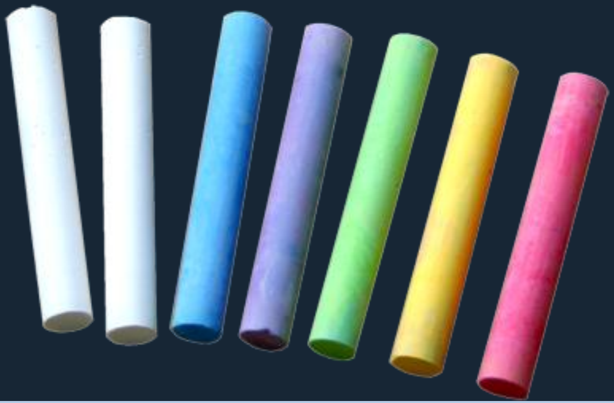
Unit-2
Classes and Objects

Class

- A class is a user defined data type with a template that serves to define its properties.

- java is a true object-oriented language and therefore anything we wish to represent in a java program must be encapsulated in a class that defines the state and behavior of the basic program components known as object.

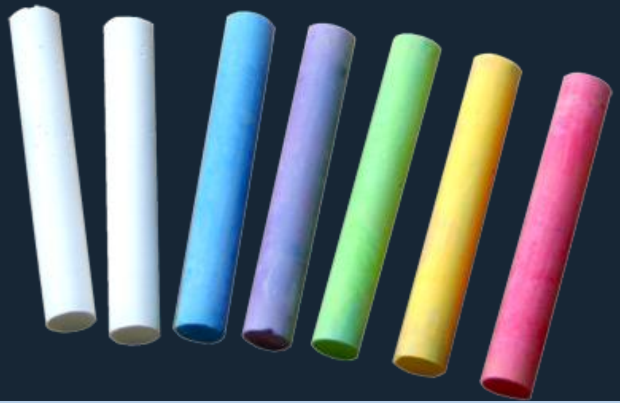


Class

- Once the class type has been defined, we can create variables of that type, called as instance of classes, which are the actual object.

The basic form of a class definition is

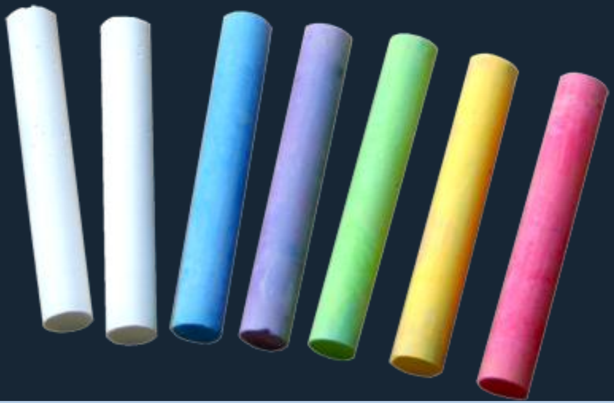
```
class classname [extend superclassname]
{
        [filed declaration;]
        [Methods declaration];
}
```



Field and Method Declaration

Syntax:

```
class classname
    {
          type instancevariable;
    [also known as member variable
    it is called instance variable because
    they are created whenever an object of
    the class is instantiated.]
          type methodname (parameter list)
          {
                method-body;
          }
    }
```



Example

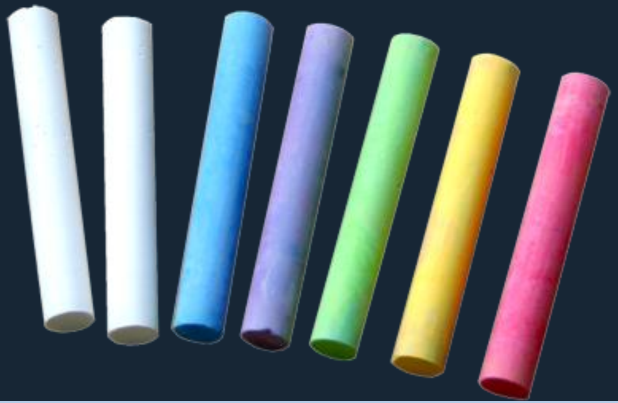
```
class rectangle
        {
                int length, width;
                void getdata (int x, int y)
                {
                }
        }
```

Creating Object

- Obtaining objects of a class is a two-step process.

First, you must declare a variable of the class type. This variable does not define object. Instead, it is simply a variable that can refer to an object.

Second, you must acquire an actual physical copy of the object and assign it to that variable.



Creating Object

We can do this using the new operator, it’ll dynamically allocates memory for an object and returns a reference to that object.

Creating Object

Syntax:

```
            classname objectname;
 // declare the object
            objectname = new classname();
//instantiate the object
```

OR

```
      classname objectname = new
     classname();
```

Creating Object

Rectangle rect1;

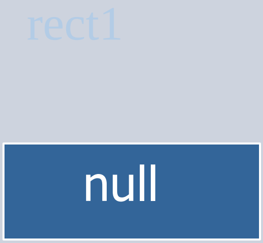
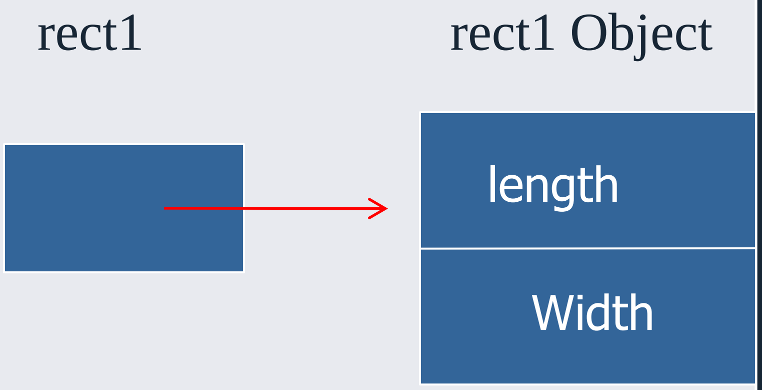
//declare a variable to hold the object reference

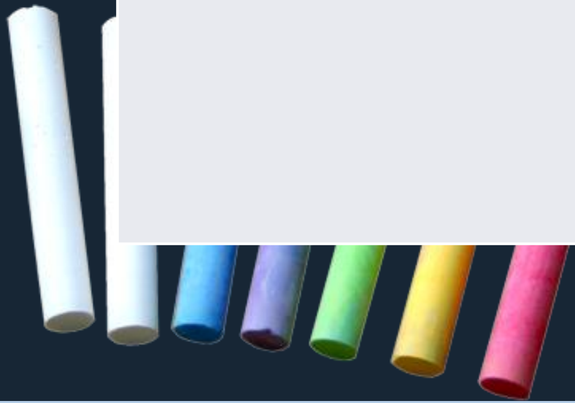
rectl1 = new Rectangle();

// actually assign the object reference to the variable. now it is an object of rectangle class.

OR

Rectangle rect1 = new Rectangle();

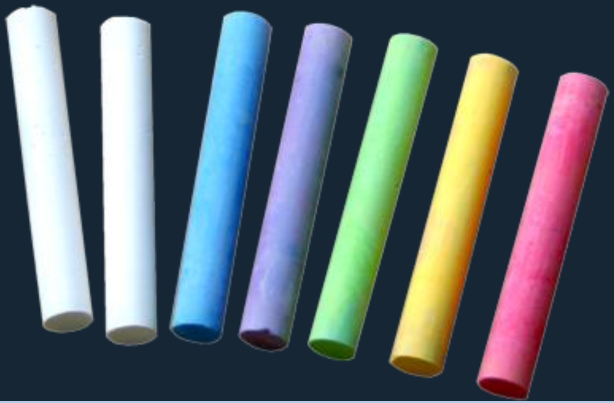
Statement	Effect
Rectangle rect1;	
rect1=new Rectangle();	



Accessing Class Members:

If we are outside the class then to access the instance variable and the methods, we must use the concerned object and the dot operator as shown below.

```
objectname.variablename = value;
objectname.methodname(paramater.list);
```



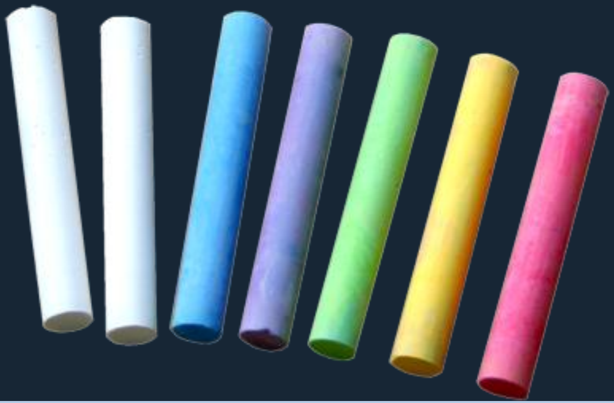
Introducing Methods:

A Java method is a set of Java statements which can be included inside a Java class.

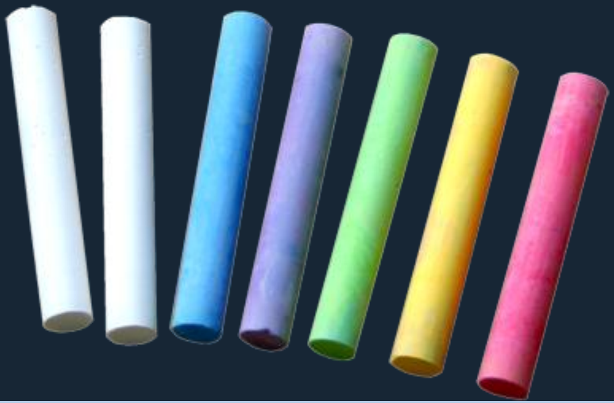
Java methods are similar to functions or procedures in other programming languages.

General Form:

```
type name(parameter-list){
        // body of method
}
```



Classification of Mehod

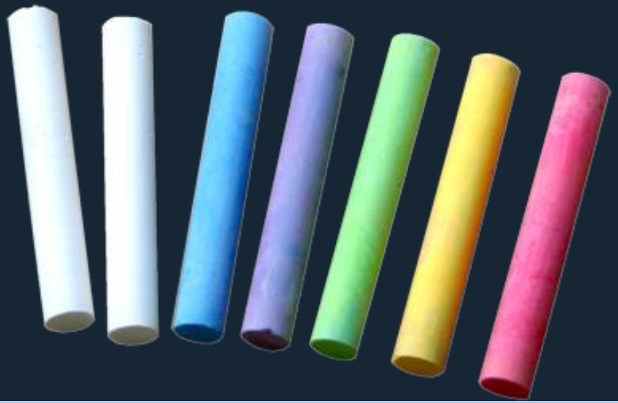


1. Based on nature of creation
   1. User Defined
   2. Predefined
2. Based on return type
   1. Return a value
   2. Return void
3. Based on method call
   1. Nonrecursive
   2. recursive

Constructors:

Constructor are special type of method that enables an object to initialize itself when it is created.

It have the same name as the class itself, and do’t specify the return type not even void because it return the instance of the class itself.



Getter and Setter Method

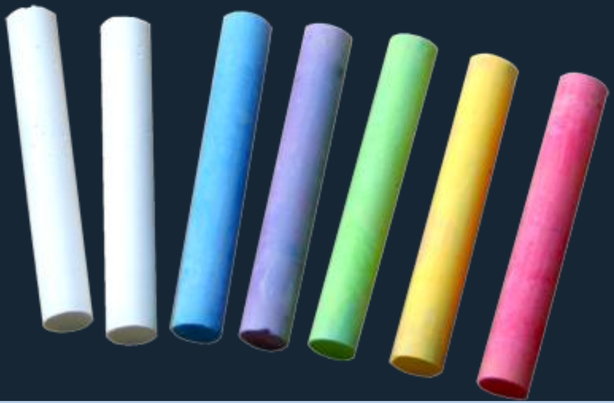
- Define a specific methods for updating or accessing too many items.

- A getter method “gets” the value of a variable.

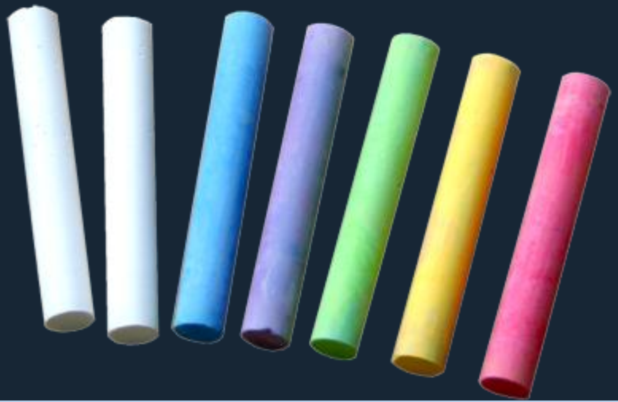
- A setter method that “sets” the value of a variable.

Garbage Collection

- Technique which destroy object and release it’s memory is called garbage collection.
- How it work?
  - When no reference to an object exist, that object is assumed to be no longer needed and the memory occupied by the object can be reclaimed(regain).
- Garbage collection only occurs periodically during the execution of your program.



The Finalize Method



- Sometimes an object will need to perform some action when it is destroy, to handle such situations, java provide a mechanism called finalization.
- For Example, if an object is holding some non-java resource such as file handle or character font, then you might want to make sure these resources are freed before an object is destroyed.

The Finalize Method

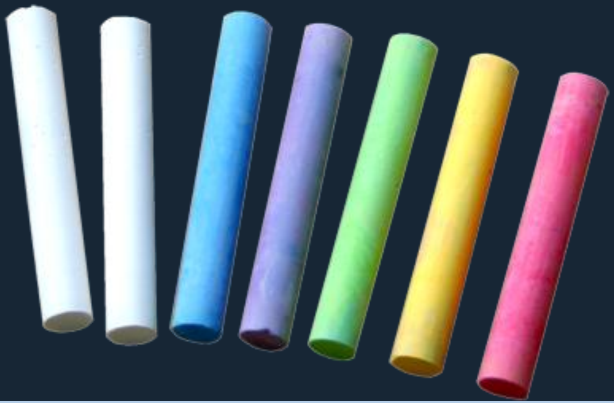
- By using finalization, you can define specific actions that will occur when an object is just about to be reclaimed by the garbage collector.

- Syntax:
  - Protected void finalize()

```
 {
         // finalization code.
 }
```

The Finalize Method

- finalize() is only called just prior to garbage collection. It is not called when an object goes out-of scope.



Final

There are three use of final.

1. A variable can be declared as final. It will prevent its content from being modified.

This mean that you must initialize a final variable when it is declared.

For Example:

```
final int x=1;
```

Final

2. Using final to prevent overriding

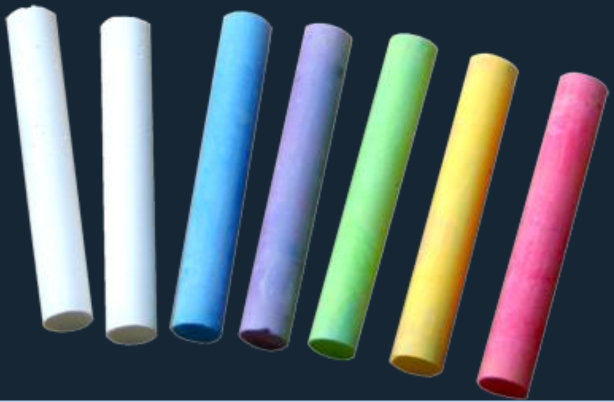
To disallow a method from beign overridden, specify final as a modifier at the start of its declaration.

Method declared as a final can not be overridden.

Syntax:

```
final return-type method-name()
{

}
```



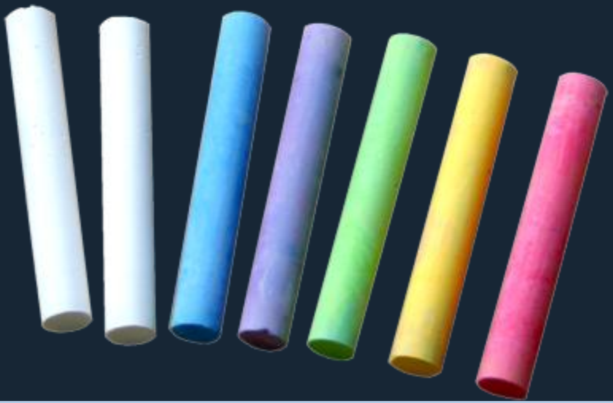
Final

3. Using final to prevent Inheritance

To disallow a class from being inherited, declare a class as a final implicitly declared all of its methods as a final, too.

Syntax:

```
final class class-name
{
}
```



WHAT IS STATIC FIELDS??..

- Static fields are also know as class fields.
- They are simply fields that have the static modifier in their declarations.
- Static field is associated with the class rather than an object.

FEATURES OF STATIC FIELD

- A Static field is shared by all objects of its class and thus relates to the class itself.

- A Static field declared by using the Static keyword as a modifier.

For Example:

```
static  int  x;
```

FEATURES OF STATIC FIELD

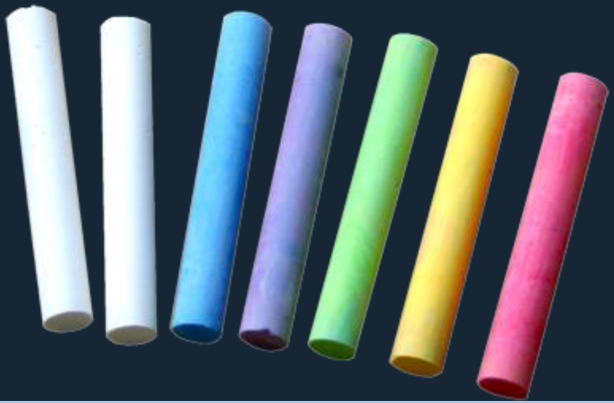
- Static variables are initialized to default values when the class is loaded into memory.

WHAT IS STATIC METHOD??..

- A static method is a simple method having static keyword as a modifier.

- Syntax:

  static ReturnType MethodName(parameters)



FEATURES OF STATIC METHOD

- static methods are created to access it without any object instance.
- A static method can be accessed without creating an instance of the class.
- Static method can call only other static methods and static variables defined in the class.

FEATURES OF STATIC METHOD

- A static method cannot access non-static/instance variables, because a static method is never associated with any instance.

- To call non static method from static method user have to use class object associated with that method.