জাভা প্রোগ্রামিং এ ইনহেরিটেন্স নিয়ে একটি ছোট্ট উদাহরন

**01APR**

আমরা জানি, জাভা একটি অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং ল্যাঙ্গুয়েজ। এই অবজেক্ট অরিয়েন্টেড প্রোগ্রামিং এর একটি অন্যতম বৈশিষ্ট্য হল ইনহেরিটেন্স(Inheritance)।

সাধারনভাবে আমরা জানি, ইনহেরিট মানে উত্তরাধিকারী হওয়া বা দখল করা। কথা কম বলি। আগে একটা কোড দিয়ে বুঝাই।

1. **public** **class** Dog {
2. **public** **void** sleep() {
3. System.*out*.println(“ঘুমাইতে পারে”);
4. }
5. **public** **void** eat() {
6. System.*out*.println(“খাইতে পারে”);
7. }
8. }

প্রথমে আমরা একটা ক্লাস তৈরি করলাম Dog নামে। এই ক্লাসের মধ্যে Dog এর কিছু বৈশিষ্ট্য দেয়া হয়েছে । যার ভিতরে দুইটা মেথড এর মধ্যে বলে দিয়েছি Dog  কি কি পারে। ১ম sleep() ম্যাথোডে(3 ও 4 নং লাইনে) বলে দিয়েছি “ঘুমাইতে পারে” এবং ২য় eat() ম্যাথোডে (6 ও 7 নং লাইনে) বলে দিয়েছি “খাইতে পারে”।

এখন প্রয়োজনের জন্য আমাদের হয়ত এমন কিছু বৈশিষ্ট্য লাগতে পারে যেখানে উপরের Dog এর বৈশিষ্ট্যগুলোতো থাকবেই সাথে আরো কিছু এক্সট্রা বৈশিষ্ট্য থাকবে। ধরলাম, একটা Bird, সেও খাইতে পারে, ঘুমাইতে পারে যেটা Dog ও পারে। কিন্তু Bird এর একটা বেশি বৈশিষ্ট্য আছে সেটা হল Bird উড়তেও পারে। এখন Bird নামে যদি আমরা একটা ক্লাস তৈরি করতে চাই তাহলে কি আমরা আবার নতুন করে একটা ক্লাস লিখে তার ভিতরে sleep(),eat() মেথোডগুলো দিয়ে দিবো??? এটা করলে তো একই জিনিস বার বার লেখা হচ্ছে। এতে আমাদের কোডিং এর কমপ্লেক্সটিটি বাড়ছে। এটা না করে আমরা এমন একটা কিছু করতে পারিনা যেটা করলে উপরের Dog ক্লাসের বৈশিষ্ট্যগুলো Bird ক্লাসের মধ্যে অটোমেটিক্যালি প্রবেশ করে যাবে এবং Bird  এর যে এক্সট্রা বৈশিষ্ট্য আছে আমরা শুধু সেটাকেই বলে দিলাম। তাহলে আমাদের কোডিং ও অনেকখানি কমে গেলো। ।ঠিকনা ?? বোঝা যাচ্ছে কি?? ঠান্ডা মাথায় ধীরে ধীরে আমার লিখা গুলো পড়লেই বোঝা যাবে। চলুন, Bird ক্লাসটিও আমরা লিখে ফেলি।

1. **class** Bird **extends** Dog {
2. **public** **void** fly() {
3. System.*out*.println(“এবং পাখি ঊড়তেও পারে।”);
4. }
5. }

এখানে ভাল করে লক্ষ্য করুন, আমরা Bird নামে একটা ক্লাস লিখেছি এবং সেটাকে extends করে দিয়েছি Dog নামে। এই লাইনটির অর্থ হল Bird নামের নতুন ক্লাসটি আগের Dog নামের ক্লাসটির বৈশিষ্ট্যও নিয়ে নিবে। আর তাই আমাদের নতুন করে sleep(), eat() মেথোড গুলো লিখে দিতে হয়নি। আমরা শুধু নতুন বৈশিষ্ট্যটি যোগ করে দিয়েছি নতুন একটা মেথোড তৈরি করে যেমন fly() মেথোড।

এখন যদি আমরা Main মেথোড থেকে Dog এবং Bird ক্লাসগুলোর মেথোডগুলোকে এক্সেস করতে চাই তাহলে সেই কোডটা লিখতে হবে নিচের মত করে।

1. **public** **static** **void** main(String[] args) {
2. Dog dog = **new** Dog();
3. System.*out*.println(“কুকুর “);
4. dog.sleep();
5. dog.eat();
7. System.*out*.println();
9. Bird bird = **new** Bird();
10. System.*out*.println(“পাখি”);
11. bird.sleep();
12. bird.eat();
13. bird.fly();
14. System.*out*.println();
15. }
16. }

এখানে ২ নং লাইনে আমরা Dog ক্লাসের dog নামে একটা অবজেক্ট তৈরি করেছি । তারপর ৪ ও ৫ নং লাইনে dog অবজেক্ট দিয়ে আমাদের Dog ক্লাসের sleep() ও eat() মেথোডকে এক্সেস করেছি। আবার অনুরুপভাবে Bird ক্লাসের bird নামে একটা অবজেক্ট তৈরি করেছি। তারপর একইভাবে bird অবজেক্ট দিয়ে sleep(),eat() ও fly() মেথোডকে এক্সেস করেছি।

আপনি এখানে একটা মজার জিনিস লক্ষ্য করে দেখুন, আমাদের Bird ক্লাসের মধ্যে কিন্তু শুধু একটি মেথোড ছিল যার নাম হল fly() । অথচ আপনাকে sleep()  ও eat() মেথোড এর মধ্যেকার তথ্যগুলোও প্রিন্ট করে দেখাবে এবং সাথে এক্সট্রা fly() মেথোডের ভিতরের তথ্যও। কারন কি???? কারন আমরা Dog ক্লাসকে ইনহেরিট বা দখল করেছি Bird ক্লাসে extends করে। এটাই হল ইনহেরিটেন্স। অর্থাৎ একটি ক্লাস যখন আরেকটি ক্লাসকে ইনহেরিট করে তখন সেটাকে বলে ইনহেরিটেন্স। যে ক্লাসকে ইনহেরিট করা হয় সেই ক্লাসকে বলে সুপার ক্লাস(supper class) আর যে ক্লাস ইনহেরিট করে তাকে বলে সাব ক্লাস (sub class)। আশা করছি আপনি অবজেক্ট অরিয়েন্টেডের একটি খুব খুব গুরুত্তপুর্ন ফিচারটি বুঝে গেছেন।। তারপরেও কোন প্রশ্ন থাকলে অথবা কেমন লাগলো বা কোণ ভুল ভ্রান্তি বা পরামর্শ থাকলে অবশ্যই কমেন্ট এ জানাবেন।

আমি পুরো কোডটি সবার সুবিধার্থে দিয়ে দিলাম। অবশ্যই কোডটি রান করে দেখবেন। কোডটি জাভায় লিখা।।