BİL106 Nesne Yönelimli Programlama

Dr. Öğr. Üyesi Yavuz CANBAY

Kahramanmaraş Sütçü İmam Üniversitesi

Bilgisayar Mühendisliği Bölümü

Bölüm 6: Nesne ve Sınıflar

Nesne Yönelimli Programlama (OOP) Nedir?

- Fonksiyonel/Prosedürel programlama; veri üzerinde işlem yapan fonksiyon veya prosedürlerin yazılması
- Nesne yönelimli programlama; hem veri hem de fonksiyon içeren nesnelerin oluşturulması

OOP vs Prosedürel Programlama

- OOP hızlıdır
- OOP programlar için düzgün bir yapı sunar
- OOP, yazılan kodun tekrarlanmasını önler (helps to keep the C++ code DRY "Don't Repeat Yourself’) ve kodun yönetimini, değiştirilmesini ve hata ayıklamasını kolaylaştırır.
- Daha az kod ile tekrar kullanılabilir uygulamalar yapılmasını sağlar
- Geliştirme süresi daha kısadır

Nesnenin Temel Özellikleri

- Özellikleri (Attributes) vardır: renk, şekil, boyut, genişlik, yükseklik vb.
- Davranışları (Functions, methods) vardır: her nesnenin kendine özgü davranışları vardır.
- Üretilen yeni nesneler Mirasla (Inheritance) sınıfların bazı özellik ve yöntemlerine sahip olurlar.
- Bilgiler Gizlenebilir; nesnelerin iş görmesi için diğer nesneler hakkında bilgi sahibi olması şart değildir.

Sınıf vs Nesne

- Sınıf;
  - Konsept bir kavramdır, mantıksal bir varlıktır
  - Kendisine bir hafıza ayrılmaz
  - Herhangi bir nesne olmadan mevcut olabilirler
- Nesne;
  - Bir sınıfın örneğidir (instance)
  - Gerçek yapılardır, gerçek bir varlıktır
  - Her bir nesne kendi hafızasına sahiptir
  - Nesneler sınıf olmadan mevcut olamazlar

Sınıflar ve Nesneler

› Sınıf ve nesne nesne yönelimli programlamanın iki temel yapısıdır

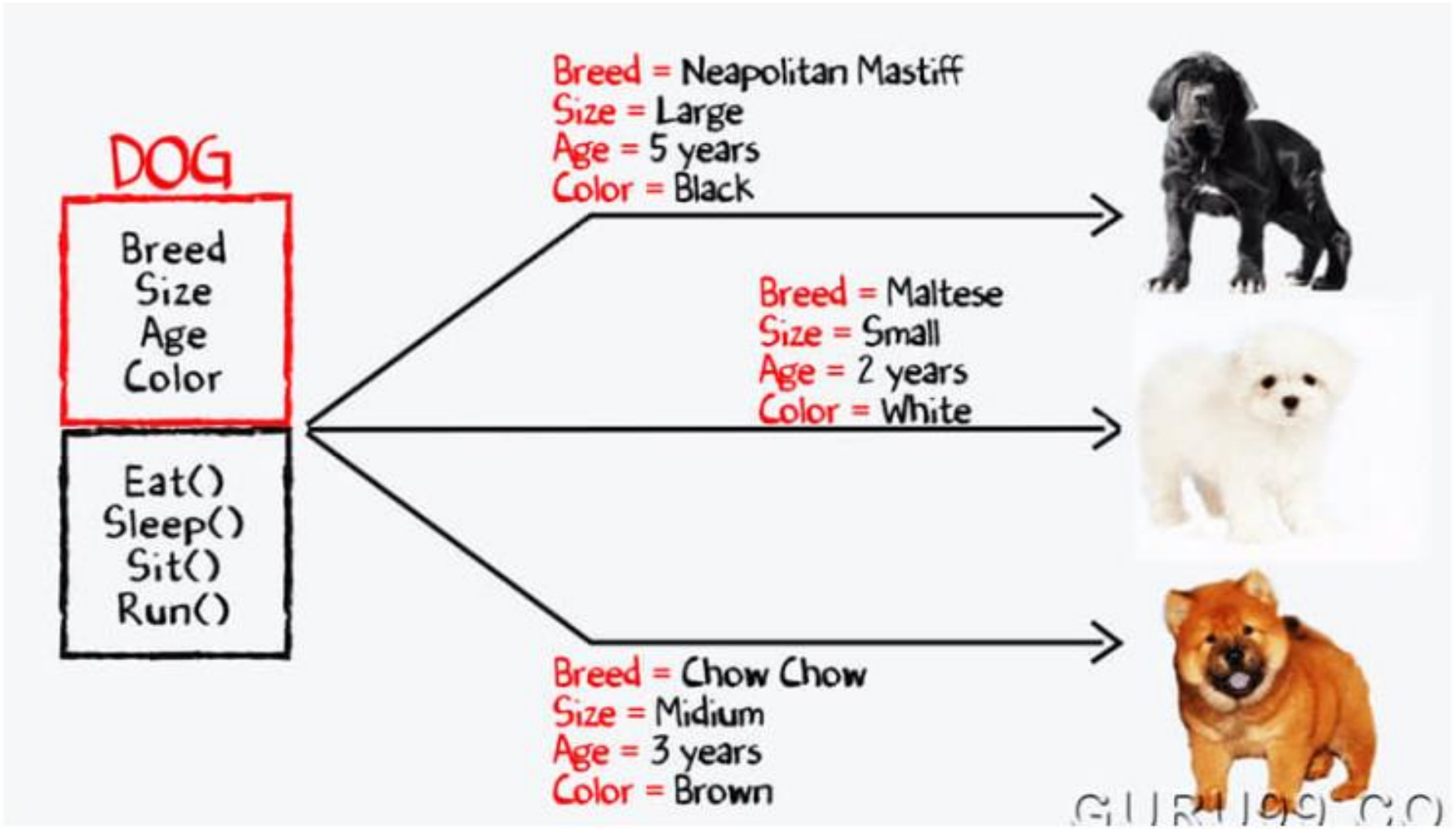
› Sınıf; nesneler için bir taslaktır

› Nesne; sınıftan türeyen bir örnektir

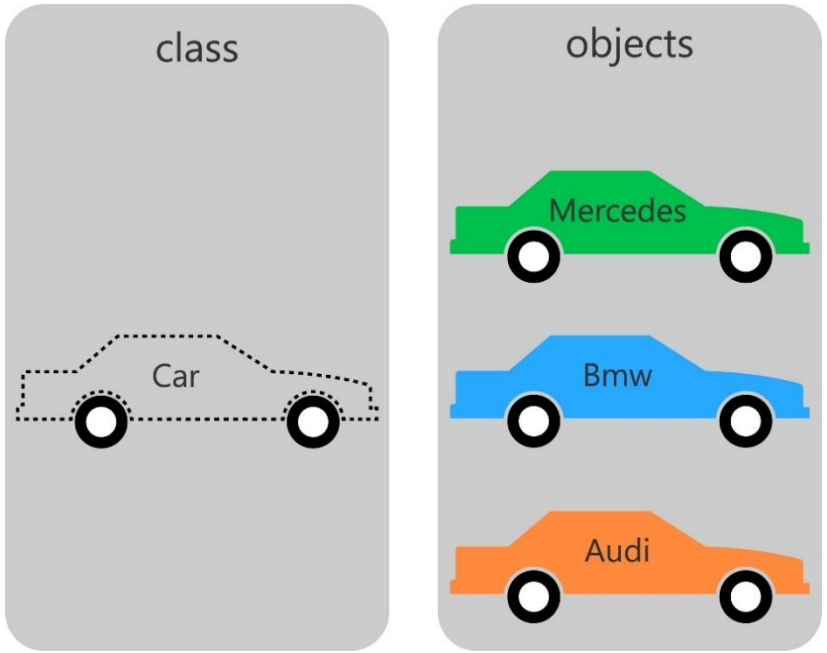
Sınıflar ve Nesneler

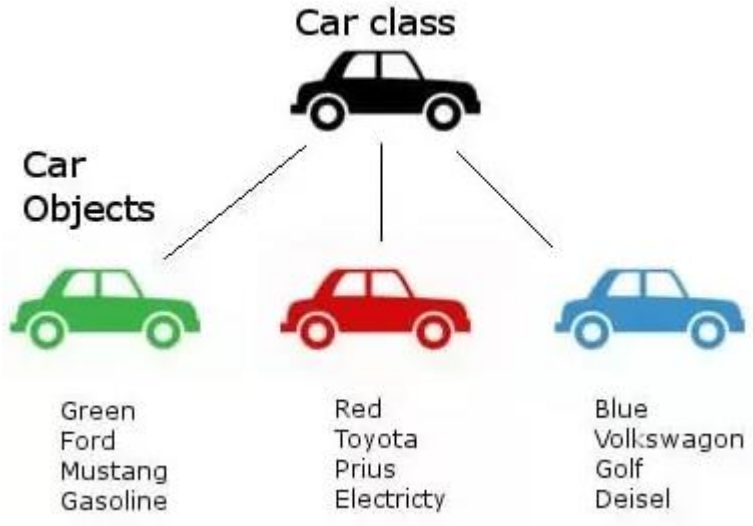
- C++'da her şey sınıf ve nesneleri kapsar,
- Gerçek hayatta Araba bir sınıftır,
  - Araba şu özelliklere sahiptir; ağırlık, renk, marka, model…
  - Araba şu metot sahiptir; sürme, frene basma, korna çalma
- Özellik ve metotlar temel olarak bir sınıfa ait değişkenler ve fonksiyonlardır.
- Değişken ve fonksiyonlar genel olarak "sınıf üyeleri" olarak isimlendirilirler.
- Bir sınıf kullanıcı tanımlı bir veri tipi olup programlarda kullandığımız yapıdır.

Sınıf ve Nesneler için Örnekler-1

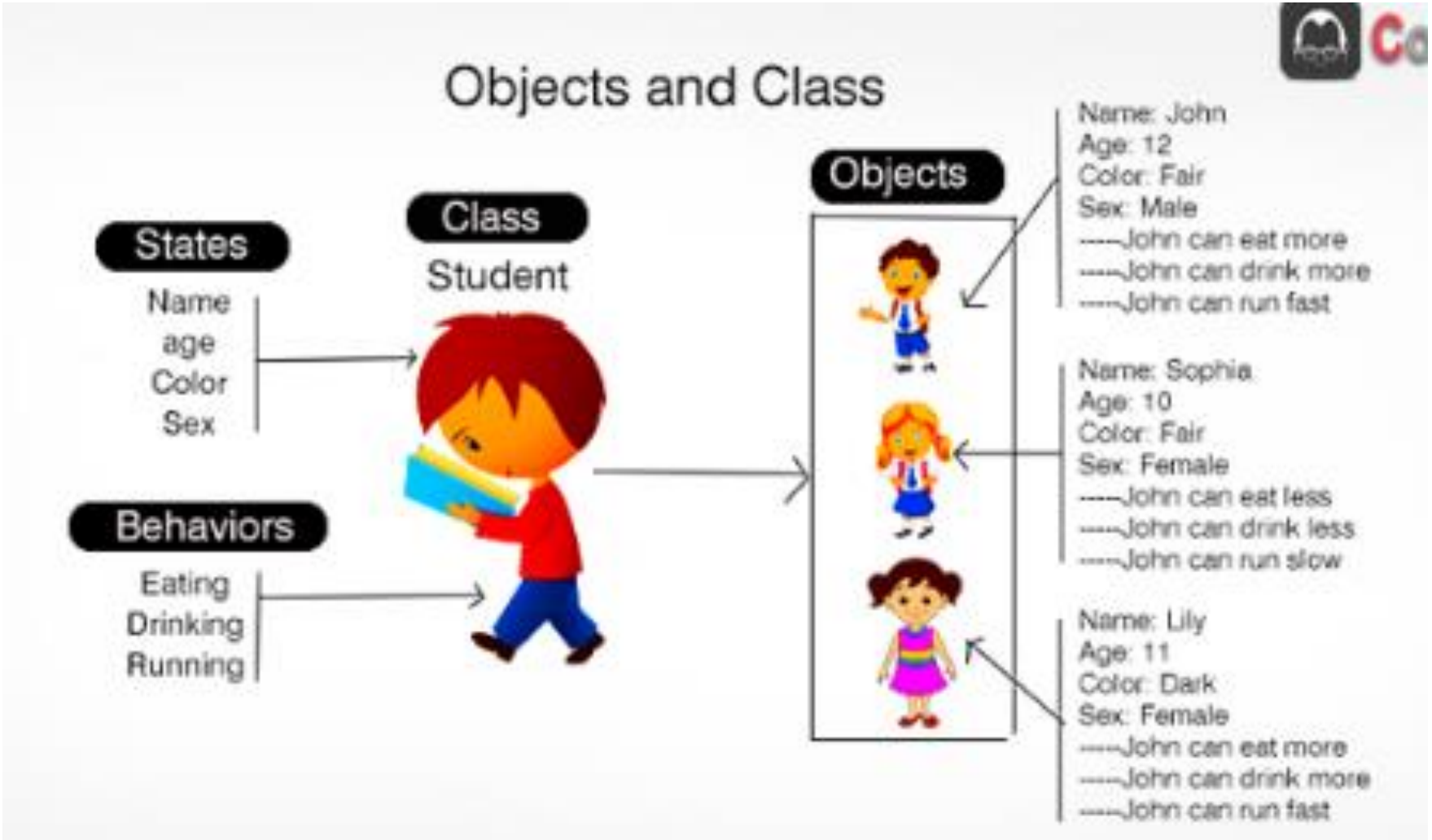


Sınıf ve Nesneler için Örnekler-2

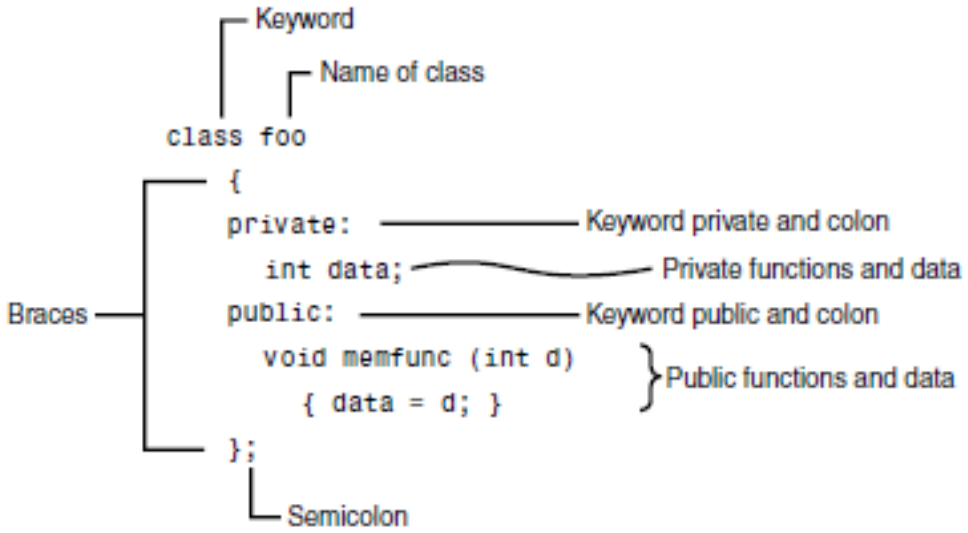




Sınıf ve Nesneler için Örnekler-3



Sınıf tanımlamanın sentaksı



Sınıf ve Nesneler için Örnekler-4

```
#include <iostream>
#include <string>
using namespace std;

class MyClass {        // The class
  public:              // Access specifier
    int myNum;         // Attribute (int variable)
    string myString;   // Attribute (string variable)
};

int main() {
  MyClass myObj;  // Create an object of MyClass

  // Access attributes and set values
  myObj.myNum = 15;
  myObj.myString = "Some text";

  // Print values
  cout << myObj.myNum << "\n";
  cout << myObj.myString;
  return 0;
}
```

Sınıf ve Nesneler için Örnekler-5

```
#include <iostream>
#include <string>
using namespace std;

class Car {
  public:
    string brand;
    string model;
    int year;
};

int main() {
  Car carObj1;
  carObj1.brand = "BMW";
  carObj1.model = "X5";
  carObj1.year = 1999;

  Car carObj2;
  carObj2.brand = "Ford";
  carObj2.model = "Mustang";
  carObj2.year = 1969;

  cout << carObj1.brand << " " << carObj1.model << " " << carObj1.year << "\n";
  cout << carObj2.brand << " " << carObj2.model << " " << carObj2.year << "\n";
  return 0;
}
```

Basit bir sınıf

```
// smallobj.cpp
// demonstrates a small, simple object
#include <iostream>
using namespace std;
////////////////////////////////////////////////////////////////
class smallobj               //define a class
   {
   private:
      int somedata;          //class data
   public:
      void setdata(int d)    //member function to set data
         { somedata = d; }
      void showdata()        //member function to display data
         { cout << "Data is " << somedata << endl; }
   };
////////////////////////////////////////////////////////////////
int main()
   {
   smallobj s1, s2;   //define two objects of class smallobj

   s1.setdata(1066);  //call member function to set data
   s2.setdata(1776);

   s1.showdata();     //call member function to display data
   s2.showdata();
   return 0;
   }
```

Nesne olarak araba parçaları

```
// objpart.cpp
// widget part as an object
#include <iostream>
using namespace std;
////////////////////////////////////////////////////////////////
class part               //define class
   {
   private:
      int modelnumber;   //ID number of widget
      int partnumber;    //ID number of widget part
      float cost;        //cost of part
   public:
      void setpart(int mn, int pn, float c)  //set data
         {

          modelnumber = mn;
          partnumber = pn;
          cost = c;
          }
       void showpart()                       //display data
          {
          cout << "Model "    << modelnumber;
          cout << ", part "   << partnumber;
          cout << ", costs $" << cost << endl;
          }
    };
 ////////////////////////////////////////////////////////////////
 int main()
    {
    part part1;                         //define object
                                        //   of class part
    part1.setpart(6244, 373, 217.55F);  //call member function
    part1.showpart();                   //call member function
    return 0;
    }
```

Nesne olarak daire

```
// circles.cpp
// circles as graphics objects
#include "msoftcon.h"          // for graphics functions
////////////////////////////////////////////////////////////////
class circle                   //graphics circle
   {
   protected:
      int xCo, yCo;            //coordinates of center
      int radius;
      color fillcolor;         //color
      fstyle fillstyle;        //fill pattern
   public:                     //sets circle attributes
      void set(int x, int y, int r, color fc, fstyle fs)
         {
         xCo = x;
         yCo = y;
         radius = r;
         fillcolor = fc;
         fillstyle = fs;
         }
      void draw()              //draws the circle
         {
         set_color(fillcolor);              //set color
         set_fill_style(fillstyle);         //set fill
         draw_circle(xCo, yCo, radius);     //draw solid circle
         }
   };
////////////////////////////////////////////////////////////////
int main()
   {
   init_graphics();           //initialize graphics system

   circle c1;                 //create circles
   circle c2;
   circle c3;
                              //set circle attributes
   c1.set(15, 7, 5, cBLUE, X_FILL);
   c2.set(41, 12, 7, cRED, O_FILL);
   c3.set(65, 18, 4, cGREEN, MEDIUM_FILL);
   c1.draw();                 //draw circles
   c2.draw();
   c3.draw();
   set_cursor_pos(1, 25);     //lower left corner
   return 0;
   }
```

Veri tipi olarak C++ nesneleri

```
// englobj.cpp
// objects using English measurements
#include <iostream>
using namespace std;
////////////////////////////////////////////////////////////////
class Distance                    //English Distance class
   {
   private:
      int feet;
      float inches;
   public:
      void setdist(int ft, float in)  //set Distance to args
         { feet = ft; inches = in; }

      void getdist()              //get length from user
         {
         cout << "\nEnter feet: ";  cin >> feet;
         cout << "Enter inches: ";  cin >> inches;
         }

      void showdist()             //display distance
         { cout << feet << "\'-" << inches << '\"'; }

          };
       ////////////////////////////////////////////////////////////////
       int main()
          {
          Distance dist1, dist2;         //define two lengths

          dist1.setdist(11, 6.25);       //set dist1
          dist2.getdist();               //get dist2 from user

                                         //display lengths
          cout << "\ndist1 = ";  dist1.showdist();
          cout << "\ndist2 = ";  dist2.showdist();
          cout << endl;
          return 0;
          }
```