CORE JAVA

UNIT II

LEARNING OUTCOMES

- TO UNDERSTAND THE FUNDAMENTALS OF CLASSES IN JAVA
- TO LEARN HOW TO WORK WITH METHODS

WHAT IS A CLASS?

ANY CLASS DEFINES A NEW DATA TYPE.

THIS TYPE CAN THEN BE USED TO CREATE OBJECTS OF THAT TYPE.

THEREFORE, A CLASS IS A TEMPLATE FOR AN OBJECT AND AN OBJECT IS AN INSTANCE OF A CLASS.

GENERAL STRUCTURE

```
class classname [extends superclassname]
 {
          [field declaration];
          [methods declaration];
  }
```

FIELD DECLARATION

- The data, or variables, defined within a class are called instance variables.
- The code is contained within methods.
- Collectively, the methods and variables defined within a class are called members of the class.
- As a general rule, it is the methods that determine how a class’ data can be used.

Example:

```
class Rectangle
{
        int length;
        int width;
}
```

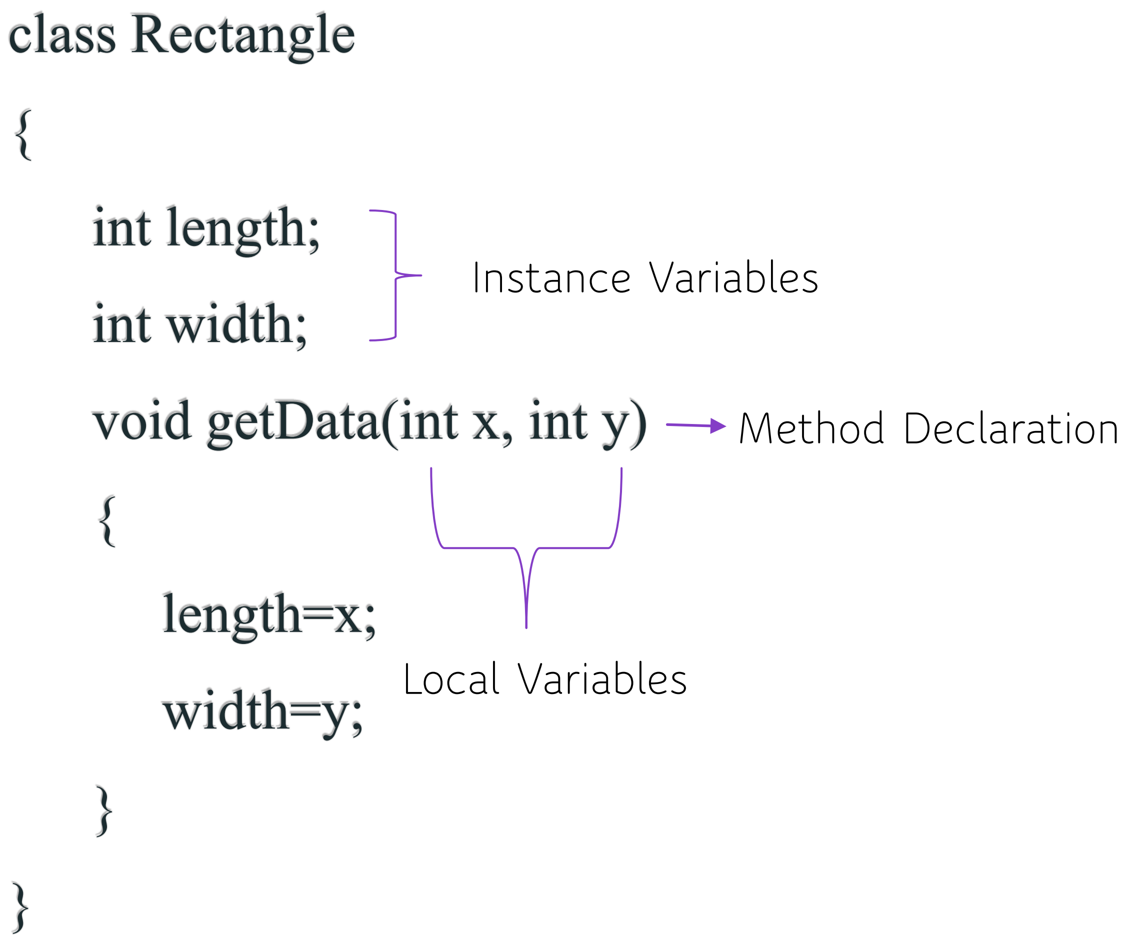
Instance Variables

WHAT IS A METHOD?

- Methods are declared inside the body of the class but immediately after the declaration of instance variables.
- General Form:

```
type method_name(parameter-list)
{
    Method – body;
}
```

EXAMPLE



```
class Rectangle
{
    int length;
    int width;
    void getData(int x, int y)
    {
        length=x;
        width=y;
    }
}
```

Instance Variables

Method Declaration

Local Variables

EXAMPLE 2:

```
class Rectangle
{
 int length,width;  //combined declaration
  void getdata(int x, int y)  //method declaration
  {
    length=x;
    width=y;
   }
        int rectarea()                // method declaration
  {
    int area=length*width;
    return (area);
  }
}
```

CREATING OBJECTS

Objects in java are created using the new operator.

The new operator creates an object of the specified class and returns a reference to that object.

Example:

- Rectangle rect1; //declare the object
- rect1=new Rectangle(); //instantiate the object

Both statements can be combined

Rectangle rect1=new Rectangle();

ACCESSING CLASS MEMBERS

- To access instance variables and methods outside the class we must use concerned object and the dot operator.

```
Objectname.variablename=value;
Objectname.methodname(parameter-list);
```

Example :-

```
rect1.length=15;        //accessing instance variables
rect1.width=20;         //accessing instance variables
rect1.getdata(20,30);   //calling the method
```

ACCESS SPECIFIERS

PUBLIC

PRIVATE

PROTECTED

TYPES OF CLASSES

PUBLIC

- A class defined public provides access to its variables and methods outside its definition.

PRIVATE

- A private class specifies that its members can only be accessed in its own definition.

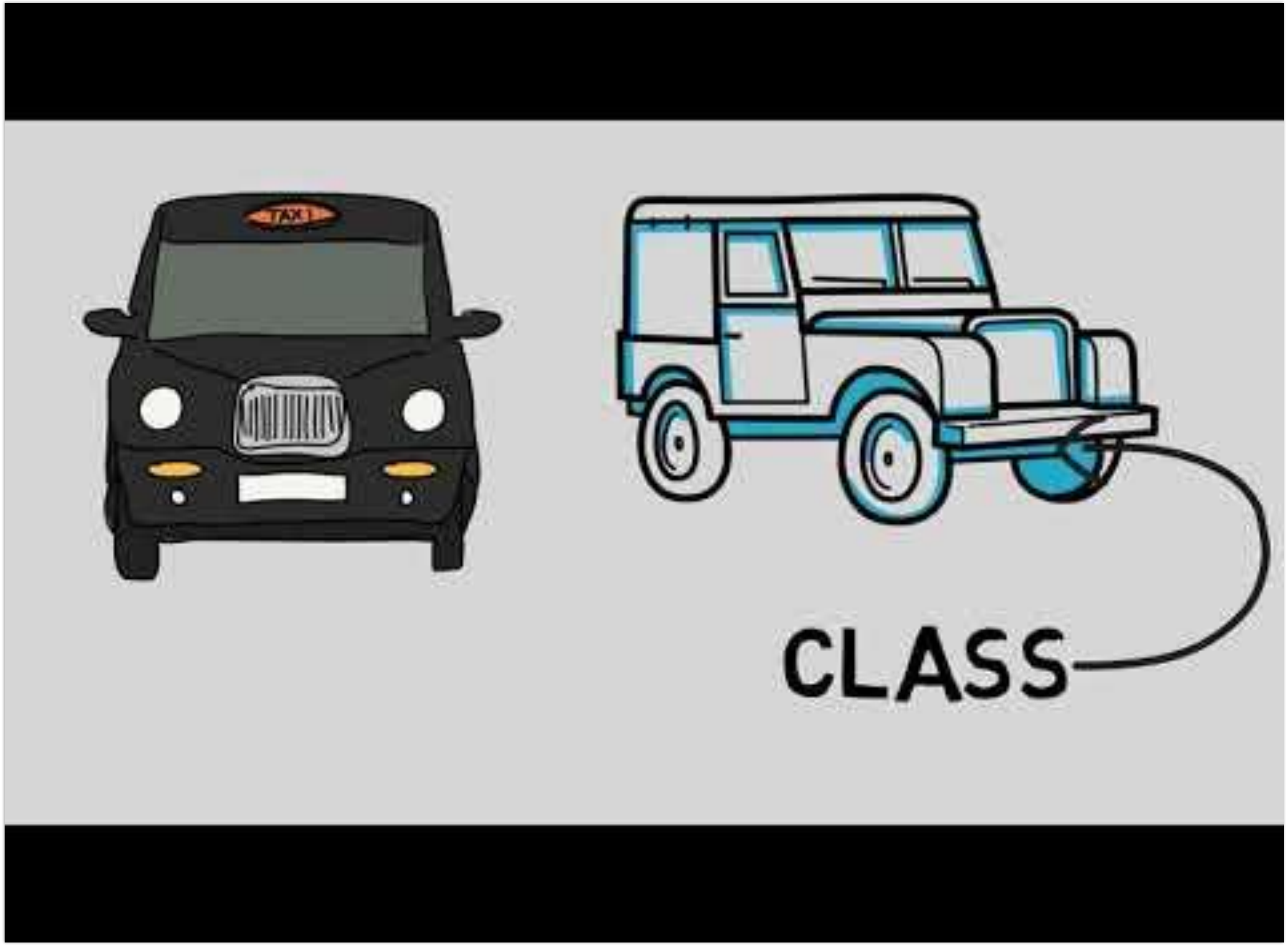
FINAL

- A class whose subclasses cannot be created is called a final class.

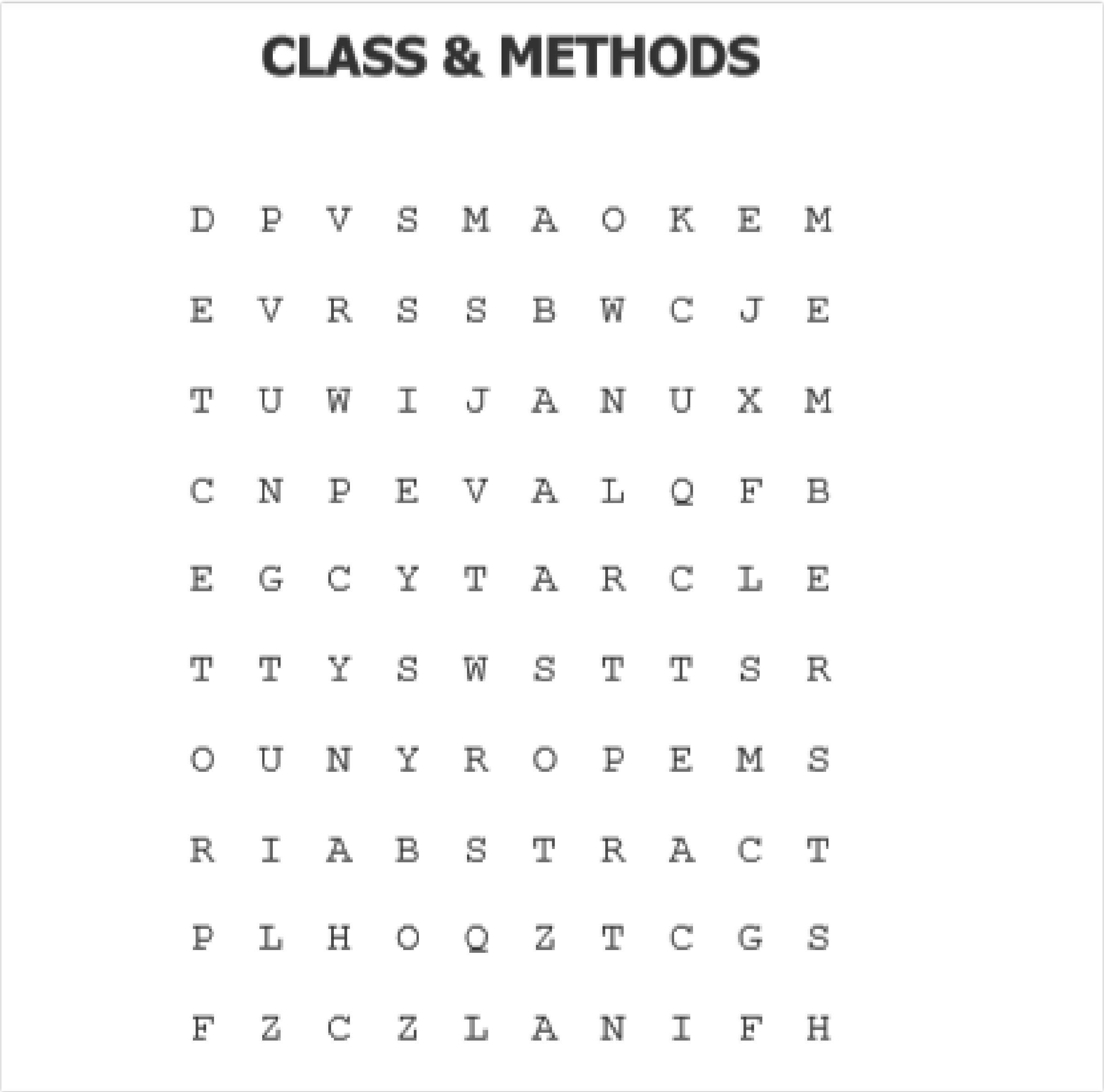
ABSTRACT

- An **abstract class** is a **class** that is declared **abstract-** it may or may not include abstract method.
- **Abstract classes** cannot be instantiated, but they can be subclassed.

VIDEO: https://www.youtube.com/watch?v=pKWRin7P4yk



PUZZLE



CLASS & METHODS

D	P	V	S	M	A	O	K	E	M
E	V	R	S	S	B	W	C	J	E
T	U	W	I	J	A	N	U	X	M
C	N	P	E	V	A	L	Q	F	B
E	G	C	Y	T	A	R	C	L	E
T	T	Y	S	W	S	T	T	S	R
O	U	N	Y	R	O	P	E	M	S
R	I	A	B	S	T	R	A	C	T
P	L	H	O	Q	Z	T	C	G	S
F	Z	C	Z	L	A	N	I	F	H

Thanks!