



# Java classes

We can create a custom type by defining a class. The class definition serves as a blueprint for all instances of the class.

```
public class Book {
```

**Class declaration**

Here, we declare the name of the class.

```
private String title ;
private int numPages ;
```

**Class attributes**

the variables that belong to the class.

These fields can potentially be accessed and/or manipulated.

```
public Book (String t, int p) {
    this.title = t ;
    this.numPages = p ;
}
```

**Constructor(s)**

Here, we define the method(s) that initialize a new instance of the class. We can overload constructors - define multiple constructors with different parameters.

```
public String getTitle ( ) {
    return this.title ;
}
```

```
public void setTitle (String t) {
    this.title = t ;
}
```

If we declare our class attributes as private, we'll need to provide public functions to read and write the values.

**Getters**

Returns the value of the field  
Consumes no parameters

```
public String getNumPages ( ) {
    return this.numPages ;
}
```

**setters**

void method that consumes a parameter.

Overwrites an attribute's value with the parameter value provided.

```
public void setNumPages (int p) {
    this.numPages = p ;
}
```

```
public String toString ( ) {
    return this.title + " : " + String.valueOf (this.numPages) ;
}
```

**class methods**

```
public static void main (String [] args) {
    Book md = new Book ("Moby Dick", 250) ;
    md.setNumPages (130) ;
    System.out.println (md.toString ( ) ) ;
}
```

**main method**

Java method that serves as the entry point of the program  
required signature of main method

```
public static void main (String [] args)
```