**Java Factory Pattern**

**import** java.util.\*;

**import** java.security.\*;

**interface** Food {

**public** String getType();

    }

**class** Pizza **implements** Food {

**public** String getType() {

**return** "Someone ordered a Fast Food!";

     }

    }

**class** Cake **implements** Food {

**public** String getType() {

**return** "Someone ordered a Dessert!";

     }

    }

**class** FoodFactory {

**public** Food getFood(String order) {

**if**(order.equalsIgnoreCase("Cake")) {

**return** **new** Cake();

             }**else** {

**return** **new** Pizza();

             }

}*//End of getFood method*

    }*//End of factory class*

**public** **class** Solution {

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

            Do\_Not\_Terminate.forbidExit();

**try**{

            Scanner sc=**new** Scanner(System.in);

*//creating the factory*

            FoodFactory foodFactory = **new** FoodFactory();

*//factory instantiates an object*

            Food food = foodFactory.getFood(sc.nextLine());

            System.out.println("The factory returned "+food.getClass());

            System.out.println(food.getType());

        }

**catch** (Do\_Not\_Terminate.ExitTrappedException e) {

            System.out.println("Unsuccessful Termination!!");

        }

     }

    }

**class** Do\_Not\_Terminate {

**public** **static** **class** ExitTrappedException **extends** SecurityException {

**private** **static** **final** **long** serialVersionUID = 1L;

        }

**public** **static** **void** forbidExit() {

**final** SecurityManager securityManager = **new** SecurityManager() {

                @Override

**public** **void** checkPermission(Permission permission) {

**if** (permission.getName().contains("exitVM")) {

**throw** **new** ExitTrappedException();

                    }

                }

            };

            System.setSecurityManager(securityManager);

        }

    }