

Java without BlueJ

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

Java without BlueJ

BlueJ is an *Integrated Development Environment* (IDE) for Java.

BlueJ provides an *editor* and access to the standard Java *compiler* and *run-time system*. Its innovation lies in its capabilities for creating objects and invoking methods on them *interactively* (supporting the passing of arguments and return of results). This works through a graphical representation of *classes* and *object* instances and simple *point-and-click* operation. This lets us immediately start *testing* newly written classes *without having to write any testing code*. *Inspectors* let us see *running results*.

Additionally, BlueJ provides a mechanism for *recording* a series of interactive tests on a class together with user-defined *correct results* for those tests. These recordings can be *re-run* (with the results checked automatically) on the class each time the class code is changed – *regression testing*. *As code is continually being maintained, this is a huge help!*

Java without BlueJ

But what if we don't have **BlueJ**?

How do we get anything written, compiled and run?

How do we test stuff?

Write (using any *editor*) each Java class in a separate file, whose name is the class name with the suffix: **.java**

For example, the **Date** class goes in the file **Date.java**

To compile, we need a *command window* (e.g. **Command Prompt** in Windows Accessories – or any Unix window).

Change directory (**cd**) to wherever the **.java** files are.

Java without BlueJ

But what if we don't have **BlueJ**?

How do we get anything written, compiled and run?

How do we test stuff?

Change directory (**cd**) to wherever the **.java** files are.

To compile, use the command: **javac Date.java**

If no errors, this produces the file: **Date.class**

.class files contain *Java byte code*. Don't try to look at it
– these files don't hold anything humans can read!

Java without BlueJ

But what if we don't have **BlueJ**?

How do we get anything written, compiled and run?

How do we test stuff?

.class files contain *Java byte code*. Don't try to look at it – these files don't hold anything humans can read!

How do we create class *objects* and execute their *methods*?

The standard **Java Development Kit (JDK)** provides no way (outside of a **Java** program) to construct *objects*! But it does have a way to invoke a (*very particular*) **static** method – for which, of course, we only need the *class* (not an *object*).

Java without BlueJ

```
class DemoMain {  
  
    public static void main (String[] args)  
    {  
  
    }  
  
}
```

Only a **static** method called **main**, *with exactly the above signature*, can be run with the standard command.

Java without BlueJ

```
class DemoMain {  
  
    /**  
     * Print arguments supplied to this program.  
     */  
    public static void main (String[] args)  
    {  
        for (String s : args) {  
            System.out.println (s);  
        }  
    }  
}
```

Only a **static** method called **main**, *with exactly the above signature*, can be run with the standard command.

Java without BlueJ

```
class DemoMain {  
  
    /**  
     * Print arguments supplied to this program.  
     */  
    public static void main (String[] args)  
    {  
        for (String s : args) {  
            System.out.println (s);  
        }  
    }  
}
```

```
% javac DemoMain.java
```

```
% java DemoMain
```

compile

invoke main

Java without BlueJ

```
class DemoMain {  
  
    /**  
     * Print arguments supplied to this program.  
     */  
    public static void main (String[] args)  
    {  
        for (String s : args) {  
            System.out.println (s);  
        }  
    }  
}
```

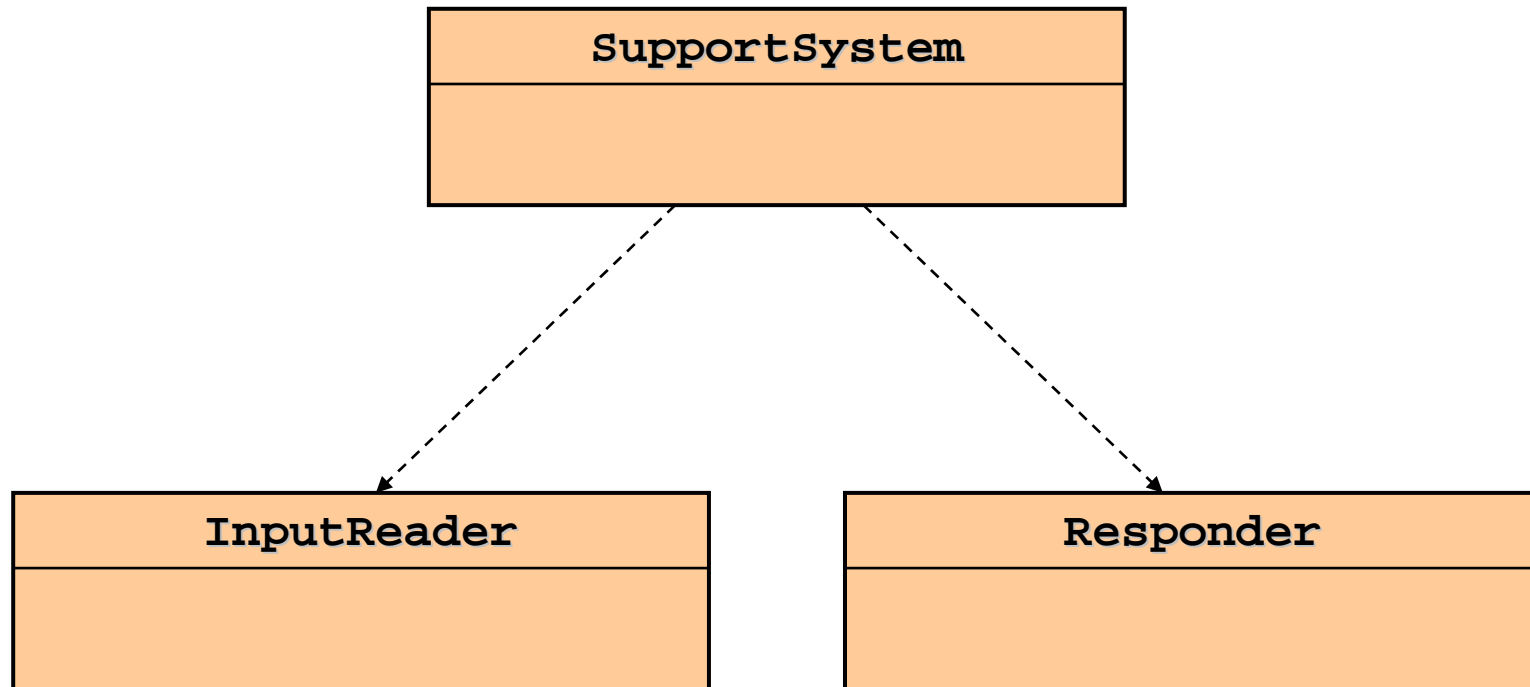
compile

```
% javac DemoMain.java  
% java DemoMain  
  
% java DemoMain one two:a-b -three  
one  
two:a-b  
-three  
  
%
```

invoke main

Java without BlueJ

For example, the **BlueJ** project *tech-support-complete*:



With **BlueJ**, we make a **SupportSystem** object, then click the **start()** method.

Java without BlueJ

For example, the **BlueJ** project *tech-support-complete*:

With **BlueJ**, we make a **SupportSystem** object, then click the **start()** method. Without **BlueJ**, we have to program:

```
class TechSupportMain {  
  
    /**  
     * Print arguments supplied to the program  
     */  
    public static void main (String[] args)  
    {  
        new SupportSystem ().start ();  
    }  
  
}
```

Java without BlueJ

```
class TechSupportMain {  
  
    /**  
     * Print arguments supplied to the program  
     */  
    public static void main (String[] args)  
    {  
        new SupportSystem ().start ();  
    }  
  
}
```

```
% javac TechSupport.java  
% java TechSupport  
Welcome to the DodgySoft Technical Support System.
```

```
Please tell us about your problem.  
We will assist you with any problem you might have.  
Please type 'bye' to exit our system.  
> Your software is a load of rubbish ...  
I need a bit more information on that.
```

compile

invoke main

Review

Java without BlueJ

javac

The *compiler* from Sun's standard **Java Development Kit**. This compiles **.java** files to **.class** files.

java

Launches the **Java Virtual Machine** (JVM), which invokes the **main** method of the *class* to which it is applied. We give it the *name* of the class only (no **.class** suffix).

The **main** method of the *class* to which **java** is applied must have the header:

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