**Using two classes in Java program**

**class** Computer {

Computer() {

System.out.println("Constructor of Computer class.");

}

**void** computer\_method() {

System.out.println("Power gone! Shut down your PC soon...");

}

**public** **static** **void** main(String[] args) {

Computer my = **new** Computer();

Laptop your = **new** Laptop();

my.computer\_method();

your.laptop\_method();

}

}

**class** Laptop {

Laptop() {

System.out.println("Constructor of Laptop class.");

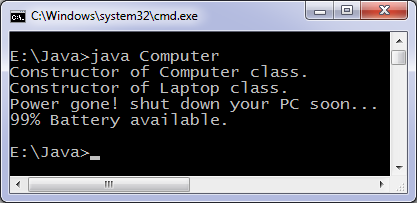
}

**void** laptop\_method() {

System.out.println("99% Battery available.");

}

}



## Java constructor example

**class** Programming {

*//constructor method*

Programming() {

System.out.println("Constructor method called.");

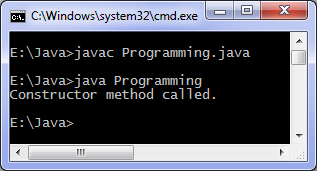
}

**public** **static** **void** main(String[] args) {

Programming object = **new** Programming(); *//creating object*

}

}



## Java constructor overloading

**class** Language {

String name;

Language() {

System.out.println("Constructor method called.");

}

Language(String t) {

name = t;

}

**public** **static** **void** main(String[] args) {

Language cpp = **new** Language();

Language java = **new** Language("Java");

cpp.setName("C++");

java.getName();

cpp.getName();

}

**void** setName(String t) {

name = t;

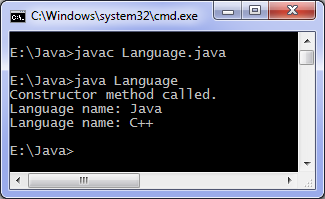
}

**void** getName() {

System.out.println("Language name: " + name);

}

}



## Java constructor chaining

**class** GrandParent {

**int** a;

GrandParent(**int** a) {

**this**.a = a;

}

}

**class** Parent **extends** GrandParent {

**int** b;

Parent(**int** a, **int** b) {

**super**(a);

**this**.b = b;

}

**void** show() {

System.out.println("GrandParent's a = " + a);

System.out.println("Parent's b = " + b);

}

}

**class** Child {

**public** **static** **void** main(String[] args) {

Parent object = **new** Parent(8, 9);

object.show();

}

}

