**Object Oriented Programming in PHP**

**1. অবজেক্ট-ওরিয়েন্টেড প্রোগ্রামিং (OOP) কি?**

OOP হলো একটি প্রোগ্রামিং প্যারাডাইম, যেখানে অবজেক্টগুলি, যা ক্লাসের ইনস্ট্যান্স, সফ্টওয়্যার মডেল এবং স্ট্রাকচার করতে এবং স্ট্রাকচার করতে ব্যবহৃত হয়।

**2. PHP-তে ক্লাস কি?**

PHP-তে একটি ক্লাস হলো অবজেক্ট তৈরির জন্য একটি ব্লুপ্রিন্ট। এটি অবজেক্টের বৈশিষ্ট্য (আট্রিবিউট) এবং মেথড (ফাংশন) সংজ্ঞান করে, যা ক্লাসের অবজেক্টগুলি থাকবে।

**3. PHP-তে ক্লাস কীভাবে সংজ্ঞান করা হয়?**

আপনি PHP-তে একটি ক্লাস সংজ্ঞান করতে class কীওয়ার্ডটি ব্যবহার করতে পারেন, যা ক্লাসের নাম এবং একটি ব্রেসের জোড়া দ্বারা অনুসরণ করে।

**4. PHP-তে অবজেক্ট কি?**

PHP-তে অবজেক্ট হলো একটি ক্লাসের ইনস্ট্যান্স। এটি ক্লাসের নাম দিয়ে তৈরি হয়, যা নতুন করে তৈরি হওয়ার জন্য new কীওয়ার্ড সাথে।

**5. OOP-তে ক্যাপসুলেশন কি?**

ক্যাপসুলেশন হলো ডেটা (আট্রিবিউট) এবং ডেটা উপর কাজ করতে মেথড (ফাংশন) গুলির একটি একক ইউনিট ক্লাস নামে সংক্রান্ত করা এবং সফ্টওয়্যার ডিজাইন করা।

**6. PHP-তে ইনহেরিট্যান্স কি?**

ইনহেরিট্যান্স হলো PHP-তে একটি নতুন ক্লাস (সাবক্লাস বা ডেরাইভড ক্লাস) যা একটি বীজ ক্লাস (সুপারক্লাস বা বেস ক্লাস) থেকে প্রপার্টিস এবং মেথডগুলি ইনহেরিট করতে পারে।

**7. PHP-তে ইনহেরিট্যান্স কীভাবে সংজ্ঞান করা হয়?**

PHP-তে ইনহেরিট্যান্স সংজ্ঞান করতে ক্লাস সংজ্ঞান করার সময় extends কীওয়ার্ড ব্যবহার করা যেতে পারে।

**8. PHP-তে কনস্ট্রাক্টর কি?**

একটি কনস্ট্রাক্টর হলো ক্লাসের একটি বিশেষ মেথড যা অটোমেটিক্যালি অবজেক্ট তৈরি করার সময় স্বয়ংক্রিয়ভাবে কল করা হয়। এটি অবজেক্টের বৈশিষ্ট্য আপনাদের সাথে পরিচিত করানোর জন্য ব্যবহৃত হয়।

**9. PHP-তে $this কীওয়ার্ড কি?**

$this কীওয়ার্ড হলো ক্লাসের মেথডের ভিতরে বর্তমান ইনস্ট্যান্সে সেই ক্লাসের প্রতিনিধিত্ব করতে ব্যবহৃত হয়।

**10. PHP-তে মেথড ওভারলোডিং কি?**

PHP সাধারণভাবে মেথড ওভারলোডিং সাপোর্ট করে না, অর্থাৎ একই নামে কিছু ভিন্ন প্যারামিটার সহ একের বেশী মেথড সংজ্ঞান করা সম্ভব নয়।

**11. PHP-তে মেথড ওভাররাইডিং কি?**

মেথড ওভাররাইডিং হলো সাবক্লাসে সুপারক্লাসের মেথডের সাথে একই নাম এবং সিগনেচার দিয়ে একটি মেথড সংজ্ঞান করা, যা সাবক্লাসে সুপারক্লাসের মেথড প্রতিস্থাপন করে।

**12. PHP-তে একটি অ্যাবস্ট্রাক্ট ক্লাস কি?**

একটি অ্যাবস্ট্রাক্ট ক্লাস হলো একটি ক্লাস যা নিজে নিজে তৈরি হতে পারে না। এটি সাবক্লাসের জন্য তৈরি হয় এবং এর সাথে অবস্ট্রাক্ট মেথড থাকতে পারে, যা সাবক্লাসের দ্বারা অবশ্যই সংজ্ঞান করতে হবে।

**13. PHP-তে একটি অ্যাবস্ট্রাক্ট ক্লাস কীভাবে সংজ্ঞান করা হয়?**

আপনি একটি অ্যাবস্ট্রাক্ট ক্লাস সংজ্ঞান করতে অ্যাবস্ট্রাক্ট কীওয়ার্ডটি সংজ্ঞান করে এবং এর সাথে একটি ক্লাসের নাম এবং একটি কোড ব্লক ব্যবহার করতে পারেন।

**14. PHP-তে ইন্টারফেস কি?**

ইন্টারফেস হলো একটি ক্লাসের জন্য একটি কন্ট্রাক্ট সংজ্ঞান করার জন্য। এটি কোনও ক্লাস যেটি এই ইন্টারফেস গুলি গ্রহণ করে, সেই ক্লাসের দ্বারা সংজ্ঞান করা মেথডগুলি স্বপর্যাপ্তভাবে ইমপ্লিমেন্ট করতে হবে।

**15. PHP-তে একটি ইন্টারফেস কীভাবে সংজ্ঞান করা হয়?**

আপনি PHP-তে একটি ইন্টারফেস সংজ্ঞান করতে interface কীওয়ার্ডটি ব্যবহার করতে পারেন, এবং সেই ইন্টারফেসের নাম এবং মেথডের সিগনেচার সম্পর্কে একটি তালিকা দিতে পারেন।

**16. PHP-তে একটি ক্লাস একাধিক ইন্টারফেস ইমপ্লিমেন্ট করতে পারে কি?**

হ্যাঁ, PHP-তে একটি ক্লাস একাধিক ইন্টারফেস ইমপ্লিমেন্ট করতে পারে, ইমপ্লিমেন্ট করতে হলে implements কীওয়ার্ডের সাথে তাদের কমা দিতে হবে।

**17. PHP-তে ট্রেইট কি?**

ট্রেট হলো একটি পুনর্ব্যবহারযোগ্য কোড পিস, যা একাধিক ক্লাসে ব্যবহার করা যেতে পারে। এটি আপনাকে ইনহেরিটেন্স ছাড়াই বিভিন্ন ক্লাসে মেথড পুনর্ব্যবহার করতে দেয়।

**18. PHP-তে ট্রেট কীভাবে সংজ্ঞান করা হয়?**

আপনি PHP-তে একটি ট্রেট সংজ্ঞান করতে trait কীওয়ার্ডটি ব্যবহার করতে পারেন, যা ট্রেটের নাম এবং একটি কোড ব্লক সহে।

**19. PHP-তে ট্রেট ক্লাসে কীভাবে ব্যবহার করা হয়?**

ট্রেট ক্লাসে ব্যবহার করতে হলে ক্লাস ডিফিনিশনের মধ্যে use কীওয়ার্ড ব্যবহার করে ট্রেটের নাম দিতে হবে।

**20. PHP-তে final কীওয়ার্ডের উপযোগ কি?**

final কীওয়ার্ডটি ব্যবহার করে কোনও ক্লাসকে সাবক্লাস হওয়া বন্ধ করা অথবা কোনও মেথড ওভাররাইড হওয়া বন্ধ করা যায়।

**21. PHP-তে static কীওয়ার্ডের উপযোগ কি?**

static কীওয়ার্ডটি ব্যবহার করে ক্লাস-লেভেলের প্রপার্টি এবং মেথড সংজ্ঞান করতে ব্যবহার করা হয়, যা ক্লাসের ইনস্ট্যান্সের সাথে সম্পর্কিত নয়, বরং ক্লাস নিজেই।

**22. PHP-তে দৃশ্যমানতা (অ্যাক্সেস মডিফায়ার) কি?**

PHP-তে তিনটি দৃশ্যমানতা মডিফায়ার সাপোর্ট করে: পাবলিক, প্রোটেক্টেড, এবং প্রাইভেট, যা ক্লাসের প্রপার্টি এবং মেথডের দখলে এবং বাইরে কি কি দেখাবে সেটা নিয়ন্ত্রণ করে।

**23. PHP-তে পাবলিক, প্রোটেক্টেড, এবং প্রাইভেট দৃশ্যমানতার মধ্যে পার্থক্য কি?**

পাবলিক মেম্বার সর্বদা সহযোগী হয়।

প্রোটেক্টেড মেম্বার শুধুমাত্র ক্লাস এবং এর সাবক্লাসের মধ্যে সহযোগী হয়।

প্রাইভেট মেম্বার কেবল ক্লাসের ভিতরে সহযোগী হয়।

**24. PHP-তে নামস্পেস কি?**

নামস্পেস হলো ক্লাস, ফাংশন, এবং কনস্টেন্টগুলি সংক্রান্ত নামের দ্বন্দ্ব সমস্যা এবং একটি নির্দিষ্ট নামস্পেসে গ্রুপ করার উপায়।

**25. PHP-তে নামস্পেস কীভাবে সংজ্ঞান করা হয়?**

আপনি একটি নামস্পেস সংজ্ঞান করতে হলে একটি PHP ফাইলের শুরুতে namespace কীওয়ার্ড ব্যবহার করতে পারেন।

**26. PHP-তে অন্য নামস্পেস থেকে উপাদান কীভাবে অ্যাক্সেস করা হয়?**

আপনি অন্য নামস্পেস থেকে উপাদান অ্যাক্সেস করতে পারেন ফুলি কোয়ালিফাইড নাম বা use কীওয়ার্ড ব্যবহার করে নামস্পেস ইমপোর্ট করে।

**27. PHP-তে আটোলোডিং কি?**

আটোলোডিং হলো ক্লাস ফাইলগুলি স্বয়ংক্রিয়ভাবে সংযোজন করার প্রক্রিয়া, ব্যক্তিগতভাবে require বা include স্টেটমেন্ট ব্যবহার না করে।

**28. PHP-তে আটোলোডিং কীভাবে প্রযোজ্য করা হয়?**

আপনি আটোলোডিং প্রয়োগ করতে spl\_autoload\_register() ব্যবহার করে একটি কাস্টম আটোলোডার ফাংশন সংজ্ঞান করতে পারেন।

**29. ওপরে উল্লিখিত সবচেয়ে কাছাকাছি কিভাবে ব্যবহার করা হয় জনপ্রিয় ডিজাইন প্যাটার্ন গুলি?**

এই ডিজাইন প্যাটার্ন গুলি ব্যবহার করতে ক্লাস এবং অবজেক্ট স্ট্রাকচার ডিজাইনে সাহায্য করে এবং সফটওয়্যার প্রক্রিয়াকরণ ব্যবস্থাপনা করে।

**30. ফ্যাক্টরি ডিজাইন প্যাটার্ন কি?**

ফ্যাক্টরি প্যাটার্ন হলো একটি ক্রিয়েশনাল ডিজাইন প্যাটার্ন, যা অবজেক্টগুলির সৃষ্টি এবং মোডিফিকেশন নির্দিষ্টভাবে পরিচালনা করতে সাহায্য করে।

**31. অবজারভার ডিজাইন প্যাটার্ন কি?**

অবজারভার প্যাটার্ন হলো একটি ব্যবহারকারী অবজেক্ট যা একটি সাবজেক্টের অবদানের পরিবর্তনে অন্য অবদানকারী অবজেক্টগুলিকে সংজ্ঞান করে এবং তাদেরকে পরিবর্তিত করে।

**32. স্ট্র্যাটেজি ডিজাইন প্যাটার্ন কি?**

স্ট্র্যাটেজি প্যাটার্ন হলো একটি ব্যবহারকারী সংজ্ঞান করে একটি কাজ সংজ্ঞান করে এবং তার লজিক এবং প্রক্রিয়া পরিবর্তন করার জন্য বিভিন্ন স্ট্র্যাটেজি সম্পর্কে নির্ধারণ করে।

**33. মডেল-ভিউ-কন্ট্রোলার (MVC) ডিজাইন প্যাটার্ন কি?**

MVC ডিজাইন প্যাটার্ন হলো সফটওয়্যার প্রক্রিয়াকরণ এবং উপাদানের নিয়ন্ত্রণ দুটি অংশে বিভাজনের জন্য ব্যবহৃত ডিজাইন প্যাটার্ন। এটি ব্যবহারকারী ইন্টারফেস এবং লজিকের প্রতিটি অংশকে আলাদা করে পরিচালনা করে।

**34. কোম্পোজাইশন ডিজাইন প্যাটার্ন কি?**

কোম্পোজিশন প্যাটার্ন হলো একটি ডিজাইন প্যাটার্ন যা অবজেক্টটিকে অন্য অবজেক্টের অংশ হিসেবে সংমিলিত করতে সাহায্য করে, যা সফটওয়্যার ব্যবহারকারী এবং প্রক্রিয়াকরণে সুস্থিত এবং পরিচালনা করতে সাহায্য করে।

**35. সিঙ্গেলটন প্যাটার্ন কীভাবে PHP-তে ব্যবহার করা হয়?**

সিঙ্গেলটন প্যাটার্ন প্রোটিপ ক্লাস এবং একটি স্ট্যাটিক মেথড ব্যবহার করে একটি ক্লাসের একমাত্র ইনস্ট্যান্স সারিয়ে নেওয়া সুন্দর ডিজাইন প্যাটার্ন। ক্লাস ডিফিনিশনে একটি প্রাইভেট স্ট্যাটিক মেথড সংজ্ঞান করে সিঙ্গেলটন ক্লাসের ইনস্ট্যান্স তৈরি এবং ইনস্ট্যান্সটি স্ট্যাটিক প্রপার্টি দ্বারা সংরক্ষণ করতে পারেন।

**36. SOLID ডিজাইন প্রিন্সিপল (SOLID) কী?**

SOLID একটি ডিজাইন প্রিন্সিপলের আক্রমণ যা প্রোগ্রামিং মডেলের সূত্র সমূহ বর্ণনা করে:

একক দায়িত্বের সূত্র (Single Responsibility Principle)

উন্মুক্ত/বন্ধ সূত্র (Open/Closed Principle)

লিস্কভ সাবস্টিটিউশন সূত্র (Liskov Substitution Principle)

ইন্টারফেস সেগ্রিগেশন সূত্র (Interface Segregation Principle)

ডিপেন্ডেন্সি ইনভারশন সূত্র (Dependency Inversion Principle)

**37. একক দায়িত্ব সূত্র (Single Responsibility Principle, SRP) কি?**

SRP বলে যে একটি ক্লাস আপনাকে কেবল একটি কারণ দিতে হবে বা কেবল একটি দায়িত্ব পালন করতে হবে তা মানে। অপ্রয়োজনীয় কোড অথবা যে কোনও অতিরিক্ত কাজ করা উচিত নয়।

**38. উন্মুক্ত/বন্ধ সূত্র (Open/Closed Principle, OCP) কি?**

OCP একটি স্বাগতমূলক সূত্র যা সফটওয়্যার ইন্টিটিগুলির সংশোধন না করে পরিবর্তন যোগ করতে সাহায্য করে। প্রয়োজনে নতুন কোড যোগ করে পূর্বের কোড পরিবর্তন না করে।

**39. লিস্কভ সাবস্টিটিউশন সূত্র (Liskov Substitution Principle, LSP) কি?**

LSP বলে যে সাবক্লাস অবজেক্টগুলি প্যারেন্ট ক্লাস অবজেক্টগুলির উপস্থানে নিজেদের বদলে নেওয়া না পারে এবং প্যারেন্ট ক্লাস অবজেক্টের প্রতিটি কাজই করতে পারে।

**40. ইন্টারফেস সেগ্রিগেশন সূত্র (Interface Segregation Principle, ISP) কি?**

ISP একটি সূত্র যা ক্লাইন্টগুলির জন্য একটি বড় এবং সাধারণ ইন্টারফেসের সাথে নিজেদের সংজ্ঞান করে তা প্রয়োজন না করে, বরং ছোট, বিশেষগুলির ইন্টারফেস সংজ্ঞান করে।

**41. ডিপেন্ডেন্সি ইনভারশন সূত্র (Dependency Inversion Principle, DIP) কি?**

DIP সূত্র যা বলে যে উচ্চ স্তরের মডিউলগুলি নিম্ন স্তরের মডিউলগুলির উপর নির্ভর না করে, দুটি স্তরের মধ্যে একটি প্রতিনির্ভরণ স্থাপন করা উচিত। এটি ইন্টারফেসের ব্যবহার এবং ডিপেন্ডেন্সি ইনজেকশনের মাধ্যমে প্রাপ্ত করা যেতে পারে।

**42. ডিজাইন অ্যান্টি-প্যাটার্ন কি?**

ডিজাইন অ্যান্টি-প্যাটার্ন হলো সফটওয়্যার ডিজাইনের একটি সাধারণ কিন্তু অসুপরিকর ডিজাইন সমাধান, যা প্রোগ্রামিং সমস্যাগুলির সৃজনশীল এবং দক্ষ সমাধান দেওয়া চেষ্টা করে কিন্তু সাফল হয় না।

**43. কিভাবে PHP-তে একটি সিঙ্গেলটন প্যাটার্ন বাস্তবায়ন করতে হয়?**

একটি সিঙ্গেলটন ক্লাস তৈরি করতে, ক্লাসের কনস্ট্রাক্টর প্রাইভেট করে দিতে হবে, এবং সিঙ্গেলটন ক্লাসের একটি স্ট্যাটিক মেথড প্রয়োজন হবে যা সিঙ্গেলটন ক্লাসের ইনস্ট্যান্স তৈরি এবং স্ট্যাটিক প্রপার্টি দ্বারা সংরক্ষণ করতে সাহায্য করে।

**44. আপনি কোনটি অ্যান্টি-প্যাটার্ন দেখেছেন এবং কেন তা একটি ডিজাইন অ্যান্টি-প্যাটার্ন হতে পারে?**

কোনটি অ্যান্টি-প্যাটার্ন হতে পারে তা সফটওয়্যার ডিজাইনের সুপারিশেষ কাজ করে কিন্তু দর্শকদের যা আশা করা হয় তা করতে পারে না। এটি সাধারণভাবে পুরো সফটওয়্যার স্ট্রাকচার ও প্রক্রিয়াকরণ ব্যতিত দিকে মুখ করে তা দেখায়, যা অবশ্যই ভাল ডিজাইন নয়।

**45. কীভাবে সম্পূর্ণ ত্রুটিহীন PHP স্ক্রিপ্ট লেখা যায়?**

সম্পূর্ণ ত্রুটিহীন PHP স্ক্রিপ্ট লেখার জন্য নিম্নলিখিত সম্পর্কে মন্তব্য করা উচিত:

সম্পূর্ণ স্ক্রিপ্টে সঠিক সিনট্যাক্স ব্যবহার করা উচিত।

স্ক্রিপ্টে যে কোনও বাক্যপ্রদান অথবা লুপের সম্মমন সঠিক হতে হবে।

ভ্যারিয়েবল এবং ফাংশন নেমিং স্ট্যান্ডার্ড অনুসরণ করা উচিত।

ভ্যারিয়েবলগুলির মান প্রয়োজন অনুযায়ী সঠিকভাবে প্রতিস্থাপন করা উচিত।

সব ফাইলের আদি এবং শেষে একটি নির্দিষ্ট লাইন নিয়ে স্ক্রিপ্ট নিশ্চিত করা উচিত।

এরর হ্যান্ডলিং সুস্থিতভাবে ব্যবহার করা উচিত এবং যে কোনও দুর্বলতা বা ত্রুটি প্রতিরোধ করা উচিত।

মন্নিত এবং সুরক্ষিত দানকারী স্ক্রিপ্টের জন্য সুরক্ষা প্রথমে প্রতিষ্ঠাপন করা উচিত।

স্ক্রিপ্টে ভালো প্রক্রিয়াবদ্ধ এবং সুস্থিত কোড কমেন্ট করা উচিত।

কোডের লজিক এবং কার্যকারিতা ব্যবহারকারীর প্রয়োজনীয় যে কোনও ধরণের ডেটা প্রদান করা উচিত।

কোডে যে কোনও যে কোনও কমেন্ট এবং ডেবাগ লগ অপসারণ করা উচিত।

**46. কীভাবে একটি PHP ক্লাস থেকে অন্য একটি নামস্পেসের উপাদানে অ্যাক্সেস করা যায়?**

আপনি একটি অন্য একটি নামস্পেসের উপাদানে অ্যাক্সেস করতে তিনটি মেথড ব্যবহার করতে পারেন:

Fully Qualified Name: নামস্পেসের পূর্ণ নামস্পেস সরাসরি উপাদানের সাথে মিলে যায়।

Use Statement: নামস্পেস ব্যবহার স্টেটমেন্ট দ্বারা নামস্পেস ইম্পোর্ট করে তার উপাদানে অ্যাক্সেস করতে পারেন।

Aliasing: উপাদানটি আপনার নামস্পেসে আপনির পছন্দের নামে এলিয়াস করে তাকে অ্যাক্সেস করতে পারেন।

**47. কীভাবে PHP-তে আটোলোডিং ইমপ্লিমেন্ট করা যায়?**

আটোলোডিং ইমপ্লিমেন্ট করতে, আপনি spl\_autoload\_register() ফাংশন ব্যবহার করে একটি কাস্টম আটোলোডার ফাংশন সংজ্ঞান করতে পারেন। নিম্নলিখিত উদাহরণ দেখায় কীভাবে এটি কাজ করে:

// কাস্টম আটোলোডার ফাংশন ডিফাইন করা

function custom\_autoloader($class\_name) {

include $class\_name . '.php';

}

// আটোলোডার ফাংশন নিবন্ধন করা

spl\_autoload\_register('custom\_autoloader');

// ক্লাস ব্যবহার করা

$obj = new MyClass();

**48. কীভাবে PHP-তে ডিজাইন কম্পোজিশন ইমপ্লিমেন্ট করা যায়?**

ডিজাইন কম্পোজিশন ইমপ্লিমেন্ট করতে, আপনি একটি অবজেক্ট কে অন্য একটি অবজেক্টের ভাগশেষ হিসেবে ব্যবহার করতে পারেন। এটি একটি ক্লাস থেকে অন্য একটি ক্লাসে অবজেক্ট তৈরি করার সবচেয়ে সাধারণ উপায় এবং সফটওয়্যার কোড রিউসেবল করতে সাহায্য করে।

**49. ডিজাইন প্যাটার্নের কিছু সাধারণ ব্যবহৃত প্যাটার্ন কী আছে PHP-তে?**

PHP-তে কিছু সাধারণভাবে ব্যবহৃত ডিজাইন প্যাটার্ন হলো:

সিঙ্গেলটন প্যাটার্ন (Singleton Pattern): এটি একটি ক্লাসের একমাত্র ইনস্ট্যান্স তৈরি করতে ব্যবহৃত হয়।

ফ্যাক্টরি প্যাটার্ন (Factory Pattern): এটি ইনস্ট্যান্স তৈরি করার জন্য একটি ইন্টারফেস প্রদান করে এবং সাবক্লাসের সাথে নির্দিষ্ট ইন্সট্যান্স তৈরি করতে ব্যবহৃত হয়।

অবজারভার প্যাটার্ন (Observer Pattern): এটি একটি সূত্রের পরিবর্তনের পরিবর্তন স্থাপন করতে সাহায্য করে, যাতে একটি অবজেক্ট আরও বেশি অবজেক্টের পরিবর্তন স্থাপন করতে পারে।

স্ট্র্যাটেজি প্যাটার্ন (Strategy Pattern): এটি একটি কাজের প্রণালী প্রদান করে এবং ক্লাস স্ট্র্যাটেজি বদলে নেওয়া যাতে ক্লাসের আবারও ব্যবহারকারীর কোনও প্রতিস্থাপন না করে কাজ করতে পারে।

MVC প্যাটার্ন (Model-View-Controller Pattern): এটি সফটওয়্যার কোডকে তিনটি পার্টে ভাগ করে - মডেল, ভিউ, এবং কন্ট্রোলার, যাতে একটি স্বাধীন ইনস্ট্যান্সের পরিবর্তন অথবা মডেলে একটি বর্গে সাবক্লাসের দ্বারা প্রতিস্থাপন করা যায় না।

**50. আপনার জন্য সেরা প্রেরিত ডিজাইন প্যাটার্ন কোনটি এবং তা কীভাবে ব্যবহার করা হয়?**

আমি কোনটি সেরা প্রেরিত ডিজাইন প্যাটার্ন হয় তা স্থির রূপে বলতে পারি না, কারণ প্রোজেক্টের সংদর্ভ, প্রকার, এবং প্রয়োজনীয়তা সবসময় পরিবর্তন করে। ডিজাইন প্যাটার্নগুলি একটি প্রোজেক্টের দরকার অনুসারে ব্যবহার করা উচিত, এবং সংদর্ভের উপর নির্ভর করে সেরা প্যাটার্ন বের করা হবে।

উদাহরণস্বরূপ, যদি আপনি একটি অবজেক্ট তৈরি করতে হন যা একটি প্রতিষ্ঠানের এবং ক্যাচ প্রক্রিয়া করতে হয়, তবে ফ্যাক্টরি প্যাটার্ন ব্যবহার করা যেতে পারে। সিঙ্গেলটন প্যাটার্ন ব্যবহার করে একটি একক সার্ভারের জন্য একটি সিঙ্গেল ইনস্ট্যান্স তৈরি করতে পারেন। MVC প্যাটার্ন সাহায্য করে কোডটি মডেল, ভিউ, এবং কন্ট্রোলারের মধ্যে ভাগ করতে সাহায্য করতে পারে।

প্রোজেক্টের প্রতিটি পরিদর্শনে সেরা ডিজাইন প্যাটার্নের প্রয়োজনীয় হবে এবং সঠিক সময়ে এবং সঠিক প্রস্থানে ব্যবহার করা উচিত।