The Factory Pattern is a creational design pattern that provides an interface for creating objects,

but delegates the decision of which class to instantiate to subclasses.

It promotes loose coupling between the client code and the created objects by abstracting the object creation process.

The Factory Pattern is particularly useful when you have a group of related objects that need to be created based on certain criteria or configuration.

*public abstract class* Computer {   
  
 *public abstract* String getRAM();  
 *public abstract* String getHDD();  
 *public abstract* String getCPU();  
  
 @Override  
 *public* String toString(){  
 *return* "RAM= "+*this*.getRAM()+", HDD="+*this*.getHDD()+", CPU="+*this*.getCPU();  
 }  
}

*public class* ComputerFactory {  
  
 *public static* Computer getComputer(String type, String ram, String hdd, String cpu){  
 *if*("PC".equalsIgnoreCase(type))  
 *return new* PC(ram, hdd, cpu);  
 *else if*("Server".equalsIgnoreCase(type))  
 *return new* Server(ram, hdd, cpu);  
  
 *return null*;  
 }  
}

*public class* PC *extends* Computer {  
  
 *private* String ram;  
 *private* String hdd;  
 *private* String cpu;  
  
 *public* PC(String ram, String hdd, String cpu){  
 *this*.ram=ram;  
 *this*.hdd=hdd;  
 *this*.cpu=cpu;  
 }  
 @Override  
 *public* String getRAM() {  
 *return this*.ram;  
 }  
  
 @Override  
 *public* String getHDD() {  
 *return this*.hdd;  
 }  
  
 @Override  
 *public* String getCPU() {  
 *return this*.cpu;  
 }  
  
}

*public class* Server *extends* Computer {  
  
 *private* String ram;  
 *private* String hdd;  
 *private* String cpu;  
  
 *public* Server(String ram, String hdd, String cpu){  
 *this*.ram=ram;  
 *this*.hdd=hdd;  
 *this*.cpu=cpu;  
 }  
 @Override  
 *public* String getRAM() {  
 *return this*.ram;  
 }  
  
 @Override  
 *public* String getHDD() {  
 *return this*.hdd;  
 }  
  
 @Override  
 *public* String getCPU() {  
 *return this*.cpu;  
 }  
  
}

*public class* TestFactory {  
  
 *public static void* main(String[] args) {  
 Computer pc = ComputerFactory.*getComputer*("pc","2 GB","500 GB","2.4 GHz");  
 Computer server = ComputerFactory.*getComputer*("server","16 GB","1 TB","2.9 GHz");  
 System.*out*.println("Factory PC Config::"+pc);  
 System.*out*.println("Factory Server Config::"+server);  
 }  
  
}

Factory PC Config::RAM= 2 GB, HDD=500 GB, CPU=2.4 GHz

Factory Server Config::RAM= 16 GB, HDD=1 TB, CPU=2.9 GHz