# Object-Oriented Programming Concepts

Session: 3

## Classes and Methods

# Objectives and Introduction

- Explain Class
- Describe Visibility Modifiers
- Explain Methods
- Explain Static data fields and Constant data fields
- Describe Accessor, Mutator, and Forward declaration

Every object can be categorized into some class

Characteristics of object are enclosed within the class as data members

To access and manipulate data members, the object has to make use of methods

# Class 1

- It is a type or a software construct that encloses the data members and functions or behavior of an entity into an enclosed structure.

The syntax of a class is shown here :

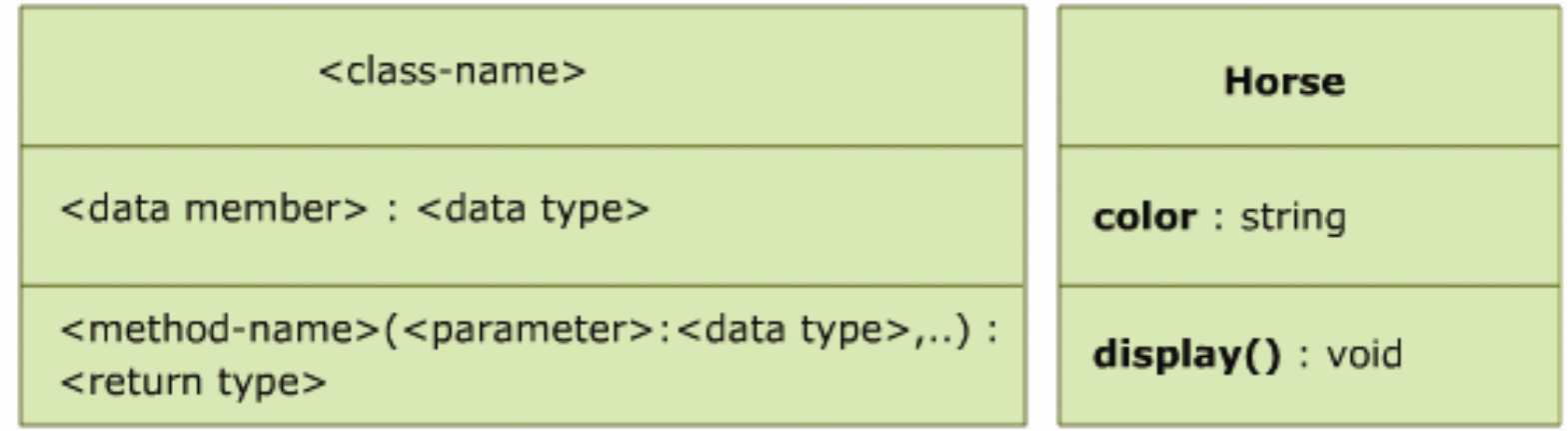
```
<access--modifier>class<class-name>
{
    <data-members>
    <methods>
}
```

Code Sinppet

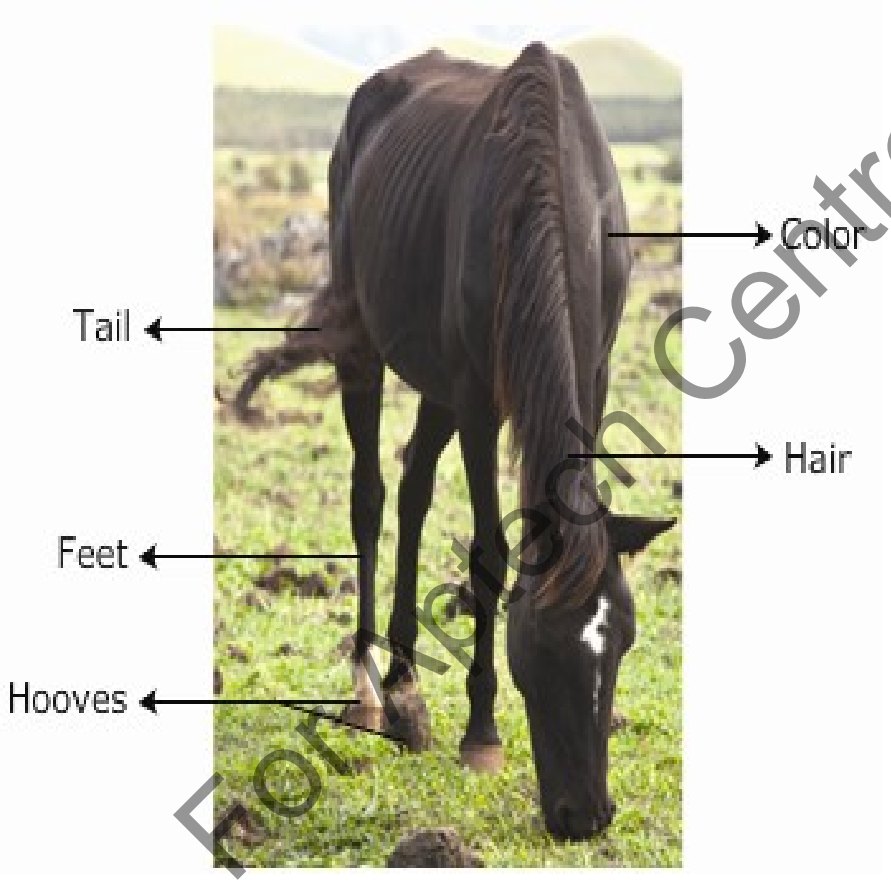
```
public class Horse
{
    string color;
    public void diplay()
    {
        console.writeLine(color);
    }
}
```

# Class 1-2

The figure shows the structure of a class diagram.



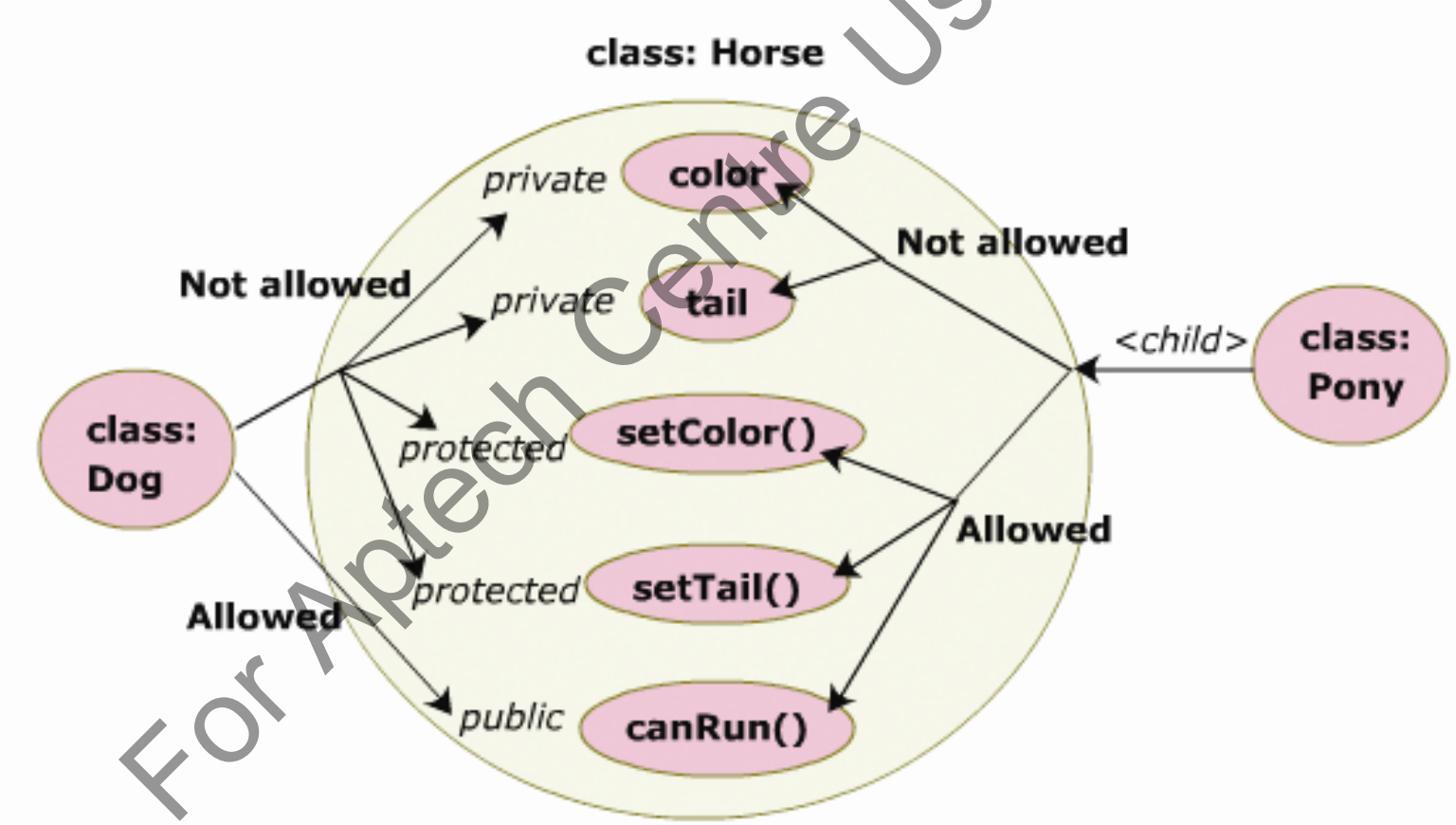
- The figure shows an example of entity with its characteristics.




| Horse |
| --- |
| **Characteristics:** |
| Color |
| Hair |
| Feet |
| Hooves |
| Tail |
| **Behavior:** |
| Run |
| Eat |
| Whine |

# Visibility Modifiers

- The figure shows how class members can be secured using visibility modifiers.



# Methods

- Defines the behavior of a class
- Consists of the actions that an object or instance of a class can perform on the data members

Syntax

```
<access-modifier> <return-type> <method-
name> (<data-type> <parameter-name>,…)
{
// statement1
// statement2
<return-value (or expression)>
}
```

# Object

- To access any member of a class, a copy of that class called the object of the class, has to be created

Code Snippet :

```
public class Horse
{
    private string color; // attribute
    public void setColor(string col) // method
    {
     color = col;
    }
    static void Main()
    {
     Horse h1 = new Horse(); // creating
                        object of class Horse
     h1.setColor("Black");
    }
 }
```

# Constructor

- constructor is a specialized method that has the same name as the class name
- It is used to construct an object of a class and initialize its data members.
- Constructor arguments are used to initialize the fields of an object at runtime

**Syntax:**
**Default or no-argument constructor**

```
<constructor-name> ()
{
   // initialization statements
}
```

# Static Data Fields

For a static or class variable, only a single copy is shared by all objects of the class

Class that is declared as static, cannot be instantiated and its members can be accessed directly using the class name

Method that is declared static, can be accessed directly using the class name without creating object of the class

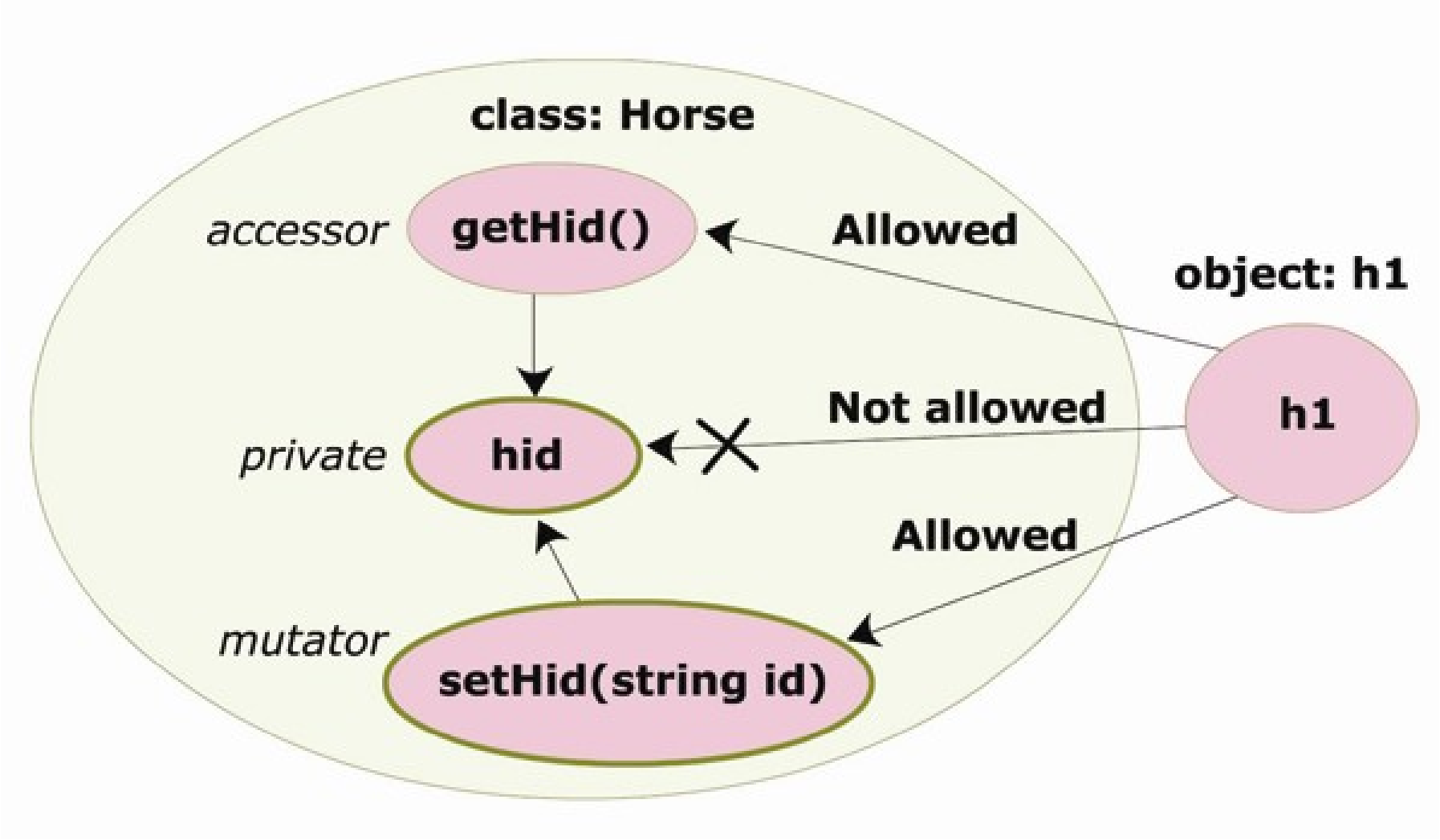
# Constant Data Fields

- It is a variable whose value is fixed during compilation and does not change later
- Values of such fields can be fixed before the object of the class is created

The following C# code demonstrates the creation of a constant.

```
public class Horse
{
public const int feet = 4;
…..
}
```

# Accessor and Mutator



# Forward Declaration

This process helps classes that are referenced later in a file to be used at an earlier stage in .the code without any conflict

It is a prior hint to the compiler that the function definition or implementation has been done later in the code

is also called a function prototype

# Forward Declaration

The C++ code given in Code Snippet \\ shows an example of function prototype.

```
#include <iostream.h> // function prototype
void display(string col); // line1
  static void Main ()
 {
  string color;
  cout << "Enter color ";
  cin >>color ;
  display(color); // line 2
 } // end of main()
void display (string col) // line 3
 {
  cout<<"Color is "+ col;
 }
```

# Summary 1

- A class is a type or a software construct that encloses the data members and of an entity into an enclosed structure.
- A visibility modifier is used with a class, data field as well as a method to restrict access to them from outside the class.
- A method defines the behavior of a class and consists of the actions that an object of a class can perform on the data members.
- A constructor is a specialized method that has the same name as the class name and is used to initialize its data members.
- A static or class variable is one for which there is only a single copy and it is shared by all objects of a class.
- A constant is a variable whose value is fixed during compilation and does not change later.
- Forward declaration of a function is a statement that informs the compiler about the signature of the function without its implementation.