অধ্যায়-৬ ঃ সফটওয়্যার ডিজাইনের কৌশলসমূহ

ভূমিকা .....

স্ট্রাকচারড ডিজাইন .....

ফাংশন ওরিয়েন্টেড ডিজাইন .....

অবজেক্ট ওরিয়েন্টেড ডিজাইন .....

সফটওয়্যার ডিজাইন প্যাটার্ন .....

সফটওয়্যার ডিজাইনের পন্থাসমূহ .....

অনুশীলনী-৬

অতি সংক্ষিপ্ত প্রশ্নোত্তর .....

সংক্ষিপ্ত প্রশ্নোত্তর .....

রচনামূলক প্রশ্নাবলি .....

অধ্যায়-৬

সফটওয়্যার ডিজাইনের কৌশলসমূহ
(Software Design Strategies)

৬.০ ভূমিকা (Introduction) ঃ

সফটওয়্যার ডিজাইন হলো একপ্রকার ধারণাগত প্রক্রিয়া, যেখানে সফটওয়্যার রিকোয়ারমেন্টসমূহকে সফটওয়্যার ইমপ্লিমেন্টেশনে রূপান্তর করা হয়। সফটওয়্যার ডিজাইন ব্যবহারকারীর রিকোয়ারমেন্টসমূহকে একটি চ্যালেঞ্জ হিসেবে নেয়, এবং একটি উত্তম সমাধান খুঁজে বের করার চেষ্টা করে। যখন কোনো একটি সফটওয়্যার প্রাথমিকভাবে কল্পনা করা হয়, তখন এর উত্তম সমাধান ইমপ্লিমেন্টেশন করার জন্য একটি পরিকল্পনা ডিজাইন করা হয়। বিভিন্ন উপায়ে একটি সফটওয়্যার ডিজাইন তৈরি করা যায়, যা এ অধ্যায়ে বিস্তারিত আলোচনা করা হলো।

৬.১ স্ট্রাকচারড ডিজাইন (Structured design) ঃ

স্ট্রাকচারড বা কাঠামোগত ডিজাইন হলো কোনো একটি সমস্যাকে একাধিক সুসংগঠিত উপাদানের সমাধান হিসেবে ধারণা করা। এটি মূলত সলুশন ডিজাইনের সঙ্গে সংশ্লিষ্ট। স্ট্রাকচারড বা কাঠামোগত ডিজাইনের সুবিধা হচ্ছে, সমস্যাটি কীভাবে সমাধান হচ্ছে সেটি এ ধরনের ডিজাইনে আরও ভালোভাবে বুঝা যায়। এ ছাড়াও স্ট্রাকচারড ডিজাইন ডিজাইনারকে সমস্যাটি আরো সঠিকভাবে বুঝার ক্ষেত্রে সহজ করে তোলে।

স্ট্রাকচারড ডিজাইন মূলত "ডিভাইড অ্যান্ড কনকার (Divide and Conquer)" পদ্ধতিতে কাজ করে, যেখানে একটি বড় সমস্যাকে বিভিন্ন প্রকার ছোট ছোট সমস্যায় ভাগ করা হয়। পরবর্তীতে সেই ছোট ছোট সমস্যাগুলো সমাধান করা হয় এবং একটা সময় সেই সমাধানগুলো একত্র করে বড় সমস্যাটির সমাধান বের করা হয়। এখানে ছোট ছোট সমস্যাকে মডিউলের মাধ্যমে সমাধান করা হয়। মডিউলগুলো সুনির্দিষ্ট সমাধান প্রদান করার জন্য ভালোভাবে সংগঠিত করা কি না, সেই বিষয়টি স্ট্রাকচারড ডিজাইনে খেয়াল রাখা হয়। সমাধানের সময় মডিউলগুলো ক্রমানুসারে থাকে। একটি ভালো স্ট্রাকচারড ডিজাইন সবসময় মডিউলগুলোর মাঝে যোগাযোগের জন্য কিছু নিয়মনীতি অনুসরণ করে থাকে, যেমন--

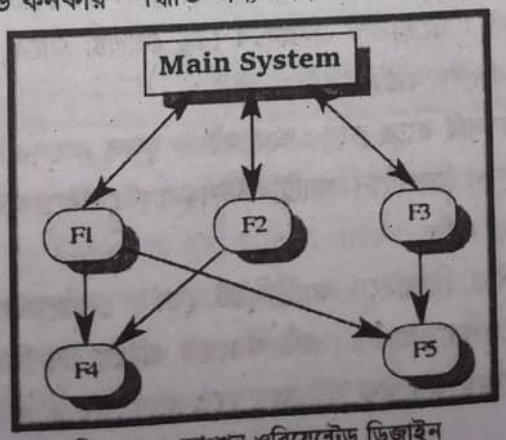
(i) কোহেশন ঃ সমস্ত কার্যকরী উপাদানসমূহের মাঝে গ্রুপিং-এর সমন্বয়।

(ii) কাপলিং ঃ বিভিন্ন প্রকার মডিউলের মাঝে যোগাযোগ।

একটি ভালো স্ট্রাকচারড ডিজাইনে উচ্চ কোহেশন এবং নিম্ন কাপলিং থাকে।

৬.২ ফাংশন ওরিয়েন্টেড ডিজাইন (Function-oriented design) ঃ

ফাংশন ওরিয়েন্টেড ডিজাইনে একটি সিস্টেমকে বিভিন্ন প্রকার সাবসিস্টেমে ভাগ করা হয়। প্রত্যেকটি সাবসিস্টেমকে ফাংশন বলা হয়। এই ফাংশনসমূহ একটা সিস্টেমের বিভিন্ন প্রকার গুরুত্বপূর্ণ কাজ করতে সক্ষম। এ প্রকার ডিজাইনে সিস্টেমটিকে সমস্ত ফাংশনের টপভিউ বা শীর্ষ-দৃশ্য কল্পনা করা হয়। ফাংশন ওরিয়েন্টেড ডিজাইনে অনেক সময় স্ট্রাকচারড ডিজাইনের কিছু বৈশিষ্ট্য ব্যবহার করা হয়, যার মধ্যে "ডিভাইড অ্যান্ড কনকার" পদ্ধতি অন্যতম।



চিত্র ঃ ৬.১ ফাংশন ওরিয়েন্টেড ডিজাইন

ফাংশন ওরিয়েন্টেড ডিজাইন পদ্ধতিতে সম্পূর্ণ সিস্টেমটিকে ছোট ছোট ফাংশনে ভাগ করা হয়, যেখানে সমস্ত ইনফরমেশন এ... এদের অপারেশন গোপন করে রাখা হয়। ইনফরমেশন গোপন করে রাখার এ প্রক্রিয়াকে সিস্টেমের অ্যাবস্ট্রাকশন প্রোপার্টি ... হয়। ছোট ছোট ফাংশনগুলো ইনফরমেশন পাসিং পদ্ধতির মাধ্যমে নিজেদের মধ্যে বিভিন্ন প্রকার তথ্য বা ইনফরমেশন আদান-প্র... করে থাকে। ফাংশনগুলোর অন্য আরেকটি বৈশিষ্ট্য হলো, যখন কোনো প্রোগ্রাম ফাংশনকে কল করে তখন ফাংশনটি প্রোগ্রাম... অবস্থা পরিবর্তন করে, যা কখনও কখনও অন্যান্য মডিউলে গ্রহণযোগ্য হয় না। ফাংশন ওরিয়েন্টেড ডিজাইন সবচেয়ে ভালো ক... করে সেই সকল সিস্টেমে, যেখানে সিস্টেমের অবস্থা (State) কোনো ব্যাপার না এবং প্রোগ্রাম বা ফাংশনগুলো একটি অবস্থা (State) পরিবর্তে ইনপুট হিসেবে কাজ করে।

ডিজাইন প্রক্রিয়া ঃ

(i) ডাটা ফ্লো ডায়াগ্রামের মাধ্যমে সিস্টেমের মধ্যে কীভাবে ডাটা প্রবাহিত হয় সেটি দেখানো হয়।

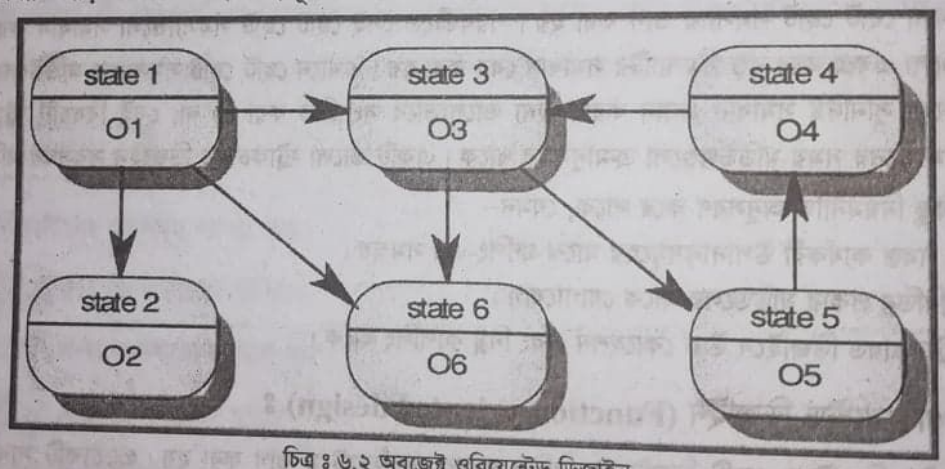
(ii) DFD-এর মাধ্যমে কীভাবে ফাংশনগুলো সম্পূর্ণ সিস্টেমের তথ্য এবং অবস্থার পরিবর্তন করে সেটি বর্ণনা করা হয়।

(iii) সম্পূর্ণ সিস্টেমটি যৌক্তিকভাবে ছোট ছোট ইউনিটে বিভক্ত করা হয়, যা সিস্টেমের মধ্যে তাদের কার্যক্রমের ভিত্তি... ফাংশন হিসাবে পরিচিত।

(iv) প্রতিটি ফাংশন বিশদভাবে বর্ণনা করা হয়।

৬.৩ অবজেক্ট ওরিয়েন্টেড ডিজাইন (Object-oriented design) ঃ

অবজেক্ট ওরিয়েন্টেড ডিজাইন সিস্টেমের ফাংশনগুলোর পরিবর্তে সিস্টেমের এনটিটি এবং এদের বৈশিষ্ট্যগুলোর উপর ভি... করে কাজ করে। এ প্রকার ডিজাইনে সিস্টেমের এনটিটি এবং এদের বৈশিষ্ট্যের উপর ফোকাস করা হয়। সফটওয়্যার সমাধা... সম্পর্কিত সমগ্র ধারণাটি জড়িত সত্তার কাছাকাছি ঘূর্ণায়মান।



চিত্র ঃ ৬.২ অবজেক্ট ওরিয়েন্টেড ডিজাইন

সংযুক্ত এনটিটিসমূহের কথা মাথায় রেখেই এ প্রকার ডিজাইনে সমস্যার সমাধান কল্পনা করা হয়। অবজেক্ট ওরিয়েন্টে... ডিজাইনের কিছু গুরুত্বপূর্ণ বৈশিষ্ট্য নিম্নে প্রদান করা হলো ঃ

অবজেক্ট ঃ সমস্যা সমাধানের জন্য ব্যবহৃত সকল এনটিটি একেকটা অবজেক্ট। উদাহরণ হিসেবে বলা যায়, কোনো ব্যাঙ্ক, কোম্পানি ইত্যাদি একেকটা অবজেক্ট। প্রত্যেকটি এনটিটির কিছু বৈশিষ্ট্য থাকে, যাদেরকে অ্যাট্রিবিউট বলা হয় এবং এ... অ্যাট্রিবিউটসমূহের অপারেশনের জন্য কিছু ফাংশন থাকে।

ক্লাস ঃ কোনো অবজেক্টের সার্বজনীন বর্ণনাই হচ্ছে ক্লাস। অবজেক্টকে মূলত ক্লাসের ইনস্ট্যান্স বলা হয়। ক্লাস সকল অ্যাট্রিবি... এবং ফাংশনকে ডিফাইন করে থাকে। সলুশন ডিজাইনে অ্যাট্রিবিউটসমূহকে ভেরিয়েবলের মতো জমা রাখা হয় এবং কতকগু... প্রসিডিউরের সাহায্যে ফাংশনালিটি বর্ণনা করা হয়।

এনক্যাপসুলেশন ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে অ্যাট্রিবিউট (ডাটা ভেরিয়েবল) এবং মেথডগুলো (ডাটার অপারেশ... একসঙ্গে সংযুক্ত করা হয়, যাকে এনক্যাপসুলেশন বলা হয়। এটি অবজেক্ট ওরিয়েন্টেড ডিজাইনের এমন একটি বৈশিষ্ট্য, যার মাধ্য... কোনো অবজেক্টের ডাটা ও ফাংশনসমূহ বাইরের অবাঞ্ছিত অ্যাক্সেস হতে রক্ষা পায় এবং ডাটা ও ফাংশনসমূহকে একত্রিত করে ... তৈরি করা যায়।

ইনহেরিটেন্স ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে যে প্রক্রিয়ায় পূর্বে ডিফাইনকৃত এক বা একাধিক ক্লাস হতে গুণাগুণ ও বৈশিষ্ট্য ইনহেরিট করে নতুন ক্লাস তৈরি করা হয়, তাকে ইনহেরিটেন্স বলে।

পলিমরফিজম ঃ পলিমরফিজমের বাংলা অর্থ বহুরূপতা। অবজেক্ট ওরিয়েন্টেড ডিজাইনে একই কাজ করার জন্য এমন কিছু পদ্ধতি প্রয়োগ করা হয়, যেখানে একই নাম বজায় রেখে ভিন্ন ভিন্ন আর্গুমেন্ট ব্যবহৃত হয়, একেই পলিমরফিজম বা বহুরূপতা বলে। এর মাধ্যমে কোনো এক ইন্টারফেসকে অনেক রকমের কাজে ব্যবহার করা যায়। ফাংশন কী চায় তার উপরে নির্ভর করে সংশ্লিষ্ট অংশে কী কোড দেয়া হবে।

৬.৪ সফটওয়্যার ডিজাইন প্যাটার্ন (Software design patterns) ঃ

সফটওয়্যার ডিজাইন প্যাটার্ন কিছু নির্দিষ্ট ধাপে সম্পন্ন করা হয়। তবে মাঝে মাঝে কিছুটা পরিবর্তন হয়, যা নির্ভর করে ডিজাইনটি ফাংশনভিত্তিক নাকি অবজেক্টভিত্তিক সেটির উপর। সফটওয়্যার ডিজাইনের প্যাটার্ন নিম্নে প্রদান করা হলো ঃ

(i) সমাধান ডিজাইন পূর্ববর্তী কোনো সিস্টেম বা সিস্টেম ডায়াগ্রাম থেকে নেয়া হতে পারে।

(ii) অবজেক্টগুলো ক্লাস এবং সংশ্লিষ্ট অ্যাট্রিবিউটের ভিত্তিতে ভাগ করা থাকে।

(iii) ক্লাসগুলোর স্তরবিন্যাস ও সম্পর্ক নির্ধারিত থাকে।

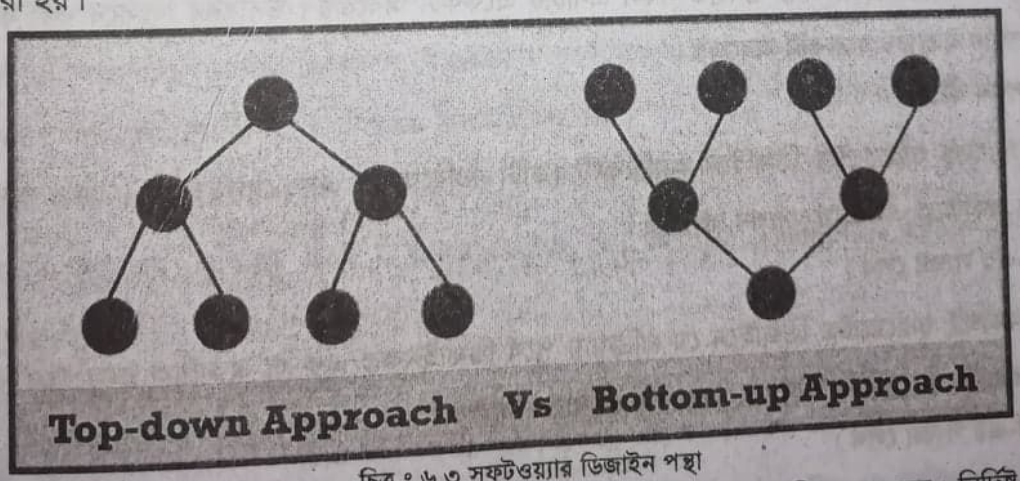
(iv) অ্যাপ্লিকেশন ফ্রেমওয়ার্ক নির্ধারিত থাকে।

৬.৫ সফটওয়্যার ডিজাইনের পন্থাসমূহ (Software design approaches) ঃ

সফটওয়্যার ডিজাইনের দুটি মৌলিক পন্থা নিচে বর্ণনা করা হলো ঃ

টপ-ডাউন ডিজাইন ঃ আমরা জানি, কোনো একটি সিস্টেম অনেকগুলো সাবসিস্টেম এবং তাদের উপাদান নিয়ে গড়ে উঠে। সেই সাবসিস্টেম আবার তাদের সাবসিস্টেম নিয়ে একটি স্তরবিন্যাস তৈরি করে। টপ-ডাউন ডিজাইন সেই পুরো সফটওয়্যার সিস্টেমটাকে একটি এনটিটি হিসেবে প্রকাশ করে এবং তারপর সেটিকে সাবসিস্টেম ও সংশ্লিষ্ট অ্যাট্রিবিউটের ভিত্তিতে ভাগ করে। তখন প্রতিটি সাবসিস্টেমকে একটি করে সিস্টেম হিসেবে ধরা হয়। একদম নিচের ধাপে স্তরবিন্যাস তৈরি হওয়ার পূর্বপর্যন্ত এ প্রক্রিয়া চলতে থাকে। টপ-ডাউন ডিজাইন একটি সরল পন্থায় শুরু হয় এবং চলতে থাকে। যখন সব উপাদান পুনঃবিন্যস্ত হয়, তখনই সম্পূর্ণ সিস্টেম অস্তিত্ব লাভ করে।

স্ক্রেচ থেকে ডিজাইন করতে গেলে এবং নির্ধারিত খুঁটিনাটি না থাকলে, সফটওয়্যার ডিজাইনে টপ-ডাউন ডিজাইন বেশি উপযুক্ত হিসেবে বিবেচনা করা হয়।



চিত্র ঃ ৬.৩ সফটওয়্যার ডিজাইন পন্থা

বটম-আপ ডিজাইন ঃ বটম আপ ডিজাইন মডেল একদম নির্দিষ্ট ও বেসিক উপাদান নিয়ে শুরু হয়। নির্দিষ্ট ও বেসিক উপাদান ব্যবহার করে ততক্ষণ পর্যন্ত উঁচু মানের উপাদান তৈরি করা হয়, যতক্ষণ পর্যন্ত একটিমাত্র উপাদানের জন্য সিস্টেম প্রস্তুত না হয়। প্রতিটি উর্ধ্বতন মানের (Higher level) জন্য অ্যাবস্ট্রাকশনের হার (Amount of abstraction) বৃদ্ধি পায়।

বিদ্যমান কোনো সিস্টেম থেকে মৌলিক ধারণা নিয়ে নতুন সিস্টেম তৈরির ক্ষেত্রে বটম-আপ ডিজাইন মডেল ব্যবহার করা যায়। টপ-ডাউন ডিজাইন এবং বটম-আপ ডিজাইন পরিপূর্ণভাবে কোনোটিই বাস্তবভিত্তিক নয়, বরং দুটির যথোপযুক্ত সমন্বয় অধিক

HP অতি সংক্ষিপ্ত প্রশ্নোত্তর ঃ

১। সফটওয়্যার ডিজাইন বলতে কী বুঝায়?

উত্তর ঃ সফটওয়্যার ডিজাইন হলো একপ্রকার ধারণাগত প্রক্রিয়া, যেখানে সফটওয়্যার রিকোয়ারমেন্টসমূহকে সফটওয়্যার ইমপ্লিমেন্টেশনে রূপান্তর করা হয়। সফটওয়্যার ডিজাইন ব্যবহারকারীর রিকোয়ারমেন্টসমূহকে একটি চ্যালেঞ্জ হিসেবে নে এবং একটি উত্তম সমাধান খুঁজে বের করার চেষ্টা করে।

২। স্ট্রাকচারড বা কাঠামোগত ডিজাইন বলতে কী বুঝায়?

উত্তর ঃ স্ট্রাকচারড বা কাঠামোগত ডিজাইন হলো কোনো একটি সমস্যাকে একাধিক সুসংগঠিত উপাদানের সমাধা হিসেবে ধারণা করা। এটি মূলত সলুশন ডিজাইনের সঙ্গে সংশ্লিষ্ট।

৩। ডিভাইড অ্যান্ড কনকার পদ্ধতি কী?

উত্তর ঃ ডিভাইড অ্যান্ড কনকার পদ্ধতি হলো কোনো একটি বড় সমস্যাকে ছোট ছোট সমস্যায় ভাগ করা। পরবর্তীতে সেই ছোট ছোট সমস্যাগুলো সমাধান করার মাধ্যমে বড় সমস্যাটি সমাধান করা হয়।

৪। ফাংশন ওরিয়েন্টেড ডিজাইনে ডাটা ফ্লো ডায়াগ্রামের কাজ কী?

উত্তর ঃ ফাংশন ওরিয়েন্টেড ডিজাইনে ডাটা ফ্লো ডায়াগ্রামের মাধ্যমে সিস্টেমের মধ্যে কীভাবে ডাটা প্রবাহিত হয় সেটি দেখানো হয়।

৫। ফাংশন ওরিয়েন্টেড ডিজাইনে DFD-এর কাজ কী?

উত্তর ঃ ফাংশন ওরিয়েন্টেড ডিজাইনে DFD-এর মাধ্যমে কীভাবে ফাংশনগুলো সম্পূর্ণ সিস্টেমের তথ্য এবং অবস্থা পরিবর্তন করে সেটি বর্ণনা করা হয়।

৬। অবজেক্ট কী?

উত্তর ঃ সমস্যা সমাধানের জন্য ব্যবহৃত সকল এনটিটি একেকটা অবজেক্ট। উদাহরণ হিসেবে বলা যায়, কোনো ব্যক্তি ব্যাংক, কোম্পানি ইত্যাদি একেকটা অবজেক্ট।

৭। এনক্যাপসুলেশন কী?

উত্তর ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে অ্যাট্রিবিউট (ডাটা ভেরিয়েবল) এবং মেথডগুলো (ডাটার অপারেশন) একসাথে সংযুক্ত করার পদ্ধতিকে এনক্যাপসুলেশন বলা হয়।

৮। ইনহেরিটেন্স-এর সংজ্ঞা লেখ।

উত্তর ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে যে প্রক্রিয়ায় পূর্বে ডিফাইনকৃত এক বা একাধিক ক্লাস হতে গুণাগুণ ও বৈশি ইনহেরিট করেন তুন ক্লাস তৈরি করা হয়, তাকে ইনহেরিটেন্স বলে।

৯। পলিমরফিজম-এর সংজ্ঞা লেখ।

উত্তর ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে একই কাজ করার জন্য এমন কিছু পদ্ধতি প্রয়োগ করা হয়, যেখানে একই ন বজায় রেখে ভিন্ন ভিন্ন আর্গুমেন্ট ব্যবহৃত হয়, একেই পলিমরফিজম বা বহুরূপতা বলে।

১০। অবজেক্ট ওরিয়েন্টেড ডিজাইনে ক্লাস কী?

উত্তর ঃ কোনো অবজেক্ট-এর সার্বজনীন বর্ণনাই হচ্ছে ক্লাস। অবজেক্টকে মূলত ক্লাস-এর ইনস্ট্যান্স বলা হয়। ক্লাস অ্যাট্রিবিউট এবং ফাংশনকে ডিফাইন করে থাকে।

সংক্ষিপ্ত প্রশ্নোত্তর ঃ

১। **ফাংশন ওরিয়েন্টেড ডিজাইন প্রক্রিয়া লেখ।**

উত্তর ঃ ফাংশন ওরিয়েন্টেড ডিজাইন সবচেয়ে ভালো কাজ করে সেই সকল সিস্টেমে, যেখানে সিস্টেমের অবস্থা (State) কোনো ব্যাপার না এবং প্রোগ্রাম বা ফাংশনগুলো একটি অবস্থার (State) পরিবর্তে ইনপুট হিসেবে কাজ করে। ফাংশন ওরিয়েন্টেড ডিজাইন প্রক্রিয়া নিম্নোক্ত ধাপে হয়ে থাকে।

ডিজাইন প্রক্রিয়া ঃ

(i) ডাটা ফ্লো ডায়াগ্রামের মাধ্যমে সিস্টেমের মধ্যে কীভাবে ডাটা প্রবাহিত হয় সেটি দেখানো হয়।

(ii) DFD-এর মাধ্যমে কীভাবে ফাংশনগুলো সম্পূর্ণ সিস্টেমের তথ্য এবং অবস্থার পরিবর্তন করে সেটি বর্ণনা করা হয়।

(iii) সম্পূর্ণ সিস্টেমটি যৌক্তিকভাবে ছোট ছোট ইউনিটে বিভক্ত করা হয়, যা সিস্টেমের মধ্যে তাদের কার্যক্রমের ভিত্তিতে ফাংশন হিসাবে পরিচিত।

(iv) প্রতিটি ফাংশন বিশদভাবে বর্ণনা করা হয়।

২। **অবজেক্ট ওরিয়েন্টেড ডিজাইনের গুরুত্বপূর্ণ বৈশিষ্ট্যগুলো উল্লেখ কর।**

উত্তর ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনের কিছু গুরুত্বপূর্ণ বৈশিষ্ট্য নিম্নে প্রদান করা হলো ঃ

অবজেক্ট ঃ সমস্যা সমাধানের জন্য ব্যবহৃত সকল এনটিটি একেকটা অবজেক্ট। উদাহরণ হিসেবে বলা যায়, কোনো ব্যক্তি, ব্যাংক, কোম্পানি ইত্যাদি একেকটা অবজেক্ট।

ক্লাস ঃ কোনো অবজেক্টের সার্বজনীন বর্ণনাই হচ্ছে ক্লাস। অবজেক্টকে মূলত ক্লাসের ইনস্ট্যান্স বলা হয়। ক্লাস সকল অ্যাট্রিবিউট এবং ফাংশনকে ডিফাইন করে থাকে।

এনক্যাপসুলেশন ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে অ্যাট্রিবিউট (ডাটা ভেরিয়েবল) এবং মেথডগুলো (ডাটার অপারেশন)) একসঙ্গে সংযুক্ত করা হয়, যাকে এনক্যাপুলেশন বলে।

ইনহেরিটেন্স ঃ অবজেক্ট ওরিয়েন্টেড ডিজাইনে যে প্রক্রিয়ায় পূর্বে ডিফাইনকৃত এক বা একাধিক ক্লাস হতে গুণাগুণ ও বৈশিষ্ট্য ইনহেরিট করে নতুন ক্লাস তৈরি করা হয়, তাকে ইনহেরিটেন্স বলে।

পলিমরফিজম ঃ পলিমরফিজমের বাংলা অর্থ বহুরূপতা। অবজেক্ট ওরিয়েন্টেড ডিজাইনে একই কাজ করার জন্য এমন কিছু পদ্ধতি প্রয়োগ করা হয়, যেখানে একই নাম বজায় রেখে ভিন্ন ভিন্ন আর্গুমেন্ট ব্যবহৃত হয়। একেই পলিমরফিজম বা বহুরূপতা বলে।

৩। **সফটওয়্যার ডিজাইন প্যাটার্ন ব্যাখ্যা কর।**

উত্তর ঃ সফটওয়্যার ডিজাইন প্যাটার্ন কিছু নির্দিষ্ট ধাপে সম্পন্ন করা হয়। তবে মাঝে মাঝে কিছুটা পরিবর্তন হয়, যা নির্ভর করে ডিজাইনটি ফাংশনভিত্তিক নাকি অবজেক্টভিত্তিক সেটির উপর। সফটওয়্যার ডিজাইন-এর প্যাটার্ন নিম্নে প্রদান করা হলো ঃ

(i) সমাধান ডিজাইন পূর্ববর্তী কোনো সিস্টেম বা সিস্টেম ডায়াগ্রাম থেকে নেয়া হতে পারে।

(ii) অবজেক্টগুলো ক্লাস এবং সংশ্লিষ্ট অ্যাট্রিবিউটের ভিত্তিতে ভাগ করা থাকে।

(iii) ক্লাসগুলোর স্তরবিন্যাস ও সম্পর্ক নির্ধারিত থাকে।

(iv) অ্যাপ্লিকেশন ফ্রেমওয়ার্ক নির্ধারিত থাকে।

৪। **টপ-ডাউন ডিজাইন ব্যাখ্যা কর।**

উত্তর ঃ আমরা জানি, কোনো একটি সিস্টেম অনেকগুলো সাবসিস্টেম এবং তাদের উপাদান নিয়ে গড়ে উঠে। সেই সা
সিস্টেম আবার তাদের সাবসিস্টেম নিয়ে একটি স্তরবিন্যাস তৈরি করে। টপ-ডাউন ডিজাইন সেই পুরো সফটওয়্যা
সিস্টেমটাকে একটি এনটিটি হিসেবে প্রকাশ করে এবং তারপর সেটিকে সাবসিস্টেম ও সংশ্লিষ্ট অ্যাট্রিবিউটের ভিত্তিতে ভা
করে। তখন প্রতিটি সাবসিস্টেমকে একটি করে সিস্টেম হিসেবে ধরা হয়। একদম নিচের ধাপে স্তরবিন্যাস তৈরি হওয়া
পূর্বপর্যন্ত এ প্রক্রিয়া চলতে থাকে। টপ-ডাউন ডিজাইন একটি সরল পন্থায় শুরু হয় এবং চলতে থাকে। যখন সব উপাদা
পুনঃবিন্যস্ত হয়, তখনই সম্পূর্ণ সিস্টেম অস্তিত্ব লাভ করে। স্ক্রেচ থেকে ডিজাইন করতে গেলে এবং নির্ধারিত খুঁটিনাটি ন
থাকলে, সফটওয়্যার ডিজাইনে টপ-ডাউন ডিজাইন বেশি উপযুক্ত হিসেবে বিবেচনা করা হয়।

৫। **বটম-আপ ডিজাইন ব্যাখ্যা কর।**

উত্তর ঃ বটম-আপ ডিজাইন মডেল একদম নির্দিষ্ট ও বেসিক উপাদান নিয়ে শুরু হয়। নির্দিষ্ট ও বেসিক উপাদান ব্যবহা
করে ততক্ষণ পর্যন্ত উঁচু মানের উপাদান তৈরি করা হয়, যতক্ষণ পর্যন্ত একটিমাত্র উপাদানের জন্য সিস্টেম প্রস্তুত না হয়
প্রতিটি উর্ধ্বতন মানের (Higher level) জন্য অ্যাবস্ট্রাকশনের হার (Amount of abstraction) বৃদ্ধি পায়।
বিদ্যমান কোনো সিস্টেম থেকে মৌলিক ধারণা নিয়ে নতুন সিস্টেম তৈরির ক্ষেত্রে বটম-আপ ডিজাইন মডেল ব্যবহার ক
যায়।

রচনামূলক প্রশ্নাবলি ঃ

১। **স্ট্রাকচারড ডিজাইন বা কাঠামোগত ডিজাইন বর্ণনা কর।**

উত্তর সংকেত ঃ ৬.১ নং অনুচ্ছেদ দ্রষ্টব্য।

২। **ফাংশন ওরিয়েন্টেড ডিজাইন বর্ণনা কর।**

উত্তর সংকেত ঃ ৬.২ নং অনুচ্ছেদ দ্রষ্টব্য।

৩। **অবজেক্ট ওরিয়েন্টেড ডিজাইন আলোচনা কর।**

উত্তর সংকেত ঃ ৬.৩ নং অনুচ্ছেদ দ্রষ্টব্য।

৪। **সফটওয়্যার ডিজাইন পন্থা বা সফটওয়্যার ডিজাইন অ্যাপ্রোচ বিস্তারিত বর্ণনা কর।**

উত্তর সংকেত ঃ ৬.৫ নং অনুচ্ছেদ দ্রষ্টব্য।