

```
public class lol {  
    /* Instance Variabel */  
    String name;  
    String breed;  
    int age;  
    String color;  
  
    /* Deklarasi konstruktor dari class */  
    public lol(String name, String breed, int age, String color)  
    {  
        this.name = name;  
        this.breed = breed;  
        this.age = age;  
        this.color = color;  
    }  
  
    /* method 1 */  
    public String getName()  
    {  
        return name;  
    }  
  
    /* method 2 */  
    public String getBreed()  
    {  
        return breed;  
    }  
  
    /* method 3 */
```

```

public int getAge()
{
    return age;
}

/* method 4 */
public String getColor()
{
    return color;
}

public String toString()
{
    return ("Hi my name is " + this.getName() +
        "\n My breed, age and color are " + this.getBreed()
        + ", " + this.getAge() + ", " + this.getColor());
}

public static void main(String[] args)
{
    lol sul = new lol("sulton", "papillon", 5, "white");
    System.out.println(sul.toString());
}
}

```

### Catatan

Class pada kode program sebelumnya mengandung sebuah constructor tunggal. Sebuah constructor dapat dikenali dari pendeklarasiannya yang menggunakan nama yang sama seperti class dan tidak memiliki tipe return.

Java compiler membedakan constructor berdasarkan angka dan tipe dari argumennya. Constructor pada class lol mengambil empat argument. Statement berikut menyediakan "sulton", "papillon", 5, "white" sebagai nilai untuk argument:

```
lol sul = new lol("sulton", "papillon", 5, "white");
```

Semua class setidaknya memiliki satu constructor. Jika sebuah class dideklarasikan secara eksplisit, maka compiler Java secara otomatis menyediakan sebuah no-argument constructor yang juga disebut dengan default constructor. Default constructor memanggil class parent no-argument constructor (seperti hanya memiliki satu statement yaitu `super();`), atau class object constructor jika class tidak memiliki parent yang lain (seperti class object adalah parent dari semua class baik secara langsung maupun tidak langsung).

**Variabel lokal :** Variabel didefinisikan didalam method konstruktor atau blok disebut variabel lokal, variabel akan dideklarasikan dalam method dan variabel akan dihancurkan ketika method telah selesai.

**Variabel Instance :** Variabel dalam class tetapi diluar method apapun, variabel ini dipakai ketika class dimuat. Variabel instance dapat diakses dari dalam method, konstruktor atau class tertentu.

**variabel Class :** Variabel dideklarasikan dalam class, diluar method apapun, dengan kata kunci `static`.

**Deklarasi:** Deklarasi variabel dengan nama variabel dan jenis objek.

**Instansiasi:** kata kunci `"new"` digunakan untuk membuat objek.

**Inisialisasi:** kata kunci `"new"` diikuti dengangan panggilan untuk konstuktor, panggilan ini menginisialisasi objek baru.

Fungsi program menampilkan nama breed age dan color.