

আমরা সবাই প্রোগ্রামার

আপনি যে পেশারই মানুষ হোন না কেন, পড়াশুনা যে বিষয়েই হোক না কেন,

যে বয়সেরই হোন না কেন, আপনি প্রোগ্রামিং শিখবেনই!!!

বিশেষত হাইস্কুলের ছেলেমেয়েদের জন্য উপযোগী করে লেখা

লেখক:

নাজমুল হাসান নিরো

এবং

সার্বিক সহযোগিতায়: মোঃ শাহাদাত সরকার

ভূমিকা

কোন না কোনভাবে আমাদের প্রায় সবারই ধারণা প্রোগ্রামিং অনেক কঠিন একটা ব্যাপার। যেন তেন ব্রিলিয়ান্ট ছাড়া প্রোগ্রামিং করা সম্ভব না। আর তাই সফটওয়্যার ইঞ্জিনিয়ার কথাটা শুনলেই চোখ কপালে উঠে যায়।

আমি প্রথমত এই ভুল ধারণাটা ভেঙ্গে ফেলতে চেয়েছি। প্রোগ্রামিং যে বিরাট কঠিন কিছু না এবং হাইস্কুল লেভেলের ছেলেমেয়েরাও প্রোগ্রামিং শুরু করতে পারে, আমি চেষ্টা করেছি সেটাই প্রমাণ করতে। হাইস্কুলের ছেলেমেয়েরা যদি অংক কষতে পারে তাহলে প্রোগ্রামিংও করতে পারবে অবশ্যই। চেষ্টা করেছি বিষয়টা প্রমাণ করতে।

এবং প্রোগ্রামিং যে বিরাট কঠিন কিছু না, অন্য পেশার যে কোন মানুষ যে অন্তত ন্যূনতম ইংরেজী জানে এবং একটু আকটু কম্পিউটার ব্যবহার করতে পারে, তারাও প্রোগ্রামিং করতে পারবে অবশ্যই।

আরেকটা ব্যাপার। সেটা ক্যারিয়ার নিয়ে। CSE তে পড়েও অনেককে জবের জন্য হন্যে হয়ে বেড়াতে হয়। শুধু স্কীলের অভাবে। একারণে আমি চেয়েছি আমাদের ছেলেমেয়েরা হাইস্কুলেই প্রোগ্রামিং এর জগতটাতে ঢুকে পড়ুক। এখন যদি খুব বেশি নাও পারে তবুও আমার মতে ছেলেমেয়েরা ছোটবেলাতেই যদি জগতটাতে অন্তত ঢুকে পড়তে পারে তাহলে ভবিষ্যৎ জীবনে তাদের জব নিয়ে অন্তত দুঃশ্চিন্তায় পড়তে হবে না। নিজের ক্যারিয়ারটা নিজেই গড়ে নিতে পারবে।

আর হাইস্কুলের ছেলেমেয়েদের বলছি। তোমরা আবার এটা ভেবে নিও না যে দুই লাইন শিখে ফেললেই তুমি প্রোগ্রামার হয়ে যাবে। গ্রামে যাদের ৭০-৮০ বিঘা জমি থাকে গ্রামের লোক তাদেরকে বড়লোক বলে। মূসা বিন শমসের এর কাছে কিন্তু তিনি কিছুই না। আবার বিল গেটসের কাছে মূসা বিন শমসের কিছুই না।

ধনীর যেমন লেভেল আছে। প্রোগ্রামারেরও তেমন লেভেল আছে। প্রোগ্রামিং সম্পর্কে তোমাকে একটা গ্রহনযোগ্য নলেজ এবং কার্যপদ্ধতি দেখিয়ে দেওয়াই বইটার উদ্দেশ্য। তবে সে লেভেলটাও একেবারে ফেলনা না। বইটা শেষ করতে পারলে তুমি ছোট ছোট এ্যানড্রয়েড এ্যাপ এবং গেম ডেভেলপ করতে পারবে। কিন্তু মনে রেখো এটা একটা লং জার্নি এবং এই বই শেষ করে তোমাকে থেমে থাকলে চলবে না। এর পরের ধাপের জন্য প্রস্তুতি নিতে হবে।

তাহলে চলো শুরু করি। আমি তোমাদের সাথেই আছি!

বইটির উদ্দেশ্য

তোমরা মোবাইলে গেম খেলেছো নিশ্চয়। Clash of Clans, Angry Bird, Mine Militia এগুলো তো তোমরা সময় পেলেই খেলো। কেমন হয় বলতো তোমরা যদি নিজে মোবাইল গেম বানাতে পারো? তোমরা গেম বানাতে আর তোমাদের বানানো গেম অন্যরা খেলবে?

তাহলে অনেক মজা হবে তাই না?

আবার মোবাইলে তো অনেক এ্যাপও ব্যবহার করো তোমরা। imo, Viber, SHAREit, Music Player এগুলো এ্যাপও তো তোমরা সচরাচর ব্যবহার করো।

কেমন হবে বলতো তোমরা যদি নিজেরা নিজেরা ছোটখাট এ্যাপ বানাতে পারো? যেগুলো অন্যের কাজে আসবে। আবার সেগুলো Google Play Store এ বিক্রিও হতে পারে।

অনেক মজা হবে কি বল?

কিন্তু এগুলো বানাতে গেলে তোমাকে কম্পিউটার প্রোগ্রামিং শিখতে হবে। কারণ এই যে মোবাইলের জন্য এ্যাপ, গেম এইগুলো সবই বানানো হয় কম্পিউটার প্রোগ্রামিং করে।

এখন তো আইটি অনেক এগিয়ে গেছে। তোমরাও তো টিভি দেখো, পত্রিকা পড়ো। অনেকেই হয়তো কম্পিউটার প্রোগ্রামিং কথাটা শুনে থাকবে। অনেকের হয়তো বড় ভাই/বোন বা আংকেল আছেই যারা প্রোগ্রামার বা সফটওয়্যার ইঞ্জিনিয়ার। তাদের কাছ থেকেও শুনে থাকতে পারো।

তবে শুনে না থাকলেও অসুবিধা নেই। আমরা এই বইয়ে সব কিছুই একে একে আলোচনা করবো এবং হাতে কলমে শিখবো।

প্রস্তুতি পর্ব

যেমন ধরো অংক কষতে বসার পূর্বে খাতা কলম নিতে হয়, কোন অংক কষবে সেটা ঠিক করতে হয়, সেরকমই প্রোগ্রামিং শেখা শুরু করার পূর্বে আমাদের কিছু পূর্ব প্রস্তুতি নিতে হবে। যেমন কিছু বেসিক জিনিস আগে জানতে হবে আর তোমার কম্পিউটারে কিছু সফটওয়্যার ইন্সটল করতে হবে। চলো আমরা তাহলে সেগুলো সেরে নিই।

প্রোগ্রামিং(Programming) কি?

আমরা সহজ কথায় আগে শুরু করি। আমরা যে গেম বা এ্যাপ বানাতে চাচ্ছি সেসব বানাতে গেলে প্রোগ্রামিং করেই বানাতে হয়। অর্থাৎ প্রোগ্রামিং হলো এমন একটা কাজ যেটা করে তুমি গেম বানাতে পারবে, এ্যাপ বা সফটওয়্যার বানাতে পারবে, হ্যাকারও হতে পারবে 😊 ।

আবার আরেক কথায় বলা যায় কম্পিউটার প্রোগ্রাম লেখার কাজটাই হলো প্রোগ্রামিং। তাহলে আবার প্রশ্ন এসে যায় কম্পিউটার প্রোগ্রাম কি? চলো আমরা সেটাও ব্যাখ্যা করছি।

কম্পিউটার প্রোগ্রাম(Computer Program) কি?

এটাও আমরা সহজ কথায় শুরু করি। ধরো তুমি কম্পিউটারে একটা গেম খেলছো। সেখানে তোমার নায়কের হাতে একটা বন্দুক আছে আর তুমি যদি মাউসের বাম বাটনটা ক্লিক করো তাহলে তোমার নায়কের বন্দুক থেকে একটা গুলি বের হয়।

তার অর্থ হচ্ছে কম্পিউটারকে বলে দেওয়া আছে যে তুমি যখনই মাউসের বাম বাটনে ক্লিক করবে সাথে সাথে একটা গুলি বের হবে।

এই যে বলে দেয়া, এটা হচ্ছে কম্পিউটারের প্রতি নির্দেশনা বা Instruction।

আর এই নির্দেশনা বা Instruction দেয়া হয় কম্পিউটার প্রোগ্রামের মাধ্যমে। অর্থাৎ **কম্পিউটার প্রোগ্রাম হল কম্পিউটারের প্রতি কিছু নির্দেশনা যে তাকে কখন কি করতে হবে।**

আবার যেমন ধরো তুমি মাউসটা একটু ডানে ঘোরালে তোমার নায়ক ডানে ঘুরে যায়, তুমি কীবোর্ড থেকে W চাপলে দৌড়াতে শুরু করে, এগুলো সব ইন্সট্রাকশনই কম্পিউটারকে বলে দেয়া আছে কম্পিউটার প্রোগ্রামের মাধ্যমে।

আবার ধরো তুমি একটা ভিডিও দেখছো। কীবোর্ড থেকে Space Bar চাপলে ভিডিওটা Pause হয়ে যায়, Enter চাপলে ফুলস্ক্রীন হয়ে যায়, এগুলো সমস্ত ইন্সট্রাকশনই কম্পিউটারকে বলে দেয়া আছে কম্পিউটার প্রোগ্রামের মাধ্যমে।

কম্পিউটারকে প্রোগ্রামের মাধ্যমে বলে দেয়া আছে যে যখন Space Bar চাপলে সে ভিডিও বন্ধ করবে।

আর এটা শুধু কম্পিউটারের ক্ষেত্রে না, তোমার হাতের মোবাইলের ক্ষেত্রেও। মোবাইলকেও প্রোগ্রামের মাধ্যমে বলে দেয়া আছে যে একটা বার্ড কে টেনে ছেড়ে দিলে সেটা উড়ে গিয়ে সামনে থাকা বরফের খন্ডটা ভেঙ্গে পড়বে, তুমি কিছু লিখে Send এ চাপ দিলে তোমার লেখাটা অপর পাশের জনের কাছে চলে যাবে এগুলো সব ইন্সট্রাকশনও প্রোগ্রামের মাধ্যমে বলে দিতে হয়।

চলো আমরা একটা প্রোগ্রাম দেখি:

```
public class AmraSobaiProgrammer {  
  
    public static void main(String args[]){  
        int number = 12;  
  
        int squareValue = number * number;  
  
        System.out.println("The square value is: " + squareValue);  
    }  
  
}
```

এটা খুব ছোট একটা প্রোগ্রাম। এই প্রোগ্রামটি তোমাকে কোন একটি সংখ্যার Square ভ্যালু বের করে দেবে। নির্দেশনার উপর ভিত্তি করে এরকম প্রোগ্রাম এক লাইন, দশ লাইন থেকে শুরু করে হাজার বা লাখ লাইনেরও হতে পারে। নির্ভর করবে কতটুকু কাজ কম্পিউটারকে দিয়ে করাতে চাও তুমি। যত বেশি কাজ করাতে চাইবে তত বেশি লাইনের কোড লিখতে হবে।

প্রোগ্রামিং ল্যাঙ্গুয়েজ (Programming Language)

তোমার বন্ধু তোমাকে বললো “আমি যখন বলবো ‘বাম’, তখন তুমি বামে ঘুরবে।”

তোমার বন্ধুটি তোমাকে একটি নির্দেশনা দিল। কিন্তু কোন ভাষায় দিল? বাংলাতে না? কারণ তুমি বাংলা জানো বলেই তোমাকে বাংলায় ইন্সট্রাকশন দেওয়া হলে তুমি বুঝতে পারো। এবং সেটা পালন করতে পারো। কিন্তু তুমি যদি জাপানি ভাষা না জানো আর তোমাকে যদি জাপানি ভাষায় কোন ইন্সট্রাকশন দেয়া হয় তুমি কি বুঝবে? তুমি সেটা বুঝতেও পারবে না সেটা পালনও করতে পারবে না।

কম্পিউটারের ক্ষেত্রেও তেমনি তুমি যদি তাকে ইন্সট্রাকশন দাও তাহলে তাকে তার ভাষাতেই তোমাকে ইন্সট্রাকশন দিতে হবে। এর অর্থ হচ্ছে তুমি যখন তার জন্য একটি ইন্সট্রাকশন তৈরী করবে বা প্রোগ্রাম লিখবে তখন কম্পিউটার যেভাবে বোঝে ঠিক সেইভাবেই তোমাকে প্রোগ্রাম লিখতে হবে।

যেমন ধরো তুমি চাইলে কম্পিউটারকে দিয়ে ৭৮ আর ৯৩ যোগ করিয়ে নেবে এবং তার জন্য একটি একটি প্রোগ্রাম লিখলে এরকম “৭৮ আর ৯৩ যোগ করে দেখাও”। কম্পিউটার মোটেও তোমার এই নির্দেশনা পালন করতে পারবে না। এরকম একটি যোগের প্রোগ্রাম লিখতে গেলে তোমাকে লিখতে হবে প্রোগ্রামিং ল্যাঙ্গুয়েজে।

অন্য কোন ভাবে তাকে নির্দেশনা দিলে সে বুঝতেও পারে না, পালনও করতে পারে না।

কিন্তু কম্পিউটার আবার অনেকগুলো ভাষা বুঝতে পারে। মানে কম্পিউটার প্রোগ্রাম লেখারও অনেকগুলো ভাষা রয়েছে। যেমন: C, C++, C#, Java, PHP, JavaScript, Python, Ruby etc. এই সবগুলোই মূলত একেকটা প্রোগ্রামিং ল্যাঙ্গুয়েজ। অর্থাৎ তুমি যদি এসবের যে কোন একটি ল্যাঙ্গুয়েজে প্রোগ্রাম লেখো তাহলে কম্পিউটার সেটা বুঝতে পারবে এবং পালন করতে পারবে।

এবং আরেকটা ব্যাপার হচ্ছে উপরের প্রত্যেকটি ল্যাঙ্গুয়েজেই প্রোগ্রাম লেখার নিয়ম কিন্তু আলাদা। তুমি যে ল্যাঙ্গুয়েজে প্রোগ্রাম লিখবে সেটার নিয়ম মেনেই তোমাকে প্রোগ্রাম লিখতে হবে।

যেমন তুমি বাংলায় বলো “আমি ভাত খাই”। আর ইংরেজীতে বলো “I eat rice.”

এক ভাষায় অবজেক্ট লিখছো মাঝখানে আরেক ভাষায় অবজেক্ট লিখছো শেষে। কতকটা এরকম আলাদা কিছু নিয়ম কানুন আর কি। কিন্তু আপাতত তোমাকে এটা নিয়ে চিন্তিত হতে হবে না। কারণ আমরা যে উদ্দেশ্য নিয়ে এগোচ্ছি তার জন্য উপরের সবগুলো ল্যাঙ্গুয়েজের মধ্যে আমাদের শুধু Java শিখলেই চলবে।

.....

পুরো প্ল্যান

আমাদের পুরো প্ল্যানটি খুব সরল।

- প্রথমে জাভা প্রোগ্রামিং শিখবো
- তারপর এ্যানড্রয়েড এ্যাপ ডেভেলপমেন্ট শিখবো
- নিজ হাতে একটি এ্যানড্রয়েড গেম বানাবো
- নিজ হাতে একটি এ্যানড্রয়েড এ্যাপ বানাবো

প্রোগ্রামিং এর জগতটাতে ঢুকে পড়ার জন্য আমাদের প্রস্তুতি পর্ব শেষ। এবার এসো আমরা সরাসরি ঢুকে পড়বো প্রোগ্রামিং এর মূল জগতটাতে।

আর এখানে আমাদের যাত্রাটা শুরু হবে “ডাটা” বুঝার মাধ্যমে। কারণ একটা কম্পিউটার প্রোগ্রাম যা করে তার সবই মূলত ডাটাকে ঘিরেই। সেকারণেই আমাদেরকে শুরু করতে হবে “ডাটা” দিয়েই এবং এরপর আমরা স্টেপ বাই স্টেপ এগিয়ে যাবো।

ডাটা:

ধরো তোমাকে জিজ্ঞেস করা হলো তোমার নাম কি?

তুমি বললে “Pragga Mashiat”

এই যে Pragga Mashiat এটিই হলো একটি ডাটা।

আবার জিজ্ঞেস করা হলো তোমার বয়স কত?

তুমি বললে “15”

এটিও একটি ডাটা।

জেএসসি তে তোমার পয়েন্ট কত ছিল?

তুমি বললে “4.85”

এটিও একটি ডাটা।

তোমার যমজ বোন আছে?

তুমি বললে “YES”

এটিও একটি ডাটা।

মূল বিষয়টা হলো তোমার কাছে যা তথ্য আছে তার সবকিছুই এক একটি ডাটা। আর কম্পিউটার প্রোগ্রাম মূলত যা করে তার সব কিছুই এই ডাটাকে ঘিরেই। সেটা বোঝার জন্য চলো একটা উদাহরণ দেখি।

ধরো তোমার হাতে একটি ক্যালকুলেটর আছে।

তুমি ক্যালকুলেটরে 4 এ চাপ দিলে

তারপর + এ চাপ দিলে

এবার চাপ দিলে 5 এ

সবশেষে = তে চাপ দেয়া মাত্র ক্যালকুলেটরের স্ক্রীণে 9 ভেসে উঠবে।

এখানে 4, 5 এবং 9 সবগুলোই এক-একটি ডাটা। ক্যালকুলেটরটি 4 এবং 5 এই দু'টি ডাটা তোমার কাছ থেকে নিয়ে সেটাকে প্রসেস করে 9 ডাটাটি তোমাকে দিয়েছে।

তুমি যদি কম্পিউটারে প্রোগ্রামিং করে যোগের জন্য একটি সফটওয়্যার বানাও সেটিও ঠিক একইভাবে তোমার কাছ থেকে 4 এবং 5 নেবে এবং দু'টাকে যোগ করে তোমাকে 9 প্রদান করবে।

আর এটি শুধু যোগের বেলাতেই না, অন্য সকল ক্ষেত্রেও। সব ক্ষেত্রেই কম্পিউটার ডাটা গ্রহন করে এবং সেটাকে প্রসেস করে প্রসেসকৃত ডাটাটি প্রদান করে। আর একারণেই আমরা বলেছি যে প্রোগ্রামিং এর জগতে চুকে পড়ার শুরুতেই আমাদের ডাটা বোঝাটা গুরুত্বপূর্ণ।

এবং আমরা ইতোমধ্যে দেখে ফেলেছি যে: Pragma Mashiat, 15, 4.85, YES সবগুলোই এক একটি ডাটা।

ভেরিয়েবল(Variable) কি?

তোমাকে একটি বালতি আর পানি দেয়া হলো এবং বলা হলো বালতিতে পানি রাখো। তাহলে বালতিটা কি হলো? একটি কন্টেইনার না? যেটি পানি ধারণ করতে পারে। তেমনি প্রোগ্রামিং এ ভেরিয়েবলও হলো একটি কন্টেইনার যেটি ডাটা ধারণ করে। বালতিতে যেমন তুমি পানি রাখতে পেরেছ তেমনি একটি ভেরিয়েবল এর মধ্যে তুমি ডাটা রাখতে পারবে।

যেমন তোমার বয়স 15 এটি একটি ডাটা এবং এটি তুমি একটি ভেরিয়েবলের মধ্যে রেখে দিতে পারবে। প্রোগ্রামিং এর ক্ষেত্রে ভেরিয়েবলও খুব গুরুত্বপূর্ণ একটি জিনিস।

এসো দেখি ভেরিয়েবল কেন এতো গুরুত্বপূর্ণ কম্পিউটার প্রোগ্রামিং এর ক্ষেত্রে।

ধরো তোমাকে বলা হলো 4 আর 5 যোগ করে দিতে। তুমি ধুম করে বলে দিতে পারবে 9। কিন্তু যদি তোমাকে বলা হয় 81837485625677 এবং 3958792387492 যোগ করে দাও তো। তুমি তখন কিন্তু এটি আর মনে রাখতে পারবে না। তোমাকে দু'টো নাম্বারই খাতায় লিখে রাখতে হবে।

আসলে 4 এবং 5 যোগ করার সময়ও তুমি একই কাজ করেছিলে। দুইটা নাম্বারই তুমি তোমার ব্রেইনে রেখে দিয়েছিলে তারপর যোগ করে ফলাফল বের করে দিয়েছিলে। এবার মনে রাখতে পারছো না জন্য খাতার সাহায্য নিচ্ছে। আসলে প্রতিটি ক্ষেত্রেই তুমি ডাটা দু'টো কোথাও না কোথাও রেখে দিচ্ছে এবং শেষে যোগ করে দিচ্ছে।

কম্পিউটার প্রোগ্রামও একই কাজ করে। সেটা করে ভেরিয়েবল এর মাধ্যমে। যখনই তুমি তাকে একটি ডাটা দাও সেই ডাটাকে তার কোথাও না কোথাও রেখে দিতে হবে। সেই রেখে দেওয়ার জায়গাটাই হলো ভেরিয়েবল।

আগে আমরা ক্যালকুলেটরের মাধ্যমে 4 এবং 5 যোগ করার একটা উদাহরণ দেখেছিলাম। সেই একই কাজ করার জন্য আমরা একটি কম্পিউটার প্রোগ্রাম বানানোর কথাও বলেছিলাম।

এরকম একটি প্রোগ্রাম যদি তুমি বানাতে চাও যেটি 4 এবং 5 যোগ করবে তাহলে এই দুটি ডাটাই তোমাকে দু'টি ভেরিয়েবল এর মধ্যে রাখতে হবে।

মেথড (Method) কি?

ধরো প্রাচী নুডলস রান্না করতে পারে, টাকা গুনতে পারে আর নাচতে পারে। এগুলো হলো প্রাচীর ক্যাপাবিলিটি। অর্থাৎ প্রাচী এই সমস্ত কাজ করতে পারে।

ঠিক এই এরকম-ই কম্পিউটার প্রোগ্রাম যদি কিছু পারতে চায় তাহলে তার মেথড থাকতে হয়। অর্থাৎ তুমি যদি চাও তোমার প্রোগ্রাম কিছু একটা করতে পারবে। যেমন ধরো 12 আর 19 এর যোগফল বের করতে পারবে তোমার প্রোগ্রাম, তাহলে তোমার প্রোগ্রামে একটা মেথড তৈরী করতে হবে।

মানুষের যেমন বিভিন্ন কিছু করার ক্ষমতা আছে, তেমনি একটি কম্পিউটার প্রোগ্রামের কোন কিছু করতে পারার ক্ষমতাটাই মেথড। আমরা যদি বলি প্রাচী নাচতে পারে, তাহলে প্রাচীকে আগেই নাচ শিখতে হবে। আর তুমি যদি বলো তোমার প্রোগ্রাম 12 আর 19 এর যোগফল বের করতে পারে তাহলে তোমাকে অবশ্যই 12 আর 19 এর যোগফল বের করতে পারে এরকম একটি মেথড তৈরী করতে হবে তোমার প্রোগ্রামে।

মেথড (Method) তৈরী করা

মেথড তৈরী করাটা খুব সহজ। বেসিক সিনট্যাক্সটা দেখো:

```
modifier return_type method_name(){  
  
    method_body  
  
}
```

অর্থাৎ:

- **Modifier:** এটা আমরা পরে ব্যাখ্যা করবো। আপাতত এখানে শুধু public লিখে দেবো
- এরপর একটা স্পেস
- তারপর একটি কীওয়ার্ড(Keyword) static (Keyword কী এবং static কী দুটোই আমরা পরে ব্যাখ্যা করবো), অবশ্য এটা অপশনাল(Optional), কিন্তু এই মুহূর্তে ব্যবহার করো। অপশনাল কিভাবে হবে তা আমরা পরে বুঝবো।
- এরপর একটা স্পেস
- **Return Type:** এটা নিয়েও আপাতত চিন্তিত হইও না, পরের চ্যাপ্টারেই এটা ব্যাখ্যা করবো, আপাতত আমরা এখানে void লিখবো

- তারপর একটা স্পেস
- তারপর মেথডের নাম
- তারপর () {
- তারপর মেথডের বডি: এখানে মূলত সেই কোড বা কোডগুলো থাকবে যার মাধ্যমে তুমি বলে দিবে যে তোমার মেথড কি কি করতে পারবে
- এবং সব শেষে একটি }

মেথড(Method) কল(Call) করা

প্রাচী নাচতে পারে। কিন্তু প্রজ্ঞা নাচার জন্য প্রাচীকে অনুরোধ না করলে প্রাচী নাচে না। যেমন: প্রজ্ঞা বললো, প্রাচী একটু নাচো তো। প্রাচী নেচে দেখালো।

তাহলে ব্যাপার এখানে দুটো:

- প্রাচী নাচতে পারে অর্থাৎ ওর নাচার ক্যাপাবিলিটি আছে
- কিন্তু প্রজ্ঞা নাচতে না বললে প্রাচী নাচে না

ঠিক তেমনি তোমার প্রোগ্রাম এর কোন কিছু করার ক্ষমতা আছে অর্থাৎ তার একটি মেথড আছে। কিন্তু ওই মেথডটি ততক্ষণ পর্যন্ত কোন কিছুই করবে না যতক্ষণ না তাকে অন্য কোথাও থেকে কল(Call) করা হয়।

যেমন তোমার একটি মেথড আছে যার নাম prachiDance এবং এই মেথডটি স্ক্রীণে Prachi is dancing! লেখাটি দেখাতে পারে।

কিন্তু এই মেথডটি Prachi is dancing! লেখাটি স্ক্রীণে দেখাবে না যতক্ষণ না তুমি এটাকে অন্য কোথাও থেকে কল করো।

মেথডের রিটার্ন টাইপ(Return Type)

আমরা বলেছিলাম যে প্রাচী নুডলস রান্না করতে পারে আবার টাকাও গুনতে পারে। এখন প্রজ্ঞা বললো, প্রাচী এক বাটি নুডলস রান্না করে নিয়ে আসো তো।

এখানে কিন্তু প্রজ্ঞা দুই ভাবে কথাটা বলতে পারে:

- প্রাচী নুডলস রান্না করো তো। অথবা
- প্রাচী নুডলস রান্না করে নিয়ে আসো তো।

দুটো কথার মধ্যে কিন্তু পার্থক্য আছে। প্রজ্ঞা যদি বলে প্রাচী নুডলস রান্না করে নিয়ে আসো তো। তাহলে প্রাচী এক বাটি নুডলস রান্না করে নিয়ে এসে প্রজ্ঞাকে দেবে। আর যদি শুধু বলে প্রাচী নুডলস রান্না করো তো। তাহলে সে

কিন্তু শুধু নুডলস রান্না করে বসে থাকতে পারে। কারণ তাকে বলা হয় নি যে সেটা আবার প্রজ্ঞাকে কে এনে দিতে হবে।

মেথডের রিটার্ন টাইপ(Return Type) বিষয়টা এরকমই। তুমি একটি মেথড বানাতে যে 12 আর 19 যোগ করার জন্য। সেই মেথডে যদি তুমি Return করতে না বলো তাহলে মেথডটি শুধু 12 আর 19 যোগ করে রেখে দেবে। কারণ তুমি তাকে বলো নি যে যোগফল তোমার কাছে আবার ফেরত পাঠাতে।

মেথডের প্যারামিটার(Parameter)

আমরা একটা মেথড বানিয়েছিলাম মনে আছে? যেটা 12 এবং 19 যোগ করে দেয়। ভালো কথা 12 এবং 19 যোগ করা যেহেতু সহজ জিনিস সেহেতু আমরা কঠিন একটা যোগ প্রোগ্রামকে দিয়ে করিয়ে নিই তাহলে। হতে পারে সেটা 7298347 এবং 8222234 এই দুটো সংখ্যাকে যোগ করে দেবে। তাহলে চলো দেখি।

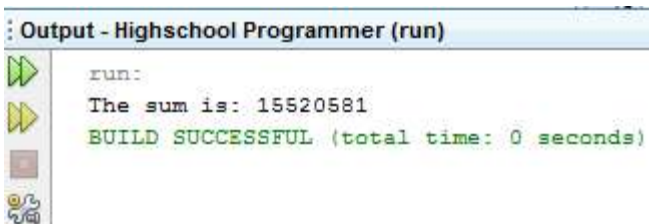
মেথড:

```
public static int sumOfTwoNumbers() {  
    int firstNumber = 7298347;  
    int secondNumber = 8222234;  
    int sum = firstNumber + secondNumber;  
    return sum;  
}
```

কল:

```
public static void main(String[] args) {  
    System.out.println("The sum is: " + sumOfTwoNumbers());  
}
```

রেজাল্ট:



```
Output - Highschool Programmer (run)  
run:  
The sum is: 15520581  
BUILD SUCCESSFUL (total time: 0 seconds)
```

এখন দেখো তোমার যদি আরো দুটো সংখ্যা যোগ করতে হয় যেমন 43482798444 এবং 988934872232 তাহলে আরো একটা মেথড তৈরী করতে হবে। এরকম আরো দুটো সংখ্যার ক্ষেত্রে আরো একটা মেথড। ব্যাপারটা কঠিন হয়ে যাচ্ছে তাই না?

এর একটা সহজ সমাধান আছে। মেথডে প্যারামিটার(Parameter) ব্যবহার করা। প্যারামিটার হলো এক বা একাধিক ভেরিয়েবল বা এ্যারে(Array) যা কোন মেথড তার নামের পরে () এই Round Bracket এর মধ্যে গ্রহন করে থাকে। কোন মেথডে প্যারামিটার ব্যবহার করলে সেটার যোগফল বের করার ক্ষমতা হয়ে যাবে আনলিমিটেড। যখন তুমি মেথডটি কল করবে তখন যে দুটো সংখ্যার যোগফল বের করতে চাও সে দুটো সংখ্যাকে প্যারামিটার হিসেবে শুধু পাঠিয়ে দিলেই হলো।

লুপিং (Looping)

ধরো প্রাচী প্রজ্ঞাকে বললো I love my sister Prachi. এই কথাটা খাতায় 5 বার লিখো। তখন ও খাতা কলম বের করে 10 বার I love my sister Prachi. কথাটা লিখলো। অর্থাৎ প্রজ্ঞা একই কাজ কিন্তু 5 বার করলো।

এরকম যখন প্রোগ্রামকে দিয়ে কোন একই কাজ বারবার করানোর প্রয়োজন পরে তখন প্রোগ্রামে Looping ব্যবহার করা হয়।

প্রোগ্রামে কয়েক ধরনের Loop ব্যবহার করা যায়। যেমন: For Loop, While Loop, Do While Loop, Foreach Loop etc.

এসো আগে আমরা For Loop টা দেখি।

For Loop

উপরের উদাহরণটায় দেখো। প্রজ্ঞা যখন খাতায় I love my sister Prachi. কথাটা 5 বার লিখলো তখন ও কিভাবে কাজটা করছে একটু ভেঙ্গে দেখি। দেখো তো ব্যাপারটা এরকম না?

১. প্রজ্ঞা প্রথমে 1 থেকে গুনতে শুরু করবে।

২. ওকে লিখতে বলা হয়েছে 5 বার সুতরাং ও চেক করে দেখবে 1 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোট।

৩. তাহলে ও খাতায় I love my sister Prachi. কথাটা লিখবে।

৪. এর পর ও গণনার সংখ্যাটাকে 1 বাড়িয়ে দেবে। অর্থাৎ আগে ছিল সেটা 1 এখন হয়ে

৫. এবার আবার ও চেক করে দেখবে 2 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোট। সুতরাং আবার ও I love my sister Prachi. কথাটা লিখবে।

৬. এর পর ও গণনার সংখ্যাটাকে 1 বাড়িয়ে দেবে। অর্থাৎ আগে ছিল সেটা 2 এখন হয়ে যাবে 3।

৭. এবার আবার ও চেক করে দেখবে 3 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোট। সুতরাং আবার লাইনটা লিখবে।

৮. লেখার পরিমাণ 1 বাড়াবে। সেটা হয়ে যাবে এবার 4।

৯. আবার চেক করে দেখবে 4 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোট।

১০. আবার লাইনটা লিখবে।

১১. সংখ্যাটাকে 1 বাড়িয়ে দেবে। অর্থাৎ আগে ছিল সেটা 4 এখন হয়ে যাবে 5।

১২. আবার চেক করে দেখবে 5 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোট নয় কিন্তু সমান।

১০. আবার লাইনটা লিখবে।

১১. সংখ্যাটাকে 1 বাড়িয়ে দেবে। অর্থাৎ আগে ছিল সেটা 5 এখন হয়ে যাবে 6।

১৩. আবার চেক করে দেখবে 6 কি 5 এর চেয়ে ছোট অথবা সমান কি না। দেখবে ছোটও নয়, সমানও নয় বরং বড়।

১৪. ও তখন লেখা বন্ধ করে দেবে।

ঠিক এইভাবেই ও যতক্ষণ না 5 বার I love my sister Prachi. লেখাটি লেখা হচ্ছে ততক্ষণ চালিয়ে যাবে। আর যখনই দেখবে 5 বার লেখা হয়ে গেছে ও লেখা থামিয়ে দেবে।

কিন্তু এখানে খেয়াল করে দেখো ও কিন্তু যতো যা-ই করুক না কেন ২,৩,৪ এই তিনটি ধাপের কাজই বারবার করছে।

তার অর্থ হলো:

১. ওকে চেক করে দেখতে হচ্ছে যতবার লেখা হয়েছে সেটার পরিমাণ যতবার লিখতে বলা হয়েছে তার চেয়ে কম বা সমান কি না।

২. যেটা লিখতে বলা হয়েছে সেটা লেখা।

৩. লেখার পরে যতবার লেখা হয়েছে তার পরিমাণ এক বাড়িয়ে দেয়া।

আর দুটো কাজ এই 3 টি স্টেপের বাইরে করতে হচ্ছে। সেটা হলো:

১. 1 থেকে গণনা শুরু করা।

২. যতবার লিখতে বলা হয়েছে গণনার সংখ্যাটা তার চেয়ে বেশি বার হয়ে গেলেই লেখা থামিয়ে দেয়া।

এখন মজার ব্যাপারটা হলো প্রোগ্রামে For Loop ঠিক একইভাবে কাজ করে। চলো আমরা একটা প্রোগ্রাম লিখে দেখি।

ডিসিশন মেকিং(Decision Making)

এবার ধরে প্রজ্ঞা আর প্রাচী কলেজে গেছে। হঠাৎ প্রজ্ঞার শরীর খারাপ লাগা শুরু করলো, বাসায় চলে যাবে। প্রাচী তখন প্রজ্ঞাকে বললো ঠিক আছে বাসায় যাবে যখন যাও, কিন্তু আজকে তো আকাশ মেঘলা, যে কোন সময় বৃষ্টি আসতে পারে।

বাসায় গিয়ে রুমের জানালা যদি খোলা দেখে তাহলে জানালাটা বন্ধ করে দিও।

এখন প্রজ্ঞা বাসায় এলো।

জানালাটা চেক করে দেখলো খোলা আছে।

জানালাটা বন্ধ করে দিলো।

এখানে প্রজ্ঞা যেটা করলো সেটা আসলে একটা ডিসিশন মেকিং। ও জানালাটা চেক করে দেখলো, দেখলো খোলা আছে, জানালাটা বন্ধ করে দিলো।

এই যে একটা চেকিং করা এবং তার ভিত্তিতে কোন একটা কাজ করা এটাই বাস্তব জীবনে ডিসিশন মেকিং। বাস্তব জীবনে আমরা এরকম সব সময়ই করি।

প্রোগ্রামেও এই কাজ একই ভাবে করা যায়। প্রোগ্রামকেও বলে দেয়া যায় যে তুমি এই কন্ডিশন চেক করে দেখবে, যদি চেকিং এর রেজাল্ট true হয় তাহলে এই কাজটি করবে।

OOP: Object Oriented Programming (অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং)

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং(Object Oriented Programming) বা OOP খুবই গুরুত্বপূর্ণ একটা বিষয় প্রোগ্রামিং এর ক্ষেত্রে।

আমরা OOP বুঝবো, তবে তার আগে চলো ছোট্ট একটা গল্প শুন। গল্পটা হয়তো তোমরা আগে থেকেই শুনে থাকতে পারো অনেকে।

এক রাজার পর্ষদ বর্গের মাঝে মন্ত্রী এবং পন্ডিত উভয়ই আছেন। মন্ত্রী সমস্ত রাজ্যের দেখভাল করেন। নিরাপত্তা, শান্তি-শৃঙ্খলা, নিয়ম কানুন অনুযায়ী পরিচালনা করা সবই উনিই দেখাশুনা করতেন।

কিন্তু রাজা পন্ডিতকে বেতন দেন মন্ত্রীর চাইতেই বেশি। আর এ নিয়ে মন্ত্রী খুবই ক্ষিপ্ত। সে একদিন রাজার কাছে গিয়ে নালিশ করে বসলো। রাজামশাই আমি এতো এতো দায়িত্ব পালন করি অথচ আপনি পন্ডিতকে বেতন দেন আমার চাইতেও বেশি, যে কিনা সারা দিন বসে বসে শুধু বই পড়ে।

তখন রাজা বললেন ঠিক আছে, কেন আমি পন্ডিতকে তোমার চেয়ে বেশি বেতন দিই তা প্রমাণ করে দিচ্ছি। রাজ বাড়ির সদর দরজার সামনে পোষা কুকুরটা বাচ্চা দিয়েছে, দেখে এসো তো।

মন্ত্রী দেখে এলো।

রাজা জিজ্ঞেস করলেন, দেখে এসেছ?

মন্ত্রী বললো হ্যাঁ।

কয়টা বাচ্চা দিয়েছে কুকুরটা?

তা তো গুণে দেখি নাই হুজুর। দাঁড়ান আমি গুণে আসি। গুণে এসে বললো, পাঁচটা।

আচ্ছা। কয়টা বাচ্চা সাদা, কয়টা বাচ্চা কালো?

আবার মন্ত্রী দৌড় মারলো এবং এসে বললো দুইটা কালো, তিনটা সাদা।

ছেলে কয়টা মেয়ে কয়টা?

আবার মন্ত্রী দৌড়লো এবং ঘুরে এসে বললো, তিনটা ছেলে দুইটা মেয়ে।

এবার রাজা মন্ত্রীর সামনেই পন্ডিতকে ডাকলেন এবং একই কথা বললেন, রাজ বাড়ির সদর দরজার সামনে পোষা কুকুরটা বাচ্চা দিয়েছে, দেখে এসো তো।

পন্ডিত ঘুরে এলে রাজা জিজ্ঞেস করলেন, কয়টা বাচ্চা দিয়েছে কুকুরটা?

পন্ডিত বললো, হুজুর পাঁচটা।

কয়টা বাচ্চা সাদা, কয়টা বাচ্চা কালো?

হুজুর দুইটা কালো, তিনটা সাদা।

ছেলে কয়টা মেয়ে কয়টা?

হুজুর তিনটা ছেলে দুইটা মেয়ে।

মন্ত্রীর কি অবস্থা বুঝতেই পারছো।

এখানে দেখো মন্ত্রী এবং পন্ডিত দুই জনই কিন্তু একই কাজ করেছে। কিন্তু তাদের কাজের ধরন এবং দক্ষতার মধ্যে অনেক তফাৎ।

প্রোগ্রামিং করার জন্য প্রসিডিউরাল প্রোগ্রামিং(Procedural Programming) এবং অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং(Object Oriented Programming) আসলে দুটো আলাদা পদ্ধতি এবং এই দুই পদ্ধতির মধ্যে পার্থক্য মূলত এই ধরনেরই।

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং পদ্ধতি মূলত তোমাকে প্রোগ্রামে অনেকগুলো ফ্যাসিলিটি বা সুবিধা যোগ করবে যেগুলো প্রসিডিউরাল পদ্ধতি দিতে পারবে না।

এখন আবার নতুন প্রশ্ন, তাহলে প্রসিডিউরাল প্রোগ্রামিং কি জিনিস।

নিচের ছবিটা দেখো:

Procedural	Object Oriented
<pre>int variable1; String variable2; float variable3; char varibale4; boolean variable5; public static void method1(){ //statement } public static void method2(){ //statement } public static void method3(){ //statement } public static void method4(){ //statement }</pre>	<pre>public class AmraSobaiProgrammer { int variable1; String variable2; float variable3; char varibale4; boolean variable5; public static void method1 () { //statement } public static void method2 () { //statement } public static void method3 () { //statement } public static void method4 () { //statement } }</pre>

দেখো বাম পাশের কোডেও কিছু ভেরিয়েবল এবং কিছু মেথড আছে। ডান পাশেও আছে। কিন্তু ডান পাশের সমস্ত ভেরিয়েবল এবং মেথড একটা ক্লাশের মধ্যে আছে।

প্রসিডিউরাল এবং অবজেক্ট ওরিয়েন্টেড এই দুই পদ্ধতির মধ্যে সবচে' বেসিক এবং ভিজুয়াল পার্থক্য হলো প্রসিডিউরাল পদ্ধতিতে কোডগুলো ক্লাশের মধ্যে থাকে না। আর অবজেক্ট ওরিয়েন্টেড এ সমস্ত কোড একটা ক্লাশের মধ্যে থাকে।

আমরা তো রাজা-গজার একটা গল্প শুনলাম। চলো এবার একটু বর্তমান যুগের একটা গল্প থেকে বুঝার চেষ্টা করি।

ধরো তুমি একটি কার(Car) কোম্পানির মালিক। নিজেকে খুব ধনী মনে হচ্ছে তাই না? আচ্ছা ধরে নাও তুমি অনেক ধনী এবং একটি কার কোম্পানি আছে তোমার।

তোমার কার কোম্পানিতে এক জন ডিজাইনার রেখে দিয়েছ যে প্রতিনিয়ত নতুন নতুন কারের ডিজাইন করে। কিন্তু ব্যাপারটা হলো সে যখন নতুন একটি কার ডিজাইন করবে তখন কি সেই কারটার সমস্ত কিছু নতুন করে ডিজাইন করবে?

না।

কেন নয় কারণ নতুন যে ডিজাইনের কারই ডিজাইন করুক না কেন সেটার তো বেসিক কিছু ফিচার এবং ফাংশনালিটি(কিছু করার ক্ষমতা) থাকতেই হবে।

যেমন ফিচার: কারটির চাকা থাকবে, ড্যাসবোর্ড থাকবে, ষ্টিয়ারিং থাকবে, সিট থাকবে, মিরর থাকবে, বনেট থাকবে ইত্যাদি।

যেমন ফাংশনালিটি: কারটি চলতে পারবে, ডানে বামে ঘুরতে পারবে, ব্রেক করতে পারবে, হুইসেল দিতে পারবে, ব্রেক করলে পেছনে লাইট জ্বালাতে পারবে ইত্যাদি।

যখন তোমার ডিজাইনার নতুন একটি কার ডিজাইন করবে তখন তো এসব বেসিক ব্যাপারগুলো নতুন করে ডিজাইন করার দরকার নেই।

সে যেটা করতে পারে:

১. ওই বেসিক ফিচারগুলো নিয়েই একটা কার বানিয়ে ফেলা যেতে পারে।

২. নতুন আরো ফিচার বা ফাংশনালিটি(কিছু করার ক্ষমতা) যোগ করতে পারে। যেমন: কার এ সিডি প্লেয়ার থাকবে, ব্যাক ক্যামেরা থাকবে, এয়ার ব্যাগ থাকবে, রেডিও থাকবে, এসি থাকবে, জাম্প করতে পারবে ইত্যাদি।

৩. কমন ফিচারগুলোতে নতুন ফ্যাসিলিটি যোগ করতে পারে। যেমন: চাকা আরো শক্তিশালী করলো(চাকাতে নতুন ফ্যাসিলিটি যোগ করলো), সিটের ডিজাইন আরো সুন্দর করলো(চাকাতে নতুন ফ্যাসিলিটি যোগ করলো), অনাকাঙ্ক্ষিতভাবে কোন কিছু কারের সামনে এলে কারটি হার্ড ব্রেক করবে(ব্রেক এ নতুন ফ্যাসিলিটি যোগ করলো), কারের সর্বোচ্চ গতি বাড়িয়ে দিলো(কারের চলতে পারার ক্ষমতাতে নতুন ফ্যাসিলিটি যোগ করলো), হুইসেলের সাউন্ডটা আরো নমনীয় করলো যাতে কানে সহনীয় হয়(হুইসেল দেবার ক্ষমতাতে নতুন ফ্যাসিলিটি যোগ করলো)

তাহলে কাজ করার পদ্ধতিটা অনেক এফিসিয়েন্ট হলো তাই না? নতুন করে রি-ওয়ার্ক করতে হলো না মানে সব কিছু নতুন করে ডিজাইন করতে হলো না, অথচ তার যা উদ্দেশ্য ছিল সবই সফল হলো।

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং এ কাজের পদ্ধতি এরকমই।

১. অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং পদ্ধতি কোড বা প্রোগ্রামকে এমনভাবে করার সুবিধা দেয় যাতে রি-ওয়ার্ক করতে হয় না। আগে কোন প্রোগ্রাম বানিয়েছ তা তোমাকে আবার বানাতে হবে না। সেটা পুনরায় ব্যবহার করা যায়।

২. আগের প্রোগ্রামে নতুন কিছু যোগ করতে হলে খুব সহজেই যোগ করা যায় বা আগের প্রোগ্রামের কোন কিছুতে সহজেই নতুন ফিচার যোগ করা যায়।

৩. কোডগুলো খুব অর্গানাইজড হয়। যখন তুমি অনেক বড় বড় প্রোগ্রাম লিখবে তখন কোডগুলো ম্যানেজ করা অনেক সহজ হবে।

৪. কোডের নিরাপত্তা প্রদান করে।

এছাড়া আরো অনেক সুবিধা দেয় যেগুলো আমরা উদাহরণ সহকারে দেখতে থাকলেই বুঝে ফেলবো। তেমন কঠিন কিছু না।

অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং(Object Oriented Programming) কি?

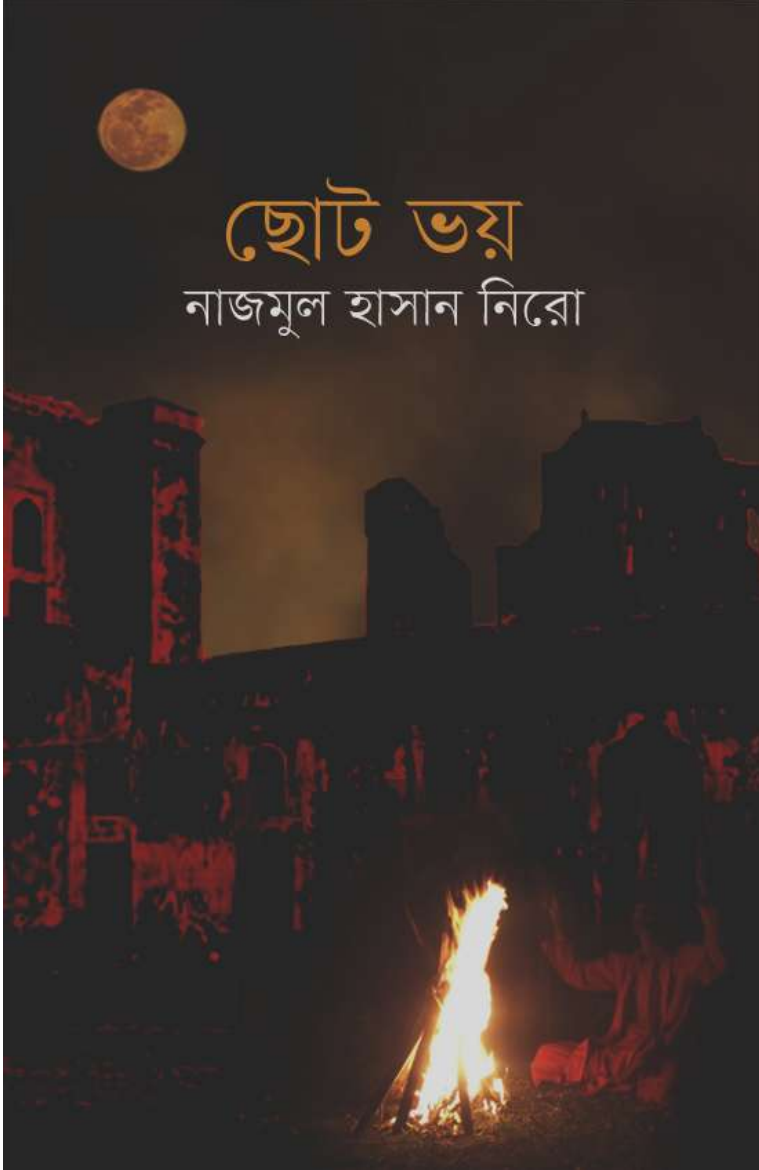
আশা করি ইতোমধ্যে তোমরা মোটামুটি একটা ওভারভিউ পেয়েছ অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং এর ব্যাপারে। যেগুলো আমরা উপরে আলোচনা করলাম। এই পদ্ধতিটি আসলে রিয়েল লাইফের সাথে সামঞ্জস্যপূর্ণ। অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং এমন একটি পদ্ধতি যা প্রোগ্রামিং করার সময় নিচের ফ্যাসিলিটিগুলো প্রদান করে:

1. Inheritance
2. Polymorphism
3. Encapsulation
4. Abstraction

চলো আমরা হাতে কলমে দেখি। উপরের ফ্যাসিলিটি গুলো পেতে হলে আমাদের বেশ কিছু কনসেপ্টের সাথে পরিচিত হতে হবে। অনেক সহজ ব্যাপার আমরা একটা একটা করে দেখলেই বুঝে ফেলবো।

নাজমুল হাসান নিরো এর অন্যান্য বই

ছোট ছোট কিছু ভয়, থ্রিলার, সাইকো থ্রিলার এবং হরতন দা সিরিজের দুটি গোয়েন্দা
গল্পের সংকলন



প্রকাশক: কুঁড়েঘর প্রকাশনী

সার্বিক যোগাযোগ(লেখকের প্রতিনিধি): 01751-565743, 01746-053233