Bachelor of Cyber Security - Pathway Program (Year 1)

CICRA
CAMPUS

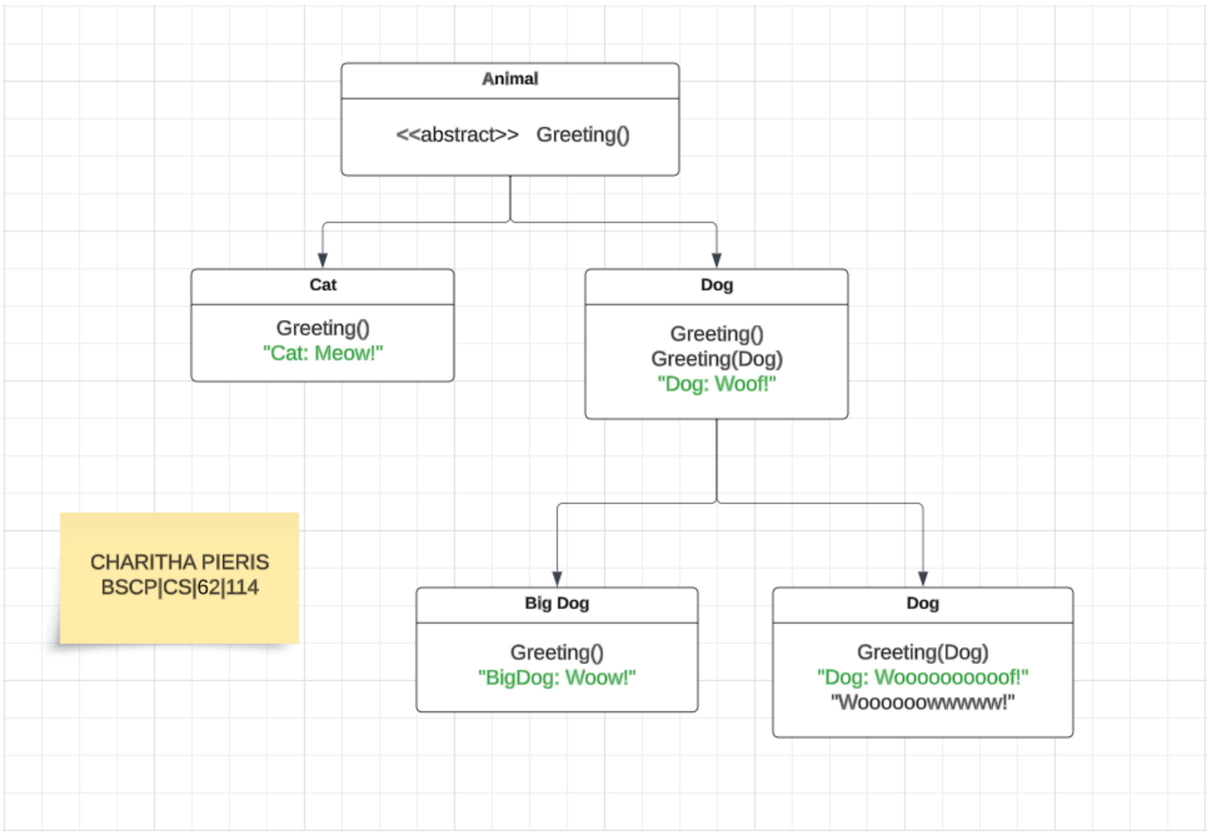
SIT232 Object‑Oriented Development Assignment 6.1P

Prepared by :
Charitha B. Pieris | BSCP|CS|62|114

1) Examine the following code and represent the classes and their relationship as a UML class diagram.

```
abstract public class Animal
{
abstract public void Greeting();
}
public class Cat : Animal
{
override public void Greeting() {
Console.WriteLine("Cat: Meow!");
}
}
public class Dog : Animal
{
override public void Greeting() {
Console.WriteLine("Dog: Woof!");
}
public void Greeting(Dog another) {
Console.WriteLine("Dog: Woooooooooof!");
}
}
public class BigDog : Dog
{
override public void Greeting() {
Console.WriteLine("BigDog: Woow!");
}
new public void Greeting(Dog another) {
Console.WriteLine("Woooooowwwww!");
}
}
```

- Animal is an abstract class with the abstract method Greeting().
- Cat and Dog are concrete classes that inherit from Animal and provide concrete implementations of the Greeting() method.
- Dog also has an overloaded method Greeting(Dog another).
- BigDog is a subclass of Dog and overrides the Greeting() method and provides a new implementation for Greeting(Dog another).



2) Explain the outputs (or existing errors) for the following test program.

```
public class TestAnimal
{
public static void Main(String[] args) {
// Using the subclasses
Cat cat1 = new Cat();
cat1.greeting();
Dog dog1 = new Dog();
dog1.greeting();
BigDog bigDog1 = new BigDog();
bigDog1.greeting();
// Using Polymorphism
Animal animal1 = new Cat();
animal1.greeting();
Animal animal2 = new Dog();
animal2.greeting();
Animal animal3 = new BigDog();
animal3.greeting();
Animal animal4 = new Animal();
// Downcast
Dog dog2 = (Dog)animal2;
BigDog bigDog2 = (BigDog)animal3;
Dog dog3 = (Dog)animal3;
Cat cat2 = (Cat)animal2;
dog2.greeting(dog3);
dog3.greeting(dog2);
dog2.greeting(bigDog2);
bigDog2.greeting(dog2);
bigDog2.greeting(bigDog1);
}
}
```

Creating an instance of an abstract class:

The line Animal animal4 = new Animal(); is an error because you cannot create an instance of an abstract class (Animal is abstract).

Downcasting:

The following downcast lines may result in runtime errors because they involve casting to a type that doesn't match the actual type of the object:

Dog dog3 = (Dog)animal3;: This line will throw a runtime error because animal3 refers to a BigDog object, and you cannot directly cast a BigDog to a Dog.

Cat cat2 = (Cat)animal2;: Similarly, this line will also throw a runtime error because animal2 refers to a Dog object, and you cannot directly cast a Dog to a Cat.