor is created for you
C) A runtime error will occur
D) A parameterized constructor call must be the first statement in the subclass

Answer: D

11) To override a method of a superclass a subclass needs have:
A) An `@override` annotation
B) The same method signature as the superclass method
C) Same method name as the superclass method
D) You cannot override methods in Java

Answer: B

12) In order to prevent a superclass method from being overridden by a subclass method use keyword _____ in the method signature.

Answer: final

Programming Question:

1.

Create an `Animal` class with the following fields and methods:

- Fields: `name, age, weight`.
- Methods:
  - Getters and setters
  - Parameterized constructor which will take `name, age, weight as an argument`
  - `makeSound()` method which will return a string: "Sound"
  - `toString()` method which will return information about the animal

Create the following `Dog` class:

- Dog has to extend the `Animal` class
- Create a parametrized constructor which will take `name, age, weight as an argument`
- Override the `makeSound()` method so that it returns "Bark"

Create the following `Cow class:`

- `Cow has to extend Animal class`
- Create a parametrized constructor which will take `name, age, weight as an argument`
- `Override the makeSound()` method so that it returns "Moo"

Create a driver class and call it Main.
In the driver class create an `ArrayList` which will hold all the animals.
Add a cow to the list with the following parameters:

```
     name="Sally", age = 2, weight = 400
Add a dog to the list with the following parameters:
     name="Bob", age = 1, weight = 50
Add an Animal to the list with the following parameters:
     name="Joe the Cat", age = 2, weight = 10
```

Iterate through the list of animals and print the sound they make as well as the information about the animal.

Answer:

```
public class Animal {
	private String name;
	private int age;
	private double weight;

	public Animal(String name, int age, double weight){
		this.name = name;
		this.age = age;
		this.weight = weight;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public int getAge() {
		return age;
	}

	public void setAge(int age) {
		this.age = age;
	}

	public double getWeight() {
		return weight;
	}

	public void setWeight(double weight) {
		this.weight = weight;
	}

	public String makeSound(){
		return "Sound";
	}

	public String toString(){
		return "The name of the animal is: " + name + "\n"
				+ "The age of the animal: " + age + "\n"
					+ "The weight is: " + weight;
	}

}


public class Dog extends Animal{
```

```
	public Dog(String name, int age, double weight) {
		super(name, age, weight);
	}

	public String makeSound(){
		return "Bark";
	}
}

public class Cow extends Animal{

	public Cow(String name, int age, double weight) {
		super(name, age, weight);
	}

	public String makeSound(){
		return "Mooo";
	}

}

import java.util.ArrayList;

public class Main {

	public static void main(String[] args) {
		ArrayList<Animal> animals = new ArrayList<>();

		animals.add(new Cow("Sally", 2, 400));
		animals.add(new Dog("Bob", 1, 50));
		animals.add(new Animal("Joe the cat", 2, 10));

		for(Animal a : animals){
			System.out.println(a.makeSound());
			System.out.println(a.toString());
		}

	}

}
```