# Lab 6

CST 338

## Task 1

The following is some code designed for a video game. There is an `Alien` class to represent a monster:

```
public class Alien {
    public static final int SNAKE_ALIEN = 0;
    public static final int OGRE_ALIEN = 1;
    public static final int MARSHMALLOW_MAN_ALIEN = 2;

    public int type; // Stores one of the three above types
    public int health; // 0=dead, 100=full strength
    public String name;

    public Alien(int type, int health, String name) {
        this.type = type;
        this.health = health;
        this.name = name;
    }
}
```

There is an `AlienPack` class that represents a band of aliens and how much damage they can inflict:

```
public class AlienPack {
    private Alien[] aliens;

    public AlienPack (int numAliens) {
        aliens = new Alien[numAliens];
    }

    public void addAlien(Alien newAlien, int index) {
        aliens[index] = newAlien;
    }

    public Alien[] getAliens() {
        return aliens;
```

```
    }

    public int calculateDamage() {
        int damage = 0;
        for (int i=0; i < aliens.length; i++) {
            if (aliens[i].type==Alien.SNAKE_ALIEN) {
                damage +=10; // Snake does 10 damage
            } else if (aliens[i].type==Alien.OGRE_ALIEN) {
                damage +=6; // Ogre does 6 damage
            } else if (aliens[i].type==Alien.MARSHMALLOW_MAN_ALIEN) {
                damage +=1; // Marshmallow Man does 1 damage
            }
        }
        return damage;
    }
}
```

The code is not very object-oriented and does not support information hiding in the `Alien` class. Rewrite the code so that inheritance is used to represent the different types of aliens instead of the “type” parameter. This should result in deletion of the “type” parameter. Also rewrite the `Alien` class to hide the instance variables and create a `getDamage()` method for each derived class that returns the amount of damage the alien inflicts. Finally, rewrite the `calculateDamage()` method to use `getDamage()` and write a `main()` method that tests the code.

## Task 2

In Task 1, the `Alien` class was rewritten to use inheritance. The rewritten `Alien` class should be made **abstract** because there will never be a need to create an instance of it, only its derived classes. Change this to an abstract class and also make the `getDamage()` method an abstract method. Test the class from your `main()` method to ensure that it still operates as expected.

Please submit your code for Tasks 1 and 2 on iLearn.