

# Abstract Classes and Interfaces



UNIVERSITAS  
INDONESIA  
*Veritas, Probitas, Iustitia*

FAKULTAS  
ILMU  
KOMPUTER

## making the abstract class

```
public abstract class SimpleAbstract {
    int x;

    SimpleAbstract() {
        System.out.println("Abstract");
    }
    //abstract method
    abstract void hello(String name);

    void concrete() {
        System.out.println("Concrete in abstract");
    }
}
```

## ABSTRACT CLASS

- Abstract Class adalah sebuah class yang tidak bisa di-instansiasi (tidak bisa dibuat menjadi objek)
- berperan sebagai 'kerangka dasar' bagi class turunannya.
- abstract class umumnya akan memiliki abstract method.

## inheriting abstract class

```
public class Simple extends SimpleAbstract {

    void hello(String name) {
        System.out.println("Hello, " + name);
    }

    public static void main(String[] args) {
        Simple s = new Simple();
        s.hello("Sis");
        System.out.println(s.x);
        s.concrete();
    }
}
```

## INTERFACES

- Interface membuat sebuah kontrak atas apa yang dapat dilakukan oleh Class
- Sebuah class dapat "extend" banyak interface. dengan menggunakan "implements"
- Method interface adalah abstract dan public
- Tidak dapat memiliki Constructor

## Interface

```
public interface GPSEnabled {
    public void printLocation();
}
```

## Implementing Interface to class

```
public class Cellphone implements GPSEnabled {
    public void printLocation() {
        System.out.println("Location");
    }
}
```

## Using the interface

```
public static void main(String[] args) {

    Cellphone ciaoMi = new Cellphone();
    ciaoMi.printLocation();
    GPSEnabled samsu = new Cellphone();
    samsu.printLocation();
}
```



BY