**Java program to demonstrate the uses of classes, objects, and the object-oriented pillars in Java**

**package** oops;

**abstract** **class** Mobile {

**private** String name;

**abstract** **void** turnOn();

**abstract** **void** turnOff();

**public** Mobile(String name) {

**this**.name = name;

}

**public** String getName() {

**return** **this**.name;

}

}

**class** Redmi **extends** Mobile {

**public** Redmi(String name) {

**super**(name);

}

**public** **void** turnOn() {

System.***out***.println("Turning on Redmi");

}

**public** **void** turnOff() {

System.***out***.println("Turning off Redmi");

}

@Override

**public** String toString() {

**return** "SmartPhone Model: "+getName();

}

}

**class** OnePlus **extends** Mobile {

**public** OnePlus(String name) {

**super**(name);

}

**public** **void** turnOn() {

System.***out***.println("Turning on OnePlus");

}

**public** **void** turnOff() {

System.***out***.println("Turning off OnePlus");

}

@Override

**public** String toString() {

**return** "SmartPhone Model: "+getName();

}

}

**class** Parent {

**public** **void** display() {

System.***out***.println("Parent Display with no arguments");

}

**public** **void** display(String message) {

System.***out***.println("Parent Message: "+message);

}

}

**public** **class** oopsConcepts **extends** Parent {

**public** **void** display() {

System.***out***.println("Child Display with no arguments");

**super**.display();

}

**public** **void** display(String message) {

**super**.display(message);

System.***out***.println("Child Message: "+message);

}

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

oopsConcepts obj1 = **new** oopsConcepts();

obj1.display();

obj1.display("Hello everyone");

Redmi r = **new** Redmi("Redmi Note 8 pro");

OnePlus o = **new** OnePlus("OnePlus nord 2");

r.turnOn();

r.turnOff();

System.***out***.println(r+"\n");

o.turnOn();

o.turnOff();

System.***out***.println(o+"\n");

}

}