

HSC Parallel Text ICT

পঞ্চম অধ্যায় : প্রোগ্রামিং ভাষা

সার্বিক ব্যবস্থাপনায়

ঊদ্যম ICT টিম

প্রচ্ছদ

মোঃ রাকিব হোসেন

অঙ্কুর বিন্যাস

রাজু ও শাকিল

অনুপ্রেরণা ও সহযোগিতায়

মাহমুদুল হাসান সোহাগ
মুহাম্মদ আবুল হাসান লিটন

কৃতজ্ঞতা

ঊদ্যম-উন্মেষ-উত্তরণ

শিক্ষা পরিবারের সকল সদস্য

প্রকাশনায়

ঊদ্যম একাডেমিক এন্ড এডমিশন কেয়ার

প্রকাশকাল

অগাস্ট, ২০২৩ ইং

অনলাইন পরিবেশক

রকমারি ডট কম



কপিরাইট © ঊদ্যম

সমস্ত অধিকার সংরক্ষিত। এই বইয়ের কোনো অংশই প্রতিষ্ঠানের লিখিত অনুমতি ব্যতীত ফটোকপি, রেকর্ডিং, বৈদ্যুতিক বা যান্ত্রিক পদ্ধতিসহ কোনও উপায়ে পুনরুৎপাদন বা প্রতিলিপি, বিতরণ বা প্রেরণ করা যাবে না। এই শর্ত লঙ্ঘিত হলে উপযুক্ত আইনি ব্যবস্থা গ্রহণ করা হবে।

প্রিয় শিক্ষার্থী বন্ধুরা,

মাধ্যমিক জীবনের ধাপ পেরিয়ে উচ্চ মাধ্যমিক পর্যায়ে পদার্পণ করায় **ঊর্দ্ধ্বমের** পক্ষ থেকে তোমাদের সকলকে অভিনন্দন। মাধ্যমিকের পড়াশুনা থেকে উচ্চ মাধ্যমিকের পড়াশুনার ধাঁচ ভিন্ন এবং ব্যাপক। মাধ্যমিক পর্যন্ত যেখানে ‘বোর্ড বই’-ই ছিল সব, সেখানে উচ্চ-মাধ্যমিকে বিষয় ভিত্তিক নির্দিষ্ট কোনো বই নেই। কিন্তু বাজারে বোর্ড অনুমোদিত বিভিন্ন লেখকের অনেক বই পাওয়া যায়। একারণেই শিক্ষার্থীরা পাঠ্যবই বাছাইয়ের ক্ষেত্রে দ্বিধায় ভোগে। এছাড়া, মাধ্যমিকের তুলনায় উচ্চ-মাধ্যমিকে সিলেবাস বিশাল হওয়া সত্ত্বেও প্রস্তুতির জন্য খুবই কম সময় পাওয়া যায়। জীবনের অন্যতম গুরুত্বপূর্ণ এই ধাপের শুরুতেই দ্বিধা-দ্বন্দ্ব থেকে মুক্তি দিতে আমাদের এই **Parallel Text**। উচ্চ মাধ্যমিক পর্যায়ে শিক্ষার্থীদের হতাশার একটি মূখ্য কারণ থাকে পাঠ্যবইয়ের তাত্ত্বিক আলোচনা বুঝতে না পারা। এজন্য শিক্ষার্থীদের মাঝে বুঝে বুঝে পড়ার প্রতি অনীহা তৈরি হয়। তারই ফলস্বরূপ শিক্ষার্থীরা HSC ও বিশ্ববিদ্যালয় ভর্তি পরীক্ষায় ভালো ফলাফল করতে ব্যর্থ হয়।

তোমাদের লেখাপড়াকে আরও সহজ ও প্রাণবন্ত করে তোলার বিষয়টি মাথায় রেখে আমাদের **Parallel Text** বইগুলো সাজানো হয়েছে সহজ-সাবলীল ভাষায়, অসংখ্য বাস্তব উদাহরণ, গল্প, কার্টুন আর চিত্র দিয়ে। প্রতিটি টপিক নিয়ে আলোচনার পরেই রয়েছে গাণিতিক উদাহরণ; যা টপিকের বাস্তব প্রয়োগ এবং গাণিতিক সমস্যা সমাধান সম্পর্কে ধারণা দেয়ার পাশাপাশি পরবর্তী টপিকগুলো বুঝতেও সাহায্য করবে। তোমাদের বোঝার সুবিধার জন্য গুরুত্বপূর্ণ সংজ্ঞা, বৈশিষ্ট্য, পার্থক্য ইত্যাদি নির্দেশকের মাধ্যমে আলাদা করা হয়েছে। এছাড়াও যেসব বিষয়ে সাধারণত ভুল হয়, সেসব বিষয় ‘সতর্কতা’ এর মাধ্যমে দেখানো হয়েছে।

তবে শুধু বুঝতে পারাটাই কিন্তু যথেষ্ট নয়, তার পাশাপাশি দরকার পর্যাপ্ত অনুশীলন। আর এই বিষয়টি আরও সহজ করতে প্রতিটি অধ্যায়ের কয়েকটি টপিক শেষে যুক্ত করা হয়েছে ‘টপিক ভিত্তিক বিগত বছরের প্রশ্ন ও সমাধান’। যার মধ্যে বিগত বোর্ড পরীক্ষার পাশাপাশি রয়েছে বুয়েট, রুয়েট, কুয়েট, চুয়েট ও ঢাকা বিশ্ববিদ্যালয়সহ বিভিন্ন বিশ্ববিদ্যালয়ের ভর্তি পরীক্ষার প্রশ্ন ও সমাধান। এভাবে ধাপে ধাপে অনুশীলন করার ফলে তোমরা বোর্ড পরীক্ষার শতভাগ প্রশ্নের পাশাপাশি ভর্তি পরীক্ষার প্রশ্নটিও নিতে পারবে এখন থেকেই। এছাড়াও অধ্যায় শেষে রয়েছে ‘গুরুত্বপূর্ণ প্র্যাক্টিস প্রবলেম’ ও ‘গাণিতিক সমস্যাবলি’ যা অনুশীলনের মাধ্যমে তোমাদের প্রশ্নটি পূর্ণাঙ্গ হবে।

আশা করছি, আমাদের এই **Parallel Text** একই সাথে উচ্চ মাধ্যমিকে তোমাদের বেসিক গঠনে সহায়তা করে HSC পরীক্ষায় A+ নিশ্চিত করবে এবং ভবিষ্যতের বিশ্ববিদ্যালয়ের ভর্তিযুদ্ধের জন্য প্রস্তুত রাখবে।

তোমাদের সার্বিক সাফল্য ও উজ্জ্বল ভবিষ্যত কামনায়-

ঊর্দ্ধ্বম ICT টিম



পঞ্চম অধ্যায় : প্রোগ্রামিং ভাষা

ক্র.নং	বিষয়বস্তু	পৃষ্ঠা
০১	প্রোগ্রামের ধারণা	০১-০২
০২	বিভিন্ন প্রজন্মের প্রোগ্রামিং ভাষা	০২-০৪
০৩	প্রোগ্রামিং ভাষার স্তর	০৫-০৬
০৪	অনুবাদক প্রোগ্রাম	০৬-১৩
০৫	টপিক ভিত্তিক বিগত বছরের প্রশ্ন ও সমাধান	১৩-১৭
০৬	অ্যালগরিদম	১৮-২০
০৭	ফ্লোচার্ট	২০-২৭
০৮	সুডোকোড	২৭-২৭
০৯	টপিক ভিত্তিক বিগত বছরের প্রশ্ন ও সমাধান	২৭-৩০
১০	সি-প্রোগ্রামিং ভাষার ইতিহাস	৩১-৩৩
১১	সি-প্রোগ্রামের গঠন	৩৩-৩৭
১২	অপারেটর	৩৮-৪৫
১৩	ফাংশন	৪৫-৪৮
১৪	ইনপুট-আউটপুট	৪৮-৬০
১৫	কন্ট্রোল স্টেটমেন্ট	৬০-৭০
১৬	লুপ	৭০-৯২
১৭	টপিক ভিত্তিক বিগত বছরের প্রশ্ন ও সমাধান	৯২-১০৪
১৮	অ্যারের ধারণা	১০৫-১০৮
১৯	মাত্রা	১০৮-১১৬
২০	স্ট্রিং	১১৬-১১৯
২১	ইউজার ডিফাইন্ড ফাংশন	১১৯-১২৫
২২	টপিক ভিত্তিক বিগত বছরের প্রশ্ন ও সমাধান	১২৫-১২৬
২৩	একত্রে সব গুরুত্বপূর্ণ বিষয়াবলি	১২৬-১২৭
২৪	গুরুত্বপূর্ণ প্র্যাক্টিস প্রবলেম	১২৮-১৩০
২৫	গাণিতিক সমস্যাগুলি	১৩১-১৩৬

Gmail

পারস্পরিক সহযোগিতা-ই পারে পৃথিবীকে আরও সুন্দর করতে ...

সুপ্রিয় শিক্ষার্থী,

আশা করি এবারের “HSC Parallel Text” তোমাদের কাছে অনেক বেশি উপকারী হিসেবে বিবেচিত হবে ইনশাআল্লাহ্। বইটি সম্পূর্ণ ত্রুটিমুক্ত রাখতে আমরা চেষ্টার কোনো ত্রুটি করি নাই। তবুও কারো দৃষ্টিতে কোন ভুল ধরা পড়লে নিম্নে উল্লিখিত ই-মেইল এ অবহিত করলে কৃতজ্ঞ থাকবো এবং আমরা তা পরবর্তী সংস্করণে সংশোধন করে নেব ইনশাআল্লাহ্।

Email : solution.udvash@gmail.com

Email-এ নিম্নলিখিত বিষয়গুলো উল্লেখ করতে হবে:

- (i) “HSC Parallel Text” এর বিষয়ের নাম, ভার্সন (বাংলা/ইংলিশ), (ii) পৃষ্ঠা নম্বর (iii) প্রশ্ন নম্বর (iv) ভুলটা কী
- (v) কী হওয়া উচিত বলে তোমার মনে হয়

উদাহরণ: “HSC Parallel Text” ICT, Bangla Version, Page-27, Question-03, দেওয়া আছে, 2 কিন্তু হবে 6।

ভুল ছাড়াও মান উন্নয়নে যেকোনো পরামর্শ আন্তরিকভাবে গ্রহণ করা হবে। পরিশেষে মহান আল্লাহর নিকট তোমাদের সাফল্য কামনা করছি।

শুভ কামনায়
ঐচ্ছিক ICT টিম

অধ্যায় ০৫

প্রোগ্রামিং ভাষা



একদিন শাকিল ও রাজু কম্পিউটারে ভিডিও গেম খেলতে খেলতে কথা বলছিল। শাকিল রাজুকে বললো, “আচ্ছা তোমাকে যদি ১০০ টি সংখ্যা যোগ করতে দিই, পারবে?”। রাজু জানালো সে পারবে কিন্তু একটু সময় লাগবে। এমন সময় শাকিল বললো কিন্তু একটি কম্পিউটারের হয়তো কয়েক মিলিসেকেন্ড লাগবে কাজটি করতে। সে আরও বললো, মূলত দ্রুত ও নির্ভুলভাবে হিসাব-নিকাশ করার জন্যই কম্পিউটার নামক যন্ত্রটির আবিষ্কার। মহাকাশে রকেট পাঠানোর সময় প্রতি মুহূর্তে বিশাল বিশাল ও জটিল সব ক্যালকুলেশন থেকে আজকের এই ভিডিও গেমের প্রতি মুহূর্তের হিসাব-নিকাশ সবকিছুই কম্পিউটার অনায়াসেই করতে পারে। রাজু বললো, তা ঠিক আছে কিন্তু এগুলো কম্পিউটারকে আমরা শিখাবো কীভাবে?

নির্দেশনাই বা কীভাবে দিবো? তখন শাকিল জানালো, এই যে কম্পিউটারকে কোনো কাজ শিখানোর বা নির্দেশনা দেওয়ার নামই কম্পিউটার প্রোগ্রামিং। প্রোগ্রামিং এর মাধ্যমে চাইলে আমরা কম্পিউটার দিয়ে দুটো স্থানের মধ্যকার সবচেয়ে কম দূরত্বের রাস্তা নির্ণয় করতে পারবো, যেমনটা আমরা দেখি “Google Maps” এ যাকে বলে Shortest Path Algorithm। আবার কোনো একটি কলেজের কোন শিক্ষক কোন বিষয় পড়াবেন জানা থাকলে আমরা খুব দ্রুত একটি শিডিউল বানাতে পারি যে কবে কোন পিরিয়ডে কোন ক্লাস হবে যাকে বলে Scheduling Algorithm। তাহলে চলো প্রবেশ করা যাক প্রোগ্রামিং এর বিশাল জগতে।



প্রোগ্রামিং

প্রোগ্রামের ধারণা (Concept of Program)

মনে কর, তোমার হাতে একটি প্রদীপ রয়েছে যাতে আছে আলাদিনের দৈত্য। কিন্তু এ দৈত্যের একটি বড় সমস্যা আছে। সেটি হল এই দৈত্য সব করতে পারে কিন্তু কীভাবে তা করবে সেটি জানে না। কীভাবে করবে তুমি তাকে তা শেখাতে হবে। কম্পিউটার হচ্ছে, এমন একটি আলাদিনের দৈত্য যে সব করতে পারবে কিন্তু তোমাকে সুনির্দিষ্টভাবে বলে দিতে হবে ওকে আসলে কী করতে হবে। এই সুনির্দিষ্ট নির্দেশগুলোকে একত্রে প্রোগ্রাম বলা হয়। কম্পিউটারের কাজ হচ্ছে তোমার দেয়া এই নির্দেশগুলোকে পর্যায়ক্রমে পালন করা।

কোন সমস্যা সমাধানের জন্য কম্পিউটারের ভাষায় ধারাবাহিকভাবে কতগুলো কমান্ড বা নির্দেশের সমষ্টিকে প্রোগ্রাম বলা হয়। প্রোগ্রাম লেখার জন্য বিভিন্ন ধরনের প্রোগ্রামিং ভাষা ব্যবহৃত হয়। প্রোগ্রামে লিখিত নির্দেশসমূহ সোর্স প্রোগ্রাম ফাইলে সারিবদ্ধভাবে লেখা হয়। কম্পিউটার এ নির্দেশসমূহকে পর্যায়ক্রমিকভাবে পালনের মাধ্যমে নির্দিষ্ট সমস্যার সমাধান করে। প্রোগ্রামের নির্দেশসমূহ সোর্স কোড নামে পরিচিত। ইংরেজ কবি লর্ড বায়রনের কন্যা লেডি অ্যাডা অগাস্টাকে কম্পিউটার প্রোগ্রামিং ধারণার প্রবর্তক মনে করা হয়।

প্রোগ্রামের ভাষা

মনে করো, তোমার হাতে যে দৈত্য আছে তাকে তুমি বাংলায় আদেশ দিচ্ছে। কিন্তু দৈত্য আদেশ মানছে না। চিন্তা করে বলতে পারবে কেন এমন হচ্ছে? কারণটা খুবই সহজ। দৈত্য বাংলা পারে না। তাকে কমান্ড দিতে হবে তার ভাষায়, যে ভাষা সে বুঝতে পারে। একই রকমভাবে কম্পিউটারকেও কিছু বিশেষ ভাষায় কমান্ড দিতে হবে যাকে বলা হয় প্রোগ্রামের ভাষা।





কম্পিউটার সিস্টেমে প্রোগ্রাম রচনার জন্য ব্যবহৃত শব্দ, বর্ণ, অক্ষর, সংকেত এবং এগুলো বিন্যাসের নিয়মগুলোকে একত্রে বলা হয় প্রোগ্রামের ভাষা বা প্রোগ্রামিং ল্যাংগুয়েজ।

বিভিন্ন ধরনের প্রোগ্রাম রচনার জন্য বিভিন্ন ধরনের প্রোগ্রাম ভাষা ব্যবহৃত হয়। প্রোগ্রামিং-এর মাধ্যমে জটিল সমস্যা অল্প সময়ে এবং সহজে সমাধান করা যায়। প্রোগ্রাম রচনার জন্য উপযুক্ত প্রোগ্রামিং ল্যাংগুয়েজ প্রয়োজন হয়। প্রতিটি প্রোগ্রামিং ভাষারই বিশেষ কিছু পদবিন্যাস (Syntax) রয়েছে এবং সিনটেক্স অনুসারেই ঐ ভাষায় প্রোগ্রাম লিখতে হয়।

বিভিন্ন প্রজন্মের প্রোগ্রামিং ভাষা

Programming এর সূচনাকাল হতে আজ পর্যন্ত যত প্রোগ্রামিং ল্যাংগুয়েজ চলমান তাদেরকে মূলত পাঁচটি প্রজন্ম বা Generation এ ভাগ করা যায়।

প্রথম প্রজন্ম (First Generation)

এই প্রজন্মের ভাষাকে বলা হয় মেশিনের ভাষা বা Machine Language। কেমন এই মেশিনের ভাষা? চলো দেখে নেওয়া যাক।

মেশিন ল্যাংগুয়েজ (Machine Language):

কম্পিউটার একটি ডিজিটাল যন্ত্র। ডিজিটাল যন্ত্রের ভিতর কোনো কমান্ড ইনপুট দেবার সনাতন পদ্ধতি হচ্ছে বাইনারি সংখ্যা পদ্ধতি ব্যবহার করা। এই পদ্ধতিতে দুটি অক্ষর রয়েছে; 0 এবং 1। কম্পিউটারের জন্য এই ভাষা বোঝা অত্যন্ত সহজ। কারণ state বা অবস্থা মাত্র দুইটি হওয়ায় কোনো কমান্ড হয় execute করতে হবে নয়তো নয়। কিন্তু এ ভাষা মানুষের জন্য বোঝা খুবই দুঃসাধ্য। তাই মানুষের পক্ষে এ ভাষায় কোড লেখা প্রায় দুঃসাধ্য।

কম্পিউটারে প্রোগ্রামিংকে সহজ করার জন্য বিভিন্ন প্রসেসর নির্মাতা প্রতিষ্ঠান তাদের প্রসেসরের সঙ্গে তৈরি করেন একটি ইনস্ট্রাকশন সেট। ইনস্ট্রাকশন সেটে কিছু সহজ ইনস্ট্রাকশন দিয়ে দেওয়া হয় যেগুলো ব্যবহার করে প্রসেসরকে নির্দেশ দেওয়া যায় বা প্রোগ্রাম তৈরি করা যায়। কেবল 0 আর 1 ব্যবহার করার চেয়ে ইনস্ট্রাকশন সেট ব্যবহার করে প্রোগ্রাম লেখা অপেক্ষাকৃত সহজ হয়। এগুলো দ্বিতীয় প্রজন্ম থেকে দেখা যায়।



মেশিনভাষায় লেখা প্রোগ্রামকে বলে অবজেক্ট প্রোগ্রাম এবং অন্য যেকোনো ভাষায় লেখা প্রোগ্রামকে বলে উৎস প্রোগ্রাম।

মেশিন ভাষায় প্রোগ্রাম রচনার সুবিধা:

- কম্পিউটারের অভ্যন্তরীণ সার্কিট বা মেমোরি অ্যাড্রেসের সাথে সরাসরি সংযোগ স্থাপন করা সম্ভব হয়।
- এ ভাষা ব্যবহার করে কম্পিউটারে বর্তনীর ভুল-ত্রুটি সংশোধন করা যায়।
- এ ভাষা অনুশীলনের মাধ্যমে কম্পিউটারের অভ্যন্তরীণ সংগঠন সম্পর্কে জানা যায়।
- প্রোগ্রাম দ্রুত কার্যকর হয়।
- কম পরিমাণ লজিক ব্যবহার করে প্রোগ্রাম নির্বাহ করা যায়।
- কম পরিমাণ মেমোরি ব্যবহার করে প্রোগ্রাম নির্বাহ করা যায়।
- এ ভাষায় রচিত প্রোগ্রাম কম্পিউটার সরাসরি বুঝতে পারে তাই কোনো অনুবাদকের প্রয়োজন হয় না।

মেশিন ভাষায় প্রোগ্রাম রচনার অসুবিধা:

- প্রোগ্রাম রচনা অত্যন্ত ক্লান্তিকর ও সময়সাপেক্ষ।
- এক ধরনের মেশিনের জন্য ব্যবহৃত লিখিত প্রোগ্রাম অন্য ধরনের মেশিনে ব্যবহার করা যায় না।
- প্রোগ্রাম রচনার জন্য কম্পিউটারের সংগঠন সম্বন্ধে ধারণা থাকা অপরিহার্য। ডিবাগ করা কষ্টকর।
- দক্ষ প্রোগ্রামারের প্রয়োজন হয়।



দ্বিতীয় প্রজন্ম (Second Generation)

এই প্রজন্মের ভাষা অ্যাসেম্বলি ভাষা।

অ্যাসেম্বলি ভাষা (Assembly Language):

মেশিন ভাষায় কোড করা অত্যন্ত কঠিন বলে প্রোগ্রামাররা আরো সহজ ভাষা আবিষ্কারের প্রচেষ্টায় ছিল। তারই ফসল অ্যাসেম্বলি ভাষা। প্রোগ্রামারদের জন্য প্রোগ্রাম লেখা সহজতর করার জন্য মেশিন ল্যাঙ্গুয়েজের পর তৈরি হলো অ্যাসেম্বলি ল্যাঙ্গুয়েজ। এটি একটি প্রোগ্রামিং ভাষা। মেশিন ল্যাঙ্গুয়েজের চেয়ে এই ভাষায় প্রোগ্রাম লেখা ও পড়া প্রোগ্রামারদের জন্য সহজ। কম্পিউটারের প্রসেসর কিন্তু সরাসরি অ্যাসেম্বলি ল্যাঙ্গুয়েজ দিয়ে তৈরি প্রোগ্রাম রান করতে পারে না। অ্যাসেম্বলি ল্যাঙ্গুয়েজে লেখা কোডকে আগে মেশিন কোডে রূপান্তর করতে হয়, তারপর প্রসেসর সেটিকে execute করতে পারে। অ্যাসেম্বলি ল্যাঙ্গুয়েজে লেখা কোডকে মেশিন কোডে রূপান্তর করার কাজটি করে যে প্রোগ্রাম তার নাম অ্যাসেম্বলার (assembler)।

তৃতীয় প্রজন্ম (Third Generation)

অ্যাসেম্বলি ভাষা যদিও মেশিন ভাষার 0 এবং 1 থেকে প্রোগ্রামারদের মুক্তি দেয়। তবুও এতে কিছু Basic এবং Repetitive কাজ করার ক্ষেত্রে বেশ কষ্ট পোহাতে হয়। তাই 'C' নামে একটি ভাষার প্রবর্তন হল যাকে আমরা বর্তমানে বলতে পারি Mother of all programming languages পরবর্তীতে যত ভাষা তৈরি হয়েছে তার অধিকাংশ 'C' থেকে উদ্ভূত। আমরা তৃতীয় প্রজন্মের উদাহরণ হিসেবে Algol, Visual Basic এবং C এর পরিচিতি জানবো।

সি (C):

C প্রোগ্রামিং ভাষার উৎপত্তি ১৯৭০ সালে। যুক্তরাষ্ট্রের বেল ল্যাবরেটরিতে ডেনিস রিচি (Dennis Ritchie) প্রথম C ভাষা তৈরি করেন। ১৯৭৮ সাল পর্যন্ত এ ভাষা কেবল বেল ল্যাবরেটরিতে ব্যবহার করা হতো। পরবর্তীতে এ ভাষা সর্বসম্মুখে উন্মুক্ত করা হয়। C একটি উচ্চ স্তরের স্ট্রাকচার্ড প্রোগ্রামিং ভাষা। C সাধারণত সবধরনের কাজের জন্য ব্যবহৃত হয়। অর্থাৎ, C একটি General Purpose প্রোগ্রামিং ভাষা। অত্যন্ত শক্তিশালী এবং কার্যকরী ভাষা হিসেবে C এখন ব্যাপক জনপ্রিয়।

**ভিজুয়াল বেসিক (Visual Basic):**

পুরাতন বেসিক ভাষার উন্নত সংস্করণ হিসেবে মাইক্রোসফট ভিজুয়াল বেসিক (সংক্ষেপে ভিবি বা VB) ভাষাকে বাজারে আনে। এটি একটি তৃতীয় প্রজন্মের ইভেন্ট ড্রাইভেন প্রোগ্রামিং ভাষা। বর্তমানে এটি মাইক্রোসফটের COM (Component Object Model) এর IDE (Integrated Development Environment)। VB এর অন্যতম সুবিধা হলো এর দৃশ্যমান বা গ্রাফিকাল বৈশিষ্ট্য। তাছাড়া VB বেসিক ভাষার উত্তরাধিকার বলে এটি আয়ত্ত এবং ব্যবহার করা তুলনামূলকভাবে সহজ। ১৯৯৮ সালে এর ৬ষ্ঠ সংস্করণ বের হয়। এটিই ছিল ভিজুয়াল বেসিকের শেষ প্রকাশনা। বর্তমানে এই ভাষাটি Visual Basic.NET দ্বারা প্রতিস্থাপিত হয়েছে। একজন প্রোগ্রামার ভিজুয়াল বেসিকের সাথে থাকা কম্পোনেন্টের দ্বারা সহজেই একটি অ্যাপ্লিকেশন প্রোগ্রাম তৈরি করতে পারেন।

**অ্যালগল (ALGOL):**

পূর্বে IAL নামে পরিচিত, অ্যালগল (ALGOL) হলো Algorithmic Language এর সংক্ষিপ্ত রূপ। ১৯৫৮ সালে এ ভাষার উদ্ভাবন হয়। বৈজ্ঞানিক গণনা ও প্রকৌশলিক সমস্যা সমাধানের জন্য এটি ব্যবহৃত হয়। এটি একটি সার্বজনীন ভাষা অর্থাৎ, সব কম্পিউটারে ব্যবহারযোগ্য।

**চতুর্থ প্রজন্ম (Fourth Generation)**

তৃতীয় প্রজন্মের ভাষা 'C' কে বলা হয় Structural Programming Language যেখানে আমরা কখন কী করবো তা ধাপে ধাপে বর্ণনা করা থাকে। এক পর্যায়ে কিছু কিছু ক্ষেত্রে এরকম Function বা কার্যকারিতা ভিত্তিক প্রোগ্রামিং ভাষায় কোড করা কষ্টসাধ্য হয়ে পড়লো। তাই শুরু হলো নতুন ট্রেন্ড Object Oriented Programming। এখানে Object বা বস্তুকে কেন্দ্র করে তার কার্যকারিতা ধরে ধরে কোড করা হয়। যেমন গাড়ি বা Car একটি Object, তার কার্যকারিতা হচ্ছে drive, stop, park etc. আবার, Airplane এ থাকবে fly, land, takeoff etc। oop বা Object Oriented Programming ভিত্তিক ভাষাগুলোকে বলা হয় চতুর্থ প্রজন্মের ভাষা।



সি++ (C++):

১৯৮০ সালে C প্রোগ্রামিং ভাষার উন্নত রূপ হিসেবে C++ এর উদ্ভব। বিয়ার্নে স্ট্রৌস্ট্রুপ (Bjarne Stroustrup) যুক্তরাষ্ট্রের AT&T Bell Laboratory-তে এটি তৈরি করেন। C++ এক ধরনের অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং ভাষা। মূলত C প্রোগ্রামিং ভাষার সুবিধাজনক বৈশিষ্ট্যগুলোর সাথে সিমুলা 67 এর Object Oriented বৈশিষ্ট্য সমন্বয় করে C++ তৈরি হয়। ফলে এতে উভয় (উচ্চ এবং নিম্ন) স্তরের ভাষার সুবিধা বিদ্যমান থাকে। তাই একে মধ্যম স্তরের প্রোগ্রামিং ভাষা বলে। এটি সর্বকালের অন্যতম জনপ্রিয় প্রোগ্রামিং ভাষা এবং সফটওয়্যার শিল্পে এর ব্যবহার ব্যাপক। যেমন- সিস্টেম সফটওয়্যার, অ্যাপ্লিকেশন সফটওয়্যার, ডিভাইস ড্রাইভার, এম্বেডেড সফটওয়্যার, উচ্চমানের সার্ভার ও ক্লাইন্ট অ্যাপ্লিকেশন, বিনোদন সফটওয়্যার যেমন- ভিডিও গেম ইত্যাদি ক্ষেত্রে C++ ব্যবহৃত হচ্ছে। C++ এর বিভিন্ন মুক্ত এবং মালিকানাধীন কম্পাইলার আছে যা বিভিন্ন দল যেমন- জিএনইউ(GNU) প্রকল্প, মাইক্রোসফট, ইন্টেল এবং বোরল্যান্ড সরবরাহ করে।

**জাভা (Java):**

সান মাইক্রোসিস্টেমের তৈরি অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং ভাষা হচ্ছে জাভা। এটি মূলত OAK প্রোগ্রামিং ভাষার পরবর্তী সংস্করণ যা ১৯৯৫ সালের জুন মাসে বাজারে আসে। সান মাইক্রোসিস্টেম OAK প্রোগ্রামিং ভাষা তৈরি করেছিলেন মূলত হ্যান্ডহেল্ড ইলেক্ট্রনিক ডিভাইসের (হাতে বহনকারী ইলেক্ট্রনিক যন্ত্র) জন্য যা জনপ্রিয়তা পায়নি। পরবর্তীতে নাম পরিবর্তন করে এতে WWW এর ফিচারসমূহ সংযোজন করা হয়। ফলে ইন্টারনেটভুক্ত যন্ত্রসমূহে জাভা প্লাটফর্ম নামে নতুন একটা পরিবেশ তৈরি হয় যা সব ধরনের অপারেটিং সিস্টেম সাপোর্ট করে।

**পাইথন (Python):**

বর্তমান সময়ের অন্যতম জনপ্রিয় প্রোগ্রামিং ভাষা হলো পাইথন। এটি একটি উচ্চস্তরের অবজেক্ট ওরিয়েন্টেড প্রোগ্রামিং ভাষা। ১৯৯১ সালে গুইডো ভ্যান রোসাম (Guido Van Rossum) এটি প্রথম প্রকাশ করেন। পাইথন নির্মাণ করার সময় প্রোগ্রামের পঠনযোগ্যতার উপর বেশি জোর দেওয়া হয়েছে। এখানে প্রোগ্রামারের পরিশ্রমকে কম্পিউটারের চেয়ে বেশি গুরুত্ব দেওয়া হয়েছে। পাইথনের কোর সিনট্যাক্স খুবই সংক্ষিপ্ত, তবে ভাষাটির স্ট্যান্ডার্ড লাইব্রেরি অনেক সমৃদ্ধ। পাইথন ভাষার মুক্ত, কমিউনিটিভিত্তিক উন্নয়ন মডেল রয়েছে, যার দায়িত্বে আছে পাইথন সফটওয়্যার ফাউন্ডেশন নামের একটি অলাভজনক প্রতিষ্ঠান। এই ভাষাটির বিভিন্ন অংশের বিধিবদ্ধ বৈশিষ্ট্য ও আদর্শ থাকলেও পুরো ভাষাটিকে এখনো সম্পূর্ণ বিধিবদ্ধ করা হয় নাই। ১৯৯৪ সালের জানুয়ারিতে পাইথন 1.0 সংস্করণে প্রবেশ করে। এ সংস্করণে যে প্রধান বিষয়াদি যুক্ত হয় তা হলো ফাংশনাল প্রোগ্রামিং টুলস lambda, map filter a reduce সিডব্লিউআই থেকে প্রকাশিত সর্বশেষ সংস্করণ হচ্ছে পাইথন 1.2। ১৯৯৫ সালে গুইডো ভ্যান রোসাম ভার্সিনিয়ার কর্পোরেশন ফর ন্যাশনাল রিসার্চ ইনিশিয়েটিভিস (সিএনআরআই) প্রতিষ্ঠান থেকে পাইথনের ওপর তার কাজ চালিয়ে যেতে থাকেন এবং এখান থেকে সফটওয়্যারটির কয়েকটি সংস্করণ বের করেন। 1.4 সংস্করণের মধ্যে পাইথনের কিছু নতুন বৈশিষ্ট্য যোগ করা হয়। এর মধ্যে উল্লেখযোগ্য হলো মডুলা-3 থেকে উদ্ভূত হয়ে গ্রহণ করা নতুন কীওয়ার্ড আর্গুমেন্ট (যা কমন লিম্পি এর কীওয়ার্ড আর্গুমেন্টের সাথে অনেকটা মেলে) এবং জটিল সংখ্যার জন্য অভ্যন্তরীণ সমর্থন। পাইথন 2.0 তার বিভিন্ন বৈশিষ্ট্যও অনেকাংশেই এসেছে ফাংশনভিত্তিক প্রোগ্রামিং ভাষা হ্যাঙ্কেল থেকে। হ্যাঙ্কেল লিস্ট ও পাইথনের মধ্যে অনেক মিল রয়েছে যদিও হ্যাঙ্কেল যতিচিহ্নকে বেশি গুরুত্ব দেয় আর পাইথন গুরুত্ব দেয় বর্ণভিত্তিক কীওয়ার্ডের উপর। পাইথন 2.0 তে গারবেজ কালেকশন ব্যবস্থা যুক্ত হয়েছে যা নিয়মিতভাবে মেমোরি পরিষ্কার করতে সক্ষম। পাইথন এর ওয়েবসাইট ডেভেলপমেন্ট, টাস্ক অটোমেশন (task automation) ডেটা অ্যানালাসিস, কৃত্রিম বুদ্ধিমত্তা ছাড়াও আরো নানা ক্ষেত্রে python ব্যবহৃত হয়।

**পঞ্চম প্রজন্ম বা ফিফথ জেনারেশন ভাষা (১৯৮০)**

প্রোগ্রামিং ভাষা কে আরো সহজ ও শ্রমহীন করতে পঞ্চম প্রজন্মের উদ্ভব। এক্ষেত্রে মানুষের স্বাভাবিক ভাষা বা ন্যাচারাল (Natural) ল্যাংগুয়েজ ব্যবহার করা হবে। এ ধরনের ভাষায় প্রোগ্রামার দ্বারা লিখিত একটি অ্যালগরিদম ব্যবহার না করে বরং প্রোগ্রামে দেওয়া সীমাবদ্ধতা (constraint) ব্যবহার করে সমস্যা সমাধান করা হবে। ব্যবহারকারীকে কেবলমাত্র কোন সমস্যা গুলো সমাধান করা দরকার এবং কোন শর্ত পূরণ করা দরকার তা নিয়ে চিন্তা করতে হবে। সেগুলো কীভাবে সমাধান করতে হবে তার জন্য অ্যালগরিদম প্রয়োগ করতে হবে না। কৃত্রিম বুদ্ধিমত্তা ব্যবহার করে সমস্যার সমাধান করা হবে। এ ধরনের ভাষাকে মেশিনের ভাষায় রূপান্তরের জন্য ব্যবহৃত অনুবাদককে বুদ্ধিমান বা ইন্টেলিজেন্ট কম্পাইলার বলা হয়। এতে মানুষের ভাষা ব্যবহার করে কম্পিউটারের সাথে স্বাভাবিক সম্পর্ক তৈরি হবে। বর্তমানে এ প্রচেষ্টা অনেক দূর এগিয়েছে এবং এগুচ্ছে। পঞ্চম প্রজন্মের ভাষাগুলি মূলত কৃত্রিম গবেষণায় ব্যবহৃত হয়। OPS5 এবং Mercury হলো ৫ম প্রজন্মের ভাষার উদাহরণ।



প্রোগ্রামিং ভাষার স্তর**নিম্নস্তরের ভাষা (Low level Language)**

মূলত প্রথম ও দ্বিতীয় প্রজন্মের ভাষা তথা Machine Language ও Assembly Language কে প্রথম প্রজন্মের ভাষা বলে। এ স্তরের ভাষাগুলো লেখা সময়সাপেক্ষ ও দুঃসাধ্য। বর্তমানে এ ধরনের ভাষার ব্যবহার তেমন দেখা যায় না।

মধ্যম স্তরের ভাষা (Mid-Level Language)

তৃতীয় স্তরের ভাষাগুলোকে মধ্যম স্তরের বলা হয়। অ্যাসেম্বলি ল্যাঙ্গুয়েজ এবং উচ্চস্তরের ভাষার মধ্যবর্তী ভাষাকে মধ্যম স্তরের ভাষা বলে। এটি কম্পিউটারের হার্ডওয়্যার এবং প্রোগ্রামিংয়ের মাঝে একটি সেতু বন্ধন তৈরি করে দেয়। সি ল্যাঙ্গুয়েজ মধ্যম স্তরের ভাষার একটি চমৎকার উদাহরণ কারণ এটি দিয়ে একদিকে অপারেটিং সিস্টেমের মতো সিস্টেম প্রোগ্রামিং করা যায় তেমনি দৈনন্দিন ব্যবহারের জন্য অ্যাপ্লিকেশন সফটওয়্যার তৈরি করা যায়। এই ভাষার একটি বড় সুবিধা হচ্ছে এতে নিম্নস্তরের ভাষার মত কমান্ড দেবার flexibility বেশি এবং একই সাথে code লেখাও কিছুটা সহজসাধ্য।

উচ্চতর ভাষা (High Level Language)

নিম্নস্তরের ভাষার একটি বড় সমস্যা হচ্ছে এটি মেশিনের উপর নির্ভর করে। অর্থাৎ, একেক কম্পিউটারের জন্য একেক ভাষা ব্যবহার করতে হবে। তাছাড়া এসব ভাষা অনেক বেশি কঠিন। যাতে প্রোগ্রাম লেখা সহজ করা যায় তার জন্য কিছু সহজসাধ্য ভাষা আবিষ্কার করা হল। এসব ভাষাকে উচ্চস্তরের ভাষা বলা হয়।

অ্যাসেম্বলি ল্যাঙ্গুয়েজ প্রোগ্রামারদের জন্য আগের চেয়ে সহজে প্রোগ্রাম লেখার ব্যবস্থা করলেও এ ভাষায় বড় বড় প্রোগ্রাম লেখাটা অনেক কঠিন এবং সময়-সাপেক্ষ। প্রোগ্রামিং ব্যবহার করে মানুষ যখন বিভিন্ন ধরনের সমস্যার সমাধান করতে লাগল, তখন প্রয়োজন হলো এমন ধরনের প্রোগ্রামিং ভাষার, যে সব ভাষায় প্রোগ্রাম লেখা ও পড়া মানুষের জন্য অনেক বেশি সহজ হবে। তখন তৈরি হলো উচ্চ স্তরের প্রোগ্রামিং ভাষা। কোবল (Cobol), ফোরট্রান (fortran), সি (C) ইত্যাদি প্রোগ্রামিং ভাষার আবিষ্কারের ফলে প্রোগ্রামিং ভাষা অনেকখানি বদলে গেল। এসব ভাষা ব্যবহার করে বিভিন্ন সমস্যা আগের চেয়ে অনেক দ্রুত সমাধান করা সম্ভব হচ্ছে। এসব ভাষাকে উচ্চস্তরের প্রোগ্রামিং ভাষা বলা হতো। তবে সময়ের সঙ্গে আরো নতুন নতুন প্রোগ্রামিং ভাষা তৈরি হলো, যেগুলো প্রোগ্রামিং ভাষাকে আরো সহজবোধ্য করল এবং এসব ভাষা ব্যবহার করে প্রোগ্রাম ডিজাইন করাও সহজ হলো। যেমন- সি প্লাস প্লাস (C+ +), জাভা (Java), সি শার্প (C#), পিএইচপি (PHP) পাইথন (Python) ইত্যাদি। বর্তমানে এগুলোকে হাই লেভেল প্রোগ্রামিং ভাষা হিসেবে বিবেচনা করা হয়।

হাই লেভেল ল্যাংগুয়েজের কতিপয় সুবিধা:

- হাই লেভেল ভাষায় লিখিত প্রোগ্রাম যেকোনো কম্পিউটারে ব্যবহার করা যায়।
- মানুষের পক্ষে নিম্নস্তরের চেয়ে উচ্চস্তরের ভাষা শেখা সহজ।
- হাই লেভেল ভাষায় তাড়াতাড়ি প্রোগ্রাম লেখা যায়।
- নিম্নস্তরের ভাষায় চার/পাঁচটি নির্দেশের জায়গায় উচ্চস্তরের ভাষায় মাত্র একটি বাক্য লিখলেই চলে।
- হাই লেভেল ল্যাংগুয়েজে লেখা প্রোগ্রামে ভুল হবার সম্ভাবনা কম ও সংশোধন করা সহজ।
- লাইব্রেরি ফাংশন সুবিধা পাওয়া যায়।
- প্রোগ্রাম লেখার সময় হার্ডওয়্যার নিয়ে ভাবতে হয় না।
- প্রোগ্রাম তুলনামূলকভাবে ছোট হয়।

হাই লেভেল ল্যাংগুয়েজের অসুবিধা:

- এ ভাষার প্রোগ্রাম কম্পিউটার সরাসরি বুঝতে পারে না।
- প্রোগ্রাম লেখার পূর্বে স্ট্রাকচার জানতে হয়। কমান্ডের সিনটেক্স জানতে হয়।
- এ ল্যাংগুয়েজে লিখিত প্রোগ্রাম মেশিন ল্যাংগুয়েজে রূপান্তরের জন্য অনুবাদক প্রোগ্রাম প্রয়োজন হয়।
- লো-লেভেল ভাষার তুলনায় দক্ষতা কম।
- লো-লেভেল ভাষার তুলনায় বেশি মেমোরি প্রয়োজন পড়ে।
- লো-লেভেল ভাষার তুলনায় Flexibility কম।



হাই লেভেল ল্যাংগুয়েজের ব্যবহার (Use of High Level Language):

হাই লেভেল ল্যাংগুয়েজ নিম্নলিখিত কাজে ব্যবহৃত হয়-

- বড় প্রোগ্রাম তৈরির কাজে।
- বৃহৎ ডেটা প্রসেসিং-এর কাজে ব্যবহৃত প্রোগ্রাম তৈরি করতে।
- যেসব ক্ষেত্রে প্রচুর মেমোরি প্রয়োজন সেসব ক্ষেত্রের সফটওয়্যার তৈরির কাজে।
- জটিল গাণিতিক হিসাব নিকাশে ব্যবহৃত সফটওয়্যার তৈরির কাজে।
- অ্যাপ্লিকেশন প্যাকেজ সফটওয়্যার তৈরির কাজে।
- বিভিন্ন ধরনের অটোমেটিক প্রসেস কন্ট্রোলের কাজে।

হাই লেভেল ল্যাংগুয়েজ ও মেশিন ল্যাংগুয়েজের মধ্যে পার্থক্য:

মেশিন ল্যাংগুয়েজ	হাই লেভেল ল্যাংগুয়েজ
(i) কম্পিউটার যে ভাষা সরাসরি বুঝতে পারে তাকে মেশিন ল্যাংগুয়েজ বলা হয়।	(i) কম্পিউটারকে সর্বজন ব্যবহার উপযোগী করে তুলতে যে ভাষা তৈরি হয় তাই উচ্চস্তরের ভাষা।
(ii) এটি মেশিনের ভাষা বিধায় কম্পিউটারকে বুঝানোর জন্য অনুবাদের প্রয়োজন হয় না।	(ii) এ ভাষায় লিখিত প্রোগ্রাম কম্পিউটারকে বুঝানোর জন্য মেশিনের ভাষায় অনুবাদ করে দিতে হয়।
(iii) মেশিন ভাষায় প্রোগ্রাম লেখা তুলনামূলকভাবে কষ্টসাধ্য এবং সময়সাপেক্ষ ব্যাপার।	(iii) মেশিন ভাষার তুলনায় উচ্চস্তরের ভাষায় প্রোগ্রাম রচনা সহজ।
(iv) মেশিন ভাষায় লিখিত প্রোগ্রামকে অবজেক্ট প্রোগ্রাম বা বস্তু প্রোগ্রাম বলে।	(iv) উচ্চস্তরের ভাষায় লিখিত প্রোগ্রামকে Source বা উৎস প্রোগ্রাম বলে।
(v) মেশিন ভাষা হচ্ছে বাইনারি, যা 0 এবং 1 দ্বারা গঠিত।	(v) কয়েকটি উচ্চস্তরের ভাষা হচ্ছে C++, BASIC, PASCAL, FORTRAN ইত্যাদি।
(vi) এ ভাষা মেশিন সহজে বুঝতে পারে।	(vi) এ ভাষা মেশিন সহজে বুঝতে পারে না।
(vii) ভুল হওয়ার সম্ভাবনা বেশি থাকে।	(vii) ভুল হওয়ার সম্ভাবনা কম থাকে।
(viii) প্রোগ্রাম ডিবাগ করা কঠিন।	(viii) প্রোগ্রাম ডিবাগ করা অপেক্ষাকৃত সহজ।

অনুবাদক প্রোগ্রাম

উচ্চস্তরের ভাষা হল অনেকটা আমাদের স্বাভাবিক ভাষার মত। এ সকল ভাষা আমাদের জন্য সহজবোধ্য হলেও কম্পিউটার বুঝতে পারবে না। এখন Computer কে বোঝাতে হলে এই high level ভাষাকে কম্পিউটারের বোধগম্য low level ভাষাতে convert করা লাগবে।



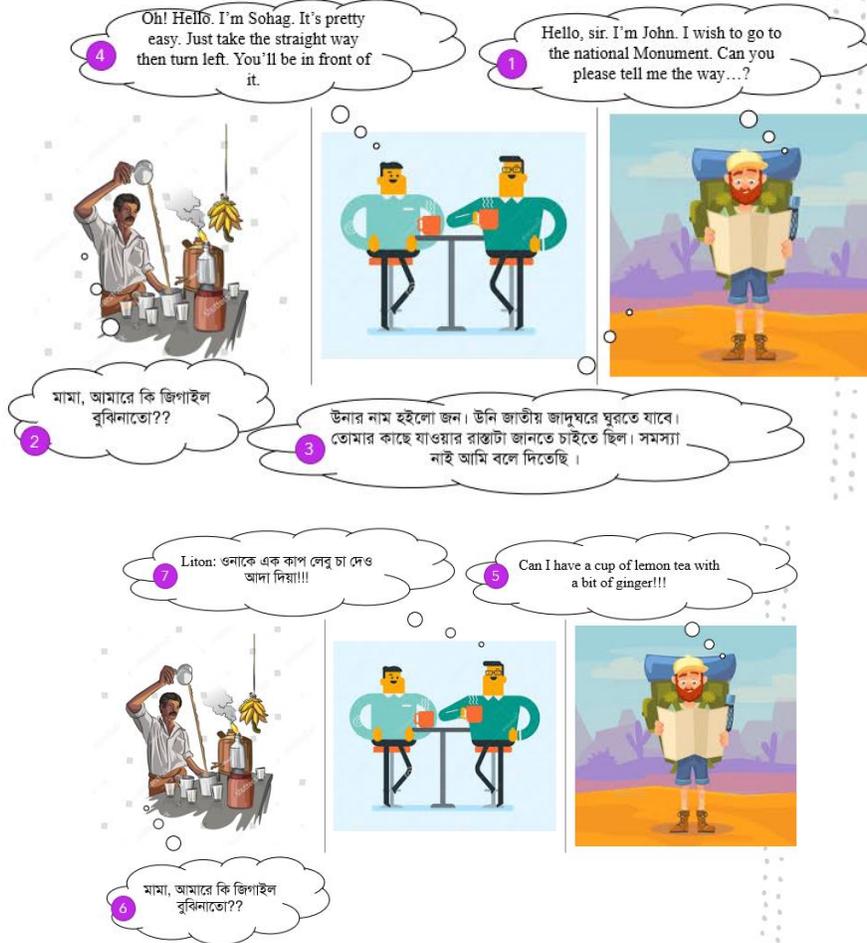
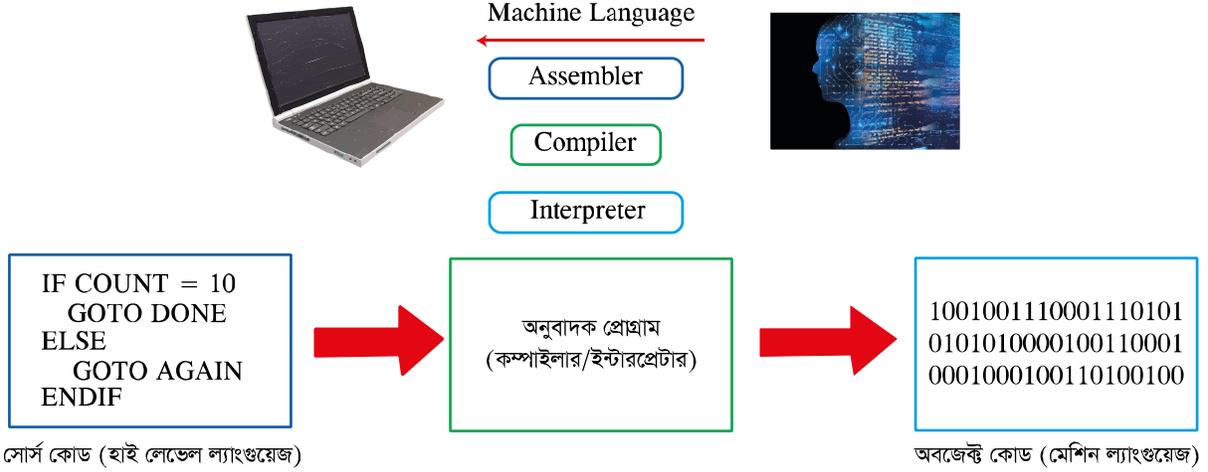
যে প্রোগ্রামের মাধ্যমে উৎস প্রোগ্রাম (Source Program) কে যন্ত্রভাষায় অনুবাদ করে বস্তু প্রোগ্রাম (Object Program) এ পরিণত করে তাকে অনুবাদক প্রোগ্রাম বলা হয়।

মূলত উচ্চস্তরের ভাষায় লিখিত প্রোগ্রামকে নিম্নস্তরের ভাষায় রূপান্তরের জন্য অনুবাদক প্রোগ্রাম ব্যবহৃত হয়। কম্পিউটার একমাত্র যন্ত্রভাষা বুঝতে পারে বলে অন্য ভাষায় লেখা উৎস প্রোগ্রামকে যন্ত্রভাষায় অনুবাদ না করলে তা কার্যকর করতে পারে না। মেশিনের ভাষা ব্যতীত অন্য যেকোন ভাষায় রচিত প্রোগ্রামকে সোর্স বা উৎস প্রোগ্রাম বলা হয় এবং মেশিনের ভাষায় রূপান্তরিত প্রোগ্রামকে অবজেক্ট বা বস্তু প্রোগ্রাম বলা হয়। তিন ধরনের অনুবাদক প্রোগ্রাম রয়েছে। যথা:

- কম্পাইলার (Compiler)
- ইন্টারপ্রেটার (Interpreter)
- অ্যাসেম্বলার (Assembler)

কিন্তু তার আগে চলো নিচের ঘটনাটি আগে একটু উপলব্ধি করি।





কম্পাইলার এবং ইন্টারপ্রেটার এর মধ্যে বেসিক পার্থক্য হচ্ছে, যদিও দুইজনের কাজই অনুবাদ করে দেয়া, কিন্তু এই কাজ দুজন করতে একটু আলাদা পদ্ধতি অবলম্বন করে।

কম্পাইলার তোমার লেখা কোড পুরোটা একসাথে ধরে একবারে অনুবাদ করে। অন্যদিকে ইন্টারপ্রেটার তোমার লেখা কোড লাইন বাই লাইন অনুবাদ করবে। আগের diagram টার দিকে খেয়াল করো। যদি মাঝের অনুবাদকেরা ব্যবহারকারীর কথা পুরোটা একবারে শুনে তারপর কম্পিউটারকে বুঝায় তবে আমরা তাকে কম্পাইলার বলব। অন্যদিকে যদি অনুবাদকেরা এক লাইন শুনতে শুনতে অনুবাদ করে যায় তবে তাকে বলব ইন্টারপ্রেটার।

অন্যদিকে অ্যাসেম্বলার একটু আলাদা ধরনের কম্পাইলার এবং ইন্টারপ্রেটার দুজনই হাই লেভেল ভাষাকে অ্যাসেম্বলি ভাষায় রূপান্তর করে এবং অ্যাসেম্বলার সেটাকে machine ভাষায় রূপান্তর করে।

কম্পাইলার (Compiler)

উচ্চস্তরের ভাষায় লেখা উৎস প্রোগ্রামকে বস্তু প্রোগ্রামে রূপান্তর করার প্রোগ্রামকে কম্পাইলার বলা হয়। কম্পাইলার সম্পূর্ণ প্রোগ্রামটিকে একসাথে পড়ে এবং একসাথে অনুবাদ করে। এ কারণে কম্পাইলার চালনার জন্য বেশি পরিমাণ মেমোরির প্রয়োজন হয়। এটি প্রোগ্রামে কোন ভুল থাকলে তা জানিয়ে দেয়। সব ভুল সংশোধন করে পুনরায় কম্পাইল করলে সোর্স প্রোগ্রাম অবজেক্ট প্রোগ্রামে রূপান্তরিত হয়। কম্পাইলার সহায়ক মেমোরিতে (Auxiliary Memory) থাকে। প্রয়োজনে র‍্যামে এনে কাজ করে। ভিন্ন ভিন্ন স্তরের ভাষার জন্য ভিন্ন ভিন্ন কম্পাইলার থাকে। কোনো একটি কম্পাইলার একটিমাত্র উচ্চস্তরের ভাষাকে যন্ত্রভাষায় পরিণত করতে পারে। একই ভাষার জন্য বিভিন্ন সফটওয়্যার কোম্পানির ভিন্ন ভিন্ন কম্পাইলার থাকতে পারে। যেমন- সি ল্যাংগুয়েজের জন্য অনেক কম্পাইলার থাকলেও GNU GCC-ই সবচেয়ে জনপ্রিয়।

উপরের প্রেক্ষাপটে আমরা যেমনটা দেখলাম সোহাগ প্রথমে জেনের পুরো কথাটাকে বুঝল তারপরে তার পুরো কথাটাকে দোকানদার মামাকে বুঝিয়ে বলল অর্থাৎ তার নিজের ভাষা যেটি মামা বুঝতে পারবেন সেই ভাষাটায় অনুবাদ করে পুরোটাই জানালো। এখানে সোহাগের কাজ যেটি হল সেটা কিন্তু প্রায় কম্পাইলারের মতই।

**কম্পাইলারের কাজ:**

- উৎস বা সোর্স প্রোগ্রামের স্টেটমেন্টসমূহকে মেশিন ভাষায় রূপান্তর।
- সংশ্লিষ্ট সাব-রুটিন এর সাথে সংযোগের ব্যবস্থা প্রদান।
- প্রধান স্মৃতির পরিসর চিহ্নিতকরণ। প্রয়োজন হলে কাগজে সোর্স প্রোগ্রাম ও অবজেক্ট প্রোগ্রামের লিখিত রূপ প্রস্তুতকরণ।
- প্রোগ্রাম ভুল থাকলে অনুবাদের সময় ভুলের তালিকা প্রণয়ন।

কম্পাইলারের সুবিধা:

- কম্পাইলার সম্পূর্ণ প্রোগ্রামটিকে একসাথে অনুবাদ করে ফলে প্রোগ্রাম নির্বাহের গতি দ্রুত হয়।
- প্রোগ্রাম নির্বাহে কম সময় লাগে।
- কম্পাইলারের মাধ্যমে রূপান্তরিত প্রোগ্রাম সম্পূর্ণরূপে মেশিন প্রোগ্রামে রূপান্তরিত হয়।
- একবার প্রোগ্রাম কম্পাইল করা হলে পরবর্তিতে আর কম্পাইলের প্রয়োজন হয় না।
- প্রোগ্রামে কোন ভুল থাকলে তা মনিটরে একসাথে প্রদর্শন করে।

কম্পাইলারের অসুবিধা:

- কম্পাইলার প্রোগ্রামের সবগুলো ভুল একসাথে প্রদর্শন করে ফলে প্রোগ্রাম সংশোধনে বেশি সময় লাগে।
- কম্পাইলার বড় ধরনের প্রোগ্রাম হওয়ায় ইহা সংরক্ষণে মেমোরিতে বেশি জায়গা লাগে।
- প্রোগ্রাম ডিবাগিং ও টেস্টিং এর কাজ ধীরগতি সম্পন্ন।

ইন্টারপ্রেটার (Interpreter)

যে অনুবাদক প্রোগ্রাম হাই লেভেল ল্যাংগুয়েজে লিখিত প্রোগ্রামের সোর্স কোডকে এক লাইন এক লাইন করে মেশিন ভাষায় অনুবাদ করে, কোন ভুল থাকলে তা সংশোধন করে পরবর্তী লাইনে কাজ করে তাকে ইন্টারপ্রেটার বলে।