TITAS
MATH SERIES

INPUT

COMPUTER TECHNIQUES

Programming with FORTRAN

ইংরেজী ও বাংলা ভার্সন

MD ELIAS HOSSAIN
MD HASAN-UL-BARI

OUTPUT

বাংলা ভাষায়
গণিতচর্চা

কম্পিউটার প্রোগ্রামিং

COMPUTER PROGRAMMING

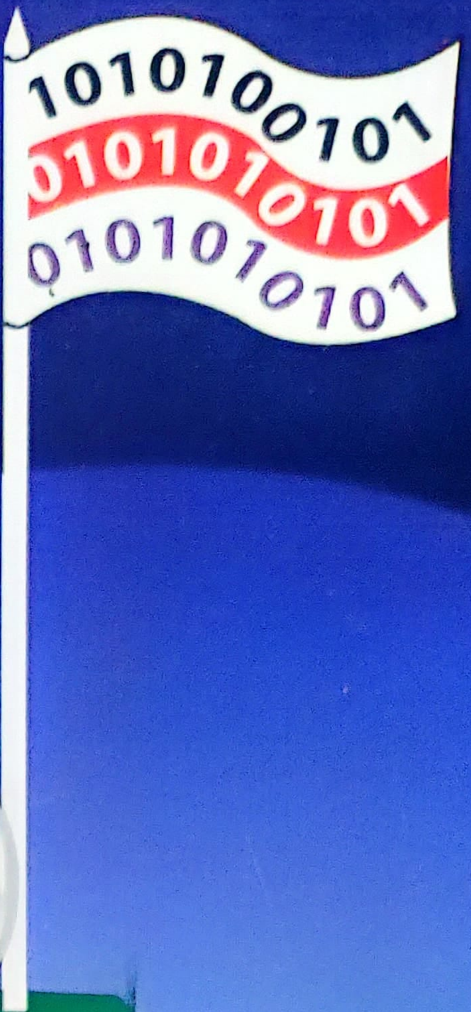
প্রফেসর মোঃ হাফিজুর রহমান
নিহার রঞ্জন বক্‌সী
অদ্বৈত কুমার দাস

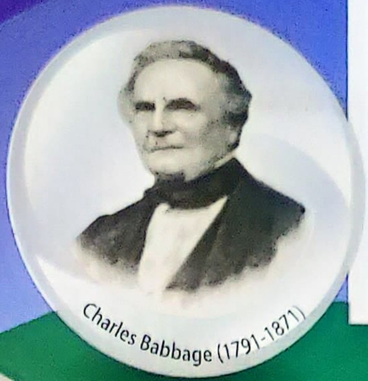
ফরট্রান প্রোগ্রামিং

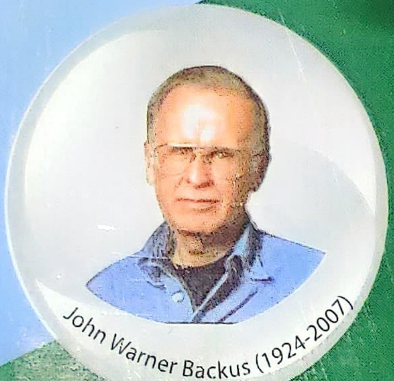
FORTRAN PROGRAMMING

AN EASIER APPROACH

- প্রফেসর ড. বি. এম. ইকরামুল হক
- ড. মোহাম্মদ সারোয়ার হোসেন







Quality Comes First
SCIENCE VIEW PUBLICATIONS

সায়েন্স ভিউ পাবলিকেশন্স
Science View Publications

অনুবাদক প্রোগ্রাম
Translator Programs

1–C.1. কম্পিউটার প্রোগ্রাম (Computer Program)

কম্পিউটারকে দিয়ে কোনো সমস্যা সমাধানের জন্য অথবা কোনো কাজ করিয়ে নেয়ার জন্য কিছু নির্দেশনা প্রদান করতে হয়। এরূপ নির্দেশনা সমষ্টিকে প্রোগ্রাম বলে। এ সমস্ত নির্দেশনা সমষ্টি সোর্স কোড বা উৎস কোড নামেও পরিচিত।

1–C.2. প্রোগ্রাম ভাষা (Programming Language) [NU(Stat)–2015, RUH–2011]

কম্পিউটারের ভাষায় লিখিত একগুচ্ছ নির্দেশনাকে বলে প্রোগ্রাম। আর এই প্রোগ্রাম যে ভাষায় লেখা হয় সেই ভাষাকে বলে প্রোগ্রাম ভাষা বা programming language. কম্পিউটারে প্রোগ্রাম রচনার জন্য ব্যবহৃত বিভিন্ন বর্ণ, প্রতীক, অঙ্ক, সংকেত এবং এগুলো ব্যবহারের নিয়মাবলি মিলিয়ে তৈরি হয় প্রোগ্রামিং ভাষা। এর সাহায্যে জটিল সমস্যা অতি অল্প সময়ে সমাধান করা যায়।

1–C.3. বিভিন্ন প্রজন্মের প্রোগ্রামিং ভাষা

এ পর্যন্ত আবিষ্কৃত কয়েকশত প্রোগ্রামিং ভাষাকে বৈশিষ্ট্য অনুযায়ী পাঁচটি স্তর বা প্রজন্মে ভাগ করা যায়। যথা–

১। ১ম প্রজন্মের ভাষা (First Generation Language) : মেশিন ভাষা (১৯৪০-১৯৪৯)

২। ২য় প্রজন্মের ভাষা (Second Generation Language) : অ্যাসেম্বেলি ভাষা (১৯৫০-১৯৫৯)

৩। ৩য় প্রজন্মের ভাষা (Third Generation Language) : উচ্চতর ভাষা (১৯৬০-১৯৭০)

৪। ৪র্থ প্রজন্মের ভাষা (Fourth Generation Language) : অতি উচ্চতর ভাষা (১৯৭১)

৫। ৫ম প্রজন্মের ভাষা বা স্বাভাবিক ভাষা (Fifth Generation/Natural Language) : স্বাভাবিক ভাষা (১৯৮০ থেকে শুরু করে ভবিষ্যতের ভাষাসমূহ)

1–C.4. মেশিন ভাষা (Machine Language) [NU(Pre)–2004]

যে ভাষা কম্পিউটার সরাসরি বুঝতে পারে তাকে মেশিন ভাষা বলে। বাইনারি ডিজিট 0 ও 1 দিয়ে মেশিন ভাষার প্রোগ্রাম লেখা হয়। বিদ্যুৎ প্রবাহের হাই ভোল্টেজকে 1 (On) এবং লো ভোল্টেজকে 0 (Off) ধরে কম্পিউটারের প্রোগ্রাম লেখার পদ্ধতিকে মেশিন ভাষা বলে। এ ভাষায় লিখিত প্রোগ্রাম কে অবজেক্ট বা বস্তুকোডও বলে।

(i) মেশিন ভাষায় প্রোগ্রাম লেখার সুবিধা (Advantages of writting programs in machine language) :

১। কম্পিউটারের অভ্যন্তরীন সার্কিট বা মেমরি ঠিকানার সাথে সরাসরি সংযোগ স্থাপন করা যায়।

২। এই ভাষায় লিখিত প্রোগ্রাম অনুবাদের প্রয়োজন হয় না।

৩। প্রোগ্রাম দ্রুত কার্যকরী হয়।

৪। লজিক কম প্রয়োগ করা লাগে।

৫। মেমরি বা স্মৃতিস্থান কম লাগে।

৬। কম্পিউটার বর্তনীর ভুল-ত্রুটি সংশোধন করা যায়।

(ii) মেশিন ভাষায় প্রোগ্রাম লেখার অসুবিধা (Disadvantages of writting programs in machine language) :

১। প্রোগ্রাম লেখার জন্য কম্পিউটারের অভ্যন্তরীন গঠন সম্পর্কে ভাল ধারণা থাকার প্রয়োজন হয়।

২। দক্ষ প্রোগ্রামারের প্রয়োজন হয়।

৩। এক মেশিনের প্রোগ্রাম অন্য মেশিনে ব্যবহার করা যায় না।

৪। ভুল-ত্রুটি বের করা জটিল।

৫। প্রোগ্রাম রচনা কষ্টকর ও সময়সাধ্য।

1–C.5. অ্যাসেম্বলি ভাষা (Assembly Language)

মেশিন ভাষার উন্নত সংস্করণ হলো অ্যাসেম্বলী ভাষা। কিছু সংকেত ব্যবহার করে এ ভাষা লেখা হয়। তাই একে সাংকেতিক ভাষাও বলে। দ্বিতীয় প্রজন্মের কম্পিউটারে এ ভাষা প্রচলিত ছিল। এর সাংকেতিক কোডকে বলা হয় নিমোনিক (Mnemonic) কোড। যেমন– LDA (Load Accumulator). অ্যাসেম্বলি ভাষার নির্দেশে চারটি অংশ থাকে। যথা- লেবেল, অপকোড, অপারেন্ড ও কমেন্ট।

1–C.6. মেশিন ভাষা ও অ্যাসেম্বলি ভাষার পার্থক্য (Difference between machine language and assembly language)

মেশিন ভাষা	অ্যাসেম্বলি ভাষা
১। বাইনারি ডিজিট 0 ও 1 দিয়ে লিখিত ভাষাকে মেশিন ভাষা বলে।	১। যে ভাষায় বিশেষ Mnemonic code ব্যবহার করা হয় তাকে অ্যাসেম্বলি ভাষা বলে।
২। মেশিন ভাষায় লিখিত প্রোগ্রামকে object program বলে।	২। এ ভাষায় লিখিত প্রোগ্রামকে Source code বলে।
৩। এ ভাষার জন্য অনুবাদক প্রোগ্রামের দরকার হয় না।	৩। অনুবাদক প্রোগ্রামের দরকার হয়।
৪। এটি মেশিন সহজে বুঝতে পারে।	৪। মেশিন অনুবাদ ছাড়া বুঝতে পারে না।
৫। ভুলত্রুটি নির্ণয় করা কঠিন।	৫। ভুল বের করা তুলনামূলক সহজ।
৬। দক্ষ প্রোগ্রামারের প্রয়োজন হয়।	৬। কম দক্ষ প্রোগ্রামার দিয়ে কোডিং করানো যায়।

1–C.7. নিম্নস্তরের ভাষা (Low level language) **[NUH(AC)–2017]**

মেশিন ভাষা এবং অ্যাসে'লি ভাষাকে একত্রে নিম্নস্তরের ভাষা বলে। নিম্নস্তরের ভাষায় প্রোগ্রাম রচনা কঠিন যদিও অ্যাসেম্বলি ভাষায় কিছু সাংকেতিক কোড ব্যবহার করা হয়।

সুবিধা (Advantages) :

১। এ ভাষা ব্যবহারে কম্পিউটারের অভ্যন্তরীন সংগঠনের ধারণা অর্জন করা যায়।

২। সবচেয়ে কম পরিমান Logic ও Memory space ব্যবহার করে এই ভাষায় প্রোগাম লেখা হয়।

৩। প্রোগ্রাম দ্রুত কার্যকরী হয়।

অসুবিধা (Disadvantages) :

১। প্রোগাম রচনা অত্যন্ত কঠিন ও সময় সাপেক্ষ।

২। এক ধরণের মেশিনের জন্য লিখিত প্রোগাম অন্য মেশিনে ব্যবহার করা যায় না।

৩। Computer নিচুস্তরের ভাষাকে সরাসরি Execute করতে পারে না।

1–C.8. উচ্চস্তরের ভাষা (High Level Language)

[NUH–03, 04, 06, 10, 18, NU(Pass)–12, NUH(AC)–17]

মানুষের মুখের ভাষার কাছাকাছি ভাষা তথা যে ভাষায় সর্বজনবোধ্য বাক্য, বর্ণ, সংখ্যা, প্রতীক ইত্যাদি ব্যবহার করে প্রোগ্রাম রচনা করা হয় তাকে উচ্চস্তরের ভাষা বলে। যেমন– C++, PASCAL, FORTRAN ইত্যাদি।

উচ্চস্তরের ভাষার সুবিধা (Advantages of high level language) : [NUH–2004, 2010, 2018]

১। উচ্চস্তরের ভাষা সহজবোধ্য।

২। যে কোনো মেশিন বুঝতে পারে।

৩। প্রোগ্রাম লেখার সময় কম লাগে ফলে কাজের গতি বৃদ্ধি পায়।

৪। ভুল কম হয় এবং ভুল হলেও তা খুঁজে বের করা সহজ।

৫। দক্ষ প্রকৌশলী দরকার হয় না।

৬। যন্ত্রাংশের উপর কম দক্ষতা লাগে।

উচ্চস্তরের ভাষার অসুবিধা (Disadvantages of high level language) : [NUH–2004, 2010]

১। উচ্চস্তরের ভাষায় লিখিত প্রোগ্রাম কম্পিউটার সরাসরি বুঝতে পারে না বলে অনুবাদকের প্রয়োজন হয়।

২। নিম্নস্তরের ভাষার চেয়ে এই ভাষায় প্রোগ্রাম নির্বাহ করতে সময় বেশি লাগে।

৩। বেশি মেমরি প্রয়োজন হয়।

৪। ভাষার নিয়ম কানুন ভালভাবে জানতে হয়।

৫। প্রোগ্রাম লেখার পূর্বে ল্যাংগুয়েজের স্ট্রাকচার জানতে হয়।

1–C.9. উচ্চস্তরের ভাষা ও নিম্নস্তরের ভাষার পার্থক্য (Differnece between high level language and low level language)

[DUH–2011]

উচ্চস্তরের ভাষা	নিম্নস্তরের ভাষা
১। কম্পিউটারের যে ভাষায় সর্বজনবোধ্য বর্ণ, প্রতীক, বাক্য, সংখ্যা ইত্যাদি ব্যবহার করে প্রোগ্রাম রচনা করা হয় তাকে উচ্চস্তরের ভাষা বলে।	১। মেশিন ভাষা ও অ্যাসেম্বলি ভাষাকে একত্রে নিম্নস্তরের ভাষা বলে।
২। এই ভাষায় লেখা প্রোগ্রামকে মেশিন ভাষায় অনুবাদের প্রয়োজন হয়।	২। অনুবাদকের প্রয়োজন হয় না তবে অ্যাসেম্বলিকে অনুবাদ করার দরকার হয়।
৩। এই ভাষায় প্রোগ্রাম লেখার জন্য যন্ত্রাংশের উপর খুব বেশি দক্ষতার প্রয়োজন হয় না।	৩। এই ভাষায় প্রোগ্রাম লেখার জন্য যন্ত্রাংশের উপর ভাল দখল থাকা প্রয়োজন।
৪। এখানে ভুল সনাক্ত করা তুলনমূলকভাবে সহজ।	৪। ভুল সনাক্ত করা কঠিন।
৫। উচ্চস্তরের ভাষায় প্রোগ্রাম লিখতে কম সময় লাগে।	৫। সময় বেশি লাগে।
৬। এই ভাষায় লিখিত প্রোগ্রাম যে কোনো কম্পিউটারে নির্বাহ করা হয়।	৬। এক মেশিনের জন্য লিখিত প্রোগ্রাম অন্য মেশিনে চলে না।
৭। দক্ষ প্রোগ্রামারের প্রয়োজন হয় না।	৭। দক্ষ প্রোগ্রামারের প্রয়োজন হয়।

1–C.10. মধ্যমস্তরের ভাষা (Mid level language)

যে প্রোগ্রামিং ভাষার মধ্যে লো-লেভেল ও হাই-লেভেল উভয় ভাষার বৈশিষ্ট্য বিদ্যমান তাকে মধ্যম স্তরের ভাষা বলে। কম্পিউটারের হার্ডওয়্যার নিয়ন্ত্রণ ও সিস্টেম প্রোগ্রাম রচনার জন্য বিট পর্যায়ের প্রোগ্রামিং ভাষা হচ্ছে মধ্যম স্তরের ভাষা। এ ভাষায় উচ্চতর ভাষার সুবিধা পাওয়া যায়। আবার নিচু স্তরের ভাষায়ও প্রোগ্রাম রচনা করা যায়। যেমন–C ভাষা।

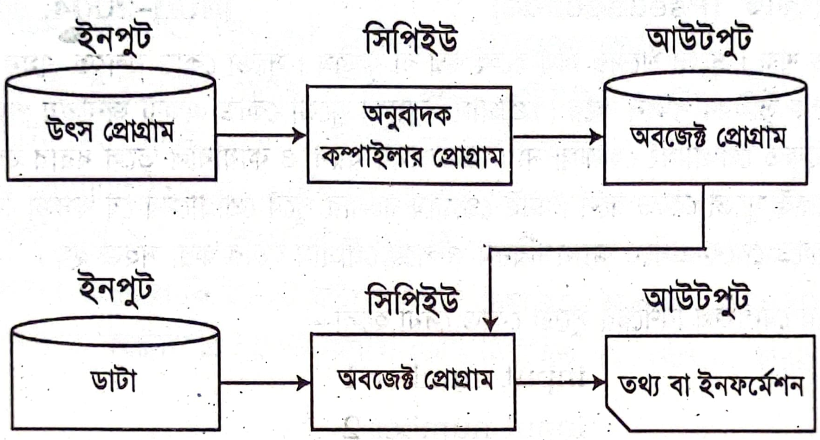
1–C.11. অনুবাদক প্রোগ্রাম (Translating Program)

নিম্নস্তর বা উচ্চতর ভাষায় লিখিত প্রোগ্রামকে সোর্স প্রোগ্রাম এবং মেশিনের ভাষায় অনুবাদ করার প্রয়োজন হয়। এই অনুবাদ করা প্রোগ্রামকে অবজেক্ট প্রোগ্রাম বলা হয়। সোর্স প্রোগ্রামকে অবজেক্ট প্রোগ্রামে পরিণত করার জন্য যে প্রোগ্রাম ব্যবহার করা হয় তাকে অনুবাদক প্রোগ্রাম বলে। সোর্স প্রোগ্রামকে অনুবাদ করার জন্য তিন ধরনের অনুবাদক প্রোগ্রাম আছে। যেমন–

১। কম্পাইলার (Compiler)   ২। অ্যাসেম্বলার (Assembler) ও   ৩। ইন্টারপ্রেটার (Interpreter)

১। কম্পাইলার (Compiler) : **[NUH–11, 15, NU(Pre)–05, DU(AC)–18]**

কম্পাইলার হলো এমন একটি অনুবাদক যা সোর্স প্রোগ্রামকে একসাথে সম্পূর্ণরূপে মেশিনের ভাষায় অনুবাদ করে একটি অবজেক্ট প্রোগ্রাম তৈরি করে এবং একটি এক্সিকিউশন ফাইল (.exe ফাইল) তৈরি করে। এখানকার এক্সিকিউশন ফাইল অত্যন্ত দ্রুততার সাথে এক্সিকিউট হয়ে থাকে। ফলে প্রোগ্রামিং প্রক্রিয়া অত্যন্ত দ্রুততর হয় এবং প্রোগ্রামারকে নিজ হাতে কোনো অনুবাদকের কাজ করতে হয় না।



২। অ্যাসেম্বলার (Assembler) : **[DUH(AC)–2018]**

অ্যাসেম্বলার হচ্ছে অ্যাসেম্বলি ভাষায় লিখিত উৎস প্রোগ্রাম অর্থাৎ নেমোনিক কোডকে মেশিনের ভাষায় লিখিত অবজেক্ট প্রোগ্রামে রূপান্তর করা এক ধরণের অনুবাদক প্রোগ্রাম। এ ভাষার অন্যতম কাজ হচ্ছে প্রধান মেমোরিতে রক্ষিত অ্যাসেম্বলি ভাষার সব নির্দেশ ঠিক আছে কিনা তা ব্যবহারকারীকে জানানো। এ ভাষার প্রত্যেকটি নির্দেশকে অ্যাসেম্বলার মেশিন ভাষার একটি নির্দেশে পরিণত করে। ভুল সংশোধনের পর এ ভাষা পুনরায় প্রথম নির্দেশ থেকে অনুবাদের কাজ শুরু করে।

৩। ইন্টারপ্রেটার (Interpreter) :

ইন্টারপ্রেটারও কম্পাইলারের মতো উচ্চতর ভাষাকে মেশিন ভাষায় রূপান্তর করে, তবে কম্পাইলার যেমন প্রথমে সোর্স প্রোগ্রামকে অবজেক্ট প্রোগ্রামে রূপান্তর করে এবং শেষে ফলাফল প্রদান করে কিন্তু ইন্টারপ্রেটার সোর্স প্রোগ্রামকে অবজেক্ট প্রোগ্রামে রূপান্তর করে না। ইন্টারপ্রেটার একটি নির্দেশ মেশিনের ভাষায় অনুবাদ করে এবং কার্যকরী করে।

প্রশ্ন ॥২.০২॥ FORTRAN এর ক্রমবিকাশ সম্পর্কে একটি সংক্ষিপ্ত নোট (Note) লিখ।
অথবা, FORTRAN IV ও Fortran-77 বলতে কি বুঝ?

সমাধান : ফোরট্রান সর্বাপেক্ষা পুরাতন হাই-লেভেল language। যা 1954-1957 সালে আবিষ্কৃত হয়েছে। সাধারণত এ ভাষা ব্যবহার করেন বিজ্ঞানী এবং প্রকৌশলীরা। Fortran Translation থেকে Fortran নামের উৎপত্তি হয়েছে। Fortran এর প্রথম ভার্সনটি উন্নয়ন করে IBM এর জন বোকাস এবং তার দল। যখন Fortran IV অবমুক্ত হয়েছিল তখন Fortran IV ছিল একটি বড় ধরনের উন্নয়ন এবং এটি পরবর্তী 15 বছর পর্যন্ত স্ট্যান্ডার্ড ভার্সন হিসেবে বিবেচিত ছিল। Fortran IV ছিল একটি বড় ধরনের উন্নয়ন। যা উচ্চতর ভার্সন কার্যকরী হিসেবে স্বীকৃত হয় এবং American National Standards Institute (ANSI) কর্তৃক ফোরট্রানের একটি নতুন স্ট্যান্ডার্ড ভার্সন অবমুক্ত করা হয়। 1978 সালে যেটি Fortran 77 নামে পরিচিত ছিল। এ ভার্সনটি খুবই জনপ্রিয় হয়। এই ANSI স্ট্যান্ডার্ড 1980 সালে (International Standards Organization (ISO) কর্তৃকও গৃহীত হয়।

কম্পিউটার ও এর সংগঠন

COMPUTER AND ITS ORGANISATION

A.1 কম্পিউটার (Computer) *[NUH-11]*

কম্পিউটার একটি ইংরেজি শব্দ। এর আভিধানিক অর্থ হতে পারে হিসাবকারী যন্ত্র। আসলে এটি হিসাবকারী যন্ত্র এবং আরো অনেক কিছু। কম্পিউটার দিয়ে গণনা করা যায়; গাণিতিক হিসাব যেমন- যোগ, বিয়োগ, গুণ, ভাগ করা যায়; এমনকি যুক্তি ও সিদ্ধান্তমূলক কাজও করা যায় কম্পিউটার দিয়ে। কাজের অনেক নির্দেশ কম্পিউটারের স্মৃতিতে সংরক্ষণ করা সম্ভব এবং প্রয়োজনে একটির পর একটি করে সে সব নির্দেশ নির্ভুলভাবে তড়িৎ গতিতে নির্বাহ করা যায়।

হিসাব, সিদ্ধান্ত ও যুক্তিমূলক সমস্যার দ্রুত ও নির্ভুল সমাধান দিয়ে মানুষের উদ্ভাবনী শক্তির কার্যকারিতা বাড়াতে সাহায্য করে কম্পিউটার। এর গতি, বিশ্বস্ততা ও নির্ভরযোগ্যতা মানুষের অনুরূপ ক্ষমতার তুলনায় অনেক উন্নত। বস্তুত মানুষের দেয়া নির্দেশ অনুসারে কম্পিউটার কাজ করে, তবে কম্পিউটার তা করে মানুষের তুলনায় অতিশয় বিশ্বস্তভাবে, অতিশয় নির্ভুলভাবে এবং অনেক দ্রুত গতিতে।

কম্পিউটার ইলেক্ট্রনিক কৌশলের একটি চমকপ্রদ প্রয়োগ; এর অভ্যন্তরে রয়েছে অনেক ইলেক্ট্রনিক বর্তনী। বিপুল কর্মশক্তি ব্যবহারের জন্য কম্পিউটারকে তার বোধগম্য ভাষায় নির্দেশ দিতে হয়। সমস্যা সমাধানের জন্য বিশেষ অনুক্রমে সাজানো এসব নির্দেশকে এক কথায় নির্দেশমালা বা প্রোগ্রাম বলে। উপযুক্ত নির্দেশের প্রভাবে কম্পিউটার জড় পদার্থ হতে গাণিতিক শক্তিসম্পন্ন একটি বুদ্ধিমান যন্ত্রে পরিণত হয়।

A.2 কম্পিউটারের সংগঠন (Computer Organisation)

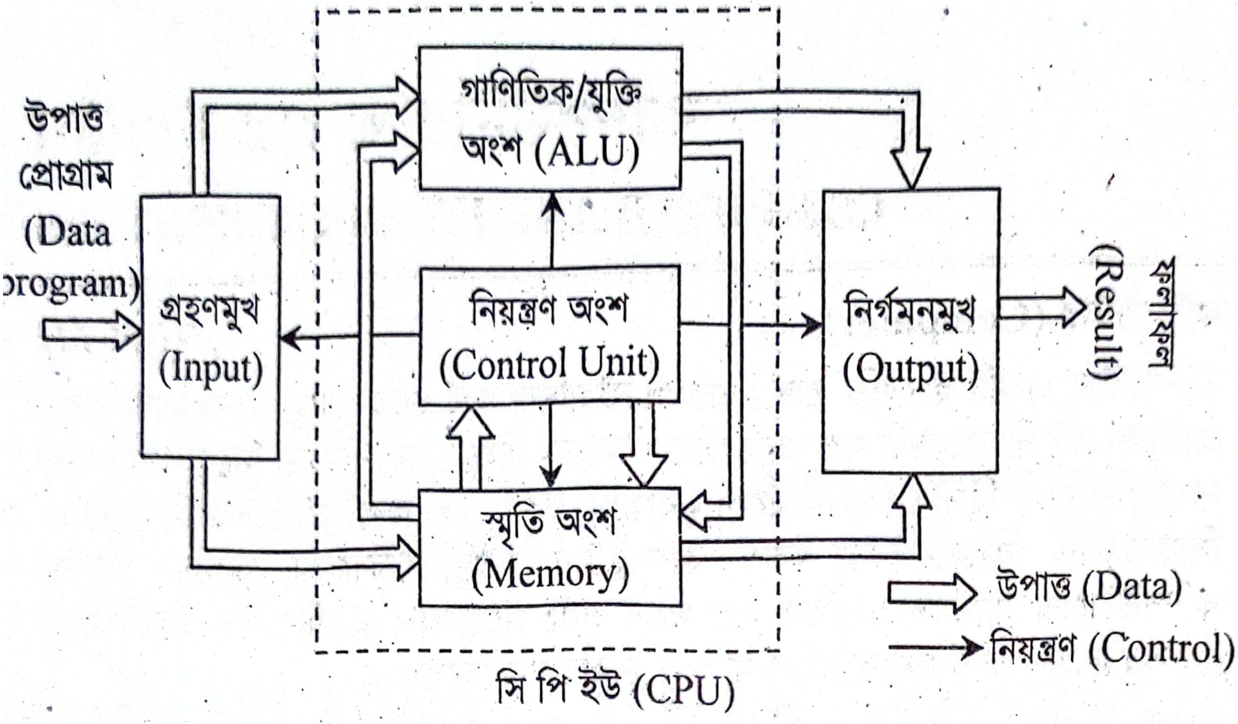
[NUH-13, 12, 09, 07; NUBScP-12, 10, 07]

কম্পিউটার একটি জটিল যন্ত্র। ইন্টিগ্রেটেড সার্কিট, মাইক্রোপ্রোসেসর, ডিস্ক ও ডিস্ক ড্রাইভ, প্রিন্টার ইত্যাদি নিয়ে এই যন্ত্রটি গঠিত। কম্পিউটারের এসব যান্ত্রিক অংশের প্রত্যেকটিকে এক একটি হার্ডওয়্যার বলা হয়। আধুনিক কম্পিউটারের প্রধান সাংগঠনিক অংশগুলি হলো:

- গ্রহণমুখ অংশ (Input Unit)
- স্মৃতি অংশ (Memory Unit)

- গাণিতিক/যুক্তি অংশ (Arithmetic/Logic Unit-ALU)
- নিয়ন্ত্রণ অংশ (Control Unit) এবং
- নির্গমনমুখ অংশ (Output Unit)

চিত্রে এই অংশসমূহের মধ্যে সংযোগ দেখানো হয়েছে। এই চিত্রে প্রশস্ত তীর চিহ্ন দ্বারা উপাত্ত স্থানান্তর পথ এবং সাধারণ তীর চিহ্ন দ্বারা নিয়ন্ত্রণ সংকেত প্রবাহের পথ দেখানো হয়েছে।



গ্রহণমুখ অংশ ঃ গ্রহণমুখ দিয়ে কম্পিউটারকে সমস্যা সমাধানের জন্য কাজের নির্দেশ দেয়া হয়। এ অংশটি বিশেষ মাধ্যম হতে উপাত্ত ও প্রোগ্রাম গ্রহণ করে বৈদ্যুতিক তরঙ্গে রূপান্তরের পর কম্পিউটারের স্মৃতিতে সংরক্ষণ করে। কীবোর্ড (Keyboard), মাউস (Mouse), কার্ড ও টেপপঠন যন্ত্র (Card and Tape Reading Device) গ্রহণমুখ অংশের উদাহরণ।

স্মৃতি অংশ ঃ কম্পিউটার স্মৃতিকে প্রধান স্মৃতি ও সহায়ক স্মৃতি এই দুই ভাগে ভাগ করা হয়। প্রধান স্মৃতি একটি পঠন/লিখন অর্ধপরিবাহী স্মৃতি। প্রক্রিয়াকরণের জন্য প্রোগ্রাম ও উপাত্তকে প্রধান স্মৃতি অংশে সংরক্ষণ করা হয়। প্রোগ্রাম নির্বাহের সময় অন্তবর্তীকালীন ফলাফলকেও এই স্মৃতিতে সংরক্ষণ করা হয়। এই স্মৃতিকে সাধারণত RAM স্মৃতি বলা হয়। স্থায়ীভাবে তথ্য সংরক্ষণের জন্য সহায়ক স্মৃতি দরকার। ফ্লপি ডিস্ক সহায়ক স্মৃতির উদাহরণ।

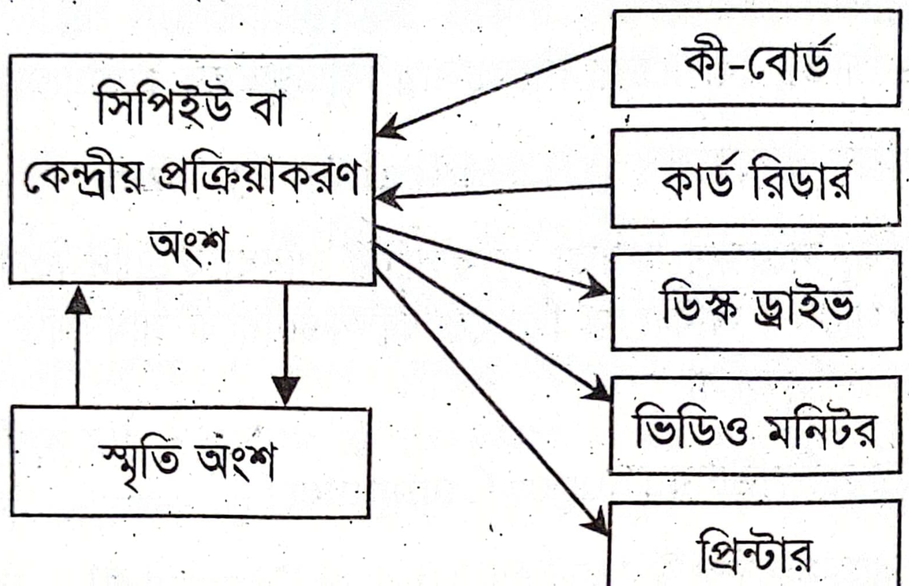
গাণিতিক/যুক্তি অংশঃ এই অংশে প্রক্রিয়াকরণের জন্য গাণিতিক, যুক্তি ও সিদ্ধান্ত মূলক কাজ সংঘটিত হয়। অংশটি কখন কি ধরনের কাজ করবে তা নিয়ন্ত্রণ অংশ নির্দিষ্ট করে দেয়।

নিয়ন্ত্রণ অংশঃ কম্পিউটার একটি প্রোগ্রাম নিয়ন্ত্রিত মেশিন। প্রোগ্রামের নির্দেশসমূহকে একটির পর একটি করে সঠিকভাবে অনুধাবনের পর নির্বাহ করা হয়। প্রোগ্রাম নির্বাহের দায়িত্ব নিয়ন্ত্রণ অংশের; এই অংশ কম্পিউটারের বিভিন্ন অংশের কাজের সমন্বয় সাধন করে।

নির্গমনমুখ অংশঃ নির্গমনমুখ অংশটি ব্যবহারকারীর অনুধাবনযোগ্য আকারে প্রক্রিয়াকরণের ফলে প্রাপ্ত ফলাফলকে উপস্থাপন করে। প্রিন্টার, টেলিটাইপরাইটার, মনিটর কয়েকটি বহুল পরিচিত নির্গমনমুখ অংশের উদাহরণ।

কেন্দ্রীয় প্রক্রিয়াকরণের অংশঃ নিয়ন্ত্রণ অংশ, অভ্যন্তরীণ স্মৃতি এবং গাণিতিক যুক্তি অংশ এই তিনটি অংশকে একসাথে কেন্দ্রীয় প্রক্রিয়াকরণ অংশ বা Central Processing Unit (CPU) বলা হয়। এটি কম্পিউটারের মস্তিষ্ক স্বরূপ। আধুনিক ক্ষুদ্র কম্পিউটার বা মাইক্রোকম্পিউটারের কেন্দ্রীয় প্রক্রিয়াকরণ অংশ হিসাবে মাইক্রোপ্রোসেসর ব্যবহার করা হয়।

কম্পিউটারের বিভিন্ন অংশকে সংযুক্ত করে কম্পিউটার পদ্ধতি তৈরী করা হয়। তবে এ সংযোগের রূপরেখা সব ধরনের কম্পিউটারে এক নয়। আগের দিনে গ্রহণমুখ, নির্গমনমুখ ও স্মৃতি অংশকে কেন্দ্রীয় প্রক্রিয়াকরণ অংশের সাথে সংযোগের জন্য পৃথক পৃথক সংযোগ ব্যবহৃত হত। এই সংযোগে খরচ বেশি হয় এবং সংযোগের উভয় মাথায় জটিল অন্তর্বর্তী বর্তনী (Interface circut) দরকার হয়। আজকাল এই ধরনের সংযোগের ব্যবহার কম হয়।



1–A.10. RAM ও ROM এর পার্থক্য (Difference Between RAM and ROM)

RAM	ROM
১. Random Access Memory কে সংক্ষেপে RAM বলে।	১. Read Only Memory কে সংক্ষেপে ROM বলে।
২. RAM অস্থায়ী স্মৃতি।	২. ROM স্থায়ী স্মৃতি
৩. RAM এর তথ্য বিদ্যুৎ চলে গেলে মুছে যায়।	৩. ROM এর তথ্য বিদ্যুৎ গেলে মুছে যায় না।
৪. RAM স্ট্যাটিক ও ডাইনামিক হতে পারে।	৪ ROM ইরেজেবল ও নন-ইরেজেবল হতে পারে।
৫. RAM পরিবর্তনযোগ্য।	৫. ROM সাধারণতঃ অপরিবর্তনযোগ্য।
৬. কম্পিউটার চালু হলে সকল তথ্য হার্ডডিস্ক থেকে র‍্যামে জমা হয়।	৬. ROM তথ্য সাধারণতঃ কোম্পানিগুলো তৈরির সময় দিয়ে দেয়।

1–A.13. সফটওয়্যার (Software) [NUH–2010, NU(Stat)–2015, NU(Pre)–2003]

কম্পিউটারকে দিয়ে যখন কোনো কাজ করানো হয় তাকে কিছু নির্দেশনা প্রদান করা হয়। কোনো নির্দিষ্ট কাজের জন্য এক গুচ্ছ নির্দেশনা একটি প্রোগ্রাম। আর অসংখ্য প্রোগ্রামের সমষ্টি একটি সফটওয়্যার।

অর্থাৎ কম্পিউটার সফটওয়্যার বলতে বোঝায় একগুচ্ছ কম্পিউটার প্রোগ্রাম, কর্মপদ্ধতি ও ব্যবহারবিধি, যার সাহায্যে কম্পিউটারে নির্দিষ্ট প্রকারের কাজ সম্পাদন করা যায়। অর্থাৎ কম্পিউটার সফটওয়্যার হলো কম্পিউটারের কিছু প্রোগ্রামের সংগ্রহ (Collection)।

কম্পিউটার সফটওয়্যার প্রধানত ৩ প্রকার যথা–

(i) সিস্টেম সফটওয়্যার (System Software)

(ii) প্রোগ্রামিং সফটওয়্যার (Programming Software)

(iii) অ্যাপ্লিকেশন সফটওয়্যার (Application Software)

1–A.14. সিস্টেম সফটওয়্যার (System Software)

[NUH(Old)–2006, BSc(Pass)–2010, NU(Pre)–2010]

যে সব কম্পিউটার সফটওয়্যার কম্পিউটারের হার্ডওয়্যারকে পরিচালনা করার জন্য এবং এপ্লিকেশন সফটওয়্যারগুলোকে কাজ করার উপযোগী পরিবেশ প্রদানের জন্য নকশা করে তৈরি করা হয়েছে সে সব সফটওয়্যার কে সিস্টেম সফটওয়্যার বলা হয়।

1–A.15. সিস্টেম সফটওয়ার এর প্রয়োজনীয়তা/কাজ (Necessity of System Software)

[BSc(Pass)–2010]

১। কম্পিউটারের হার্ডওয়ারকে একত্র করে কাজের উপযোগী করে।

২। এপ্লিকেশন সফটওয়্যার চালানোর জন্য প্লাটফর্ম তৈরি করে।

৩। কম্পিউটারের বিভিন্ন প্রোগ্রাম এবং হার্ডওয়্যার এর মাঝে সমন্বয় সাধন করে।

৪। আপনার দেয়া বিভিন্ন নির্দেশনা অনুযায়ী হার্ডওয়্যার কে পরিচালনা করে।

৫। কম্পিউটার-কে, কি করতে হবে, কিভাবে করতে হবে এসব আদেশ দেয়ার জন্য সিস্টেম সফটওয়ার ব্যাবহার করা হয়।

সিস্টেম সফটওয়্যার গুলো সাধারনত তিন প্রকার। যথা ঃ–

(i) অপারেটিং সিস্টেম (Operating systems)

(ii) ভাষা অনুবাদক (Language translators) (Assembler, compiler, Interpreter)

(iii) ইউটিলিটি প্রোগ্রাম (Utility programs)

1–A.16. অপারেটিং সিস্টেম (Operating System) [NUH–07, 10, NU(Pre)–03, 12]

অপারেটিং সিস্টম হলো এমন একটি পরিচালনা পদ্ধতি যার মাধ্যমে কোনো ব্যবহারকারী তার প্রয়োজন অনুযায়ী নির্দেশ প্রদান করে কম্পিউটারের অভ্যন্তরে হার্ডওয়ার, সফটওয়ারগুলোর মধ্যে কাজের সমন্বয় করে সমগ্র প্রক্রিয়া পরিচালনা করে। অর্থাৎ যা কম্পিউটারের সকল অংশকে একত্রে কাজ করায় তাকে অপারেটিং সিস্টেম বলে। যেমন ঃ ডেটা আদান প্রদান, প্রদর্শন এবং আউটপুট তৈরি করা। এটা এপ্লিকেশন সফটওয়্যার এবং উচ্চ-স্তরের সিস্টেম সফটওয়্যার চালানোর একটি প্লাটফর্ম। যেমন– উইন্ডোজ ১০ (Windows 10), লিনাক্স (Linux), ম্যাক ওএস, উবুন্টু ইত্যাদি।

অপারেটিং সিস্টেম ২ প্রকার (i) বর্ণভিত্তিক (ii) চিত্রভিত্তিক।

অপারেটিং সিস্টেমের গুরুত্ব/কাজ (Necessity of operating System) : **[NUH-2007, 2010]**

1. কম্পিউটার সচল ও ব্যবহার উপযোগী করা।
2. ব্যবহারকারীর প্রয়োজন অনুযায়ী বিভিন্ন এপ্লিকেশন প্রোগ্রাম মেমরীতে নিয়ে আসা এবং কাজ করা।
3. ডিস্ক ফরমেট করে কাজের উপযোগী করা।
4. বিভিন্ন সফটওয়্যারের সংযোগ সাধন, সমন্বয় সাধন, পরিচালনা ও নির্দেশ গ্রহণে সহায়তা করা।
5. ইনপুট ও আউটপুট ডিভাইসগুলো নিয়ন্ত্রন করা।

1–A.17. ইউটিলিটি সিস্টেম (Utility System)

কোনো কিছু বিশ্লেষণ, বাছাই,পছন্দ-অপছন্দ নির্ধারণ, ত্বরান্বিতকরণ এবং কিছু কিছু ক্ষেত্রে কম্পিউটার নিয়ন্ত্রন করার ক্ষমতা প্রদান করে।

1–A.18. মিডলওয়্যার (Middleware)

কম্পিউটারের বায়োস (BIOS) এবং ডিভাইস ফার্মওয়্যার (firmware) সিস্টেম সফটওয়্যার কম্পিউটারের হার্ডওয়্যার পরিচালনা এবং নিয়ন্ত্রনের কাজ করে থাকে।

1–A.19. ফার্মওয়্যার (Firmware)

যেসব নির্দেশ বা প্রোগ্রামসমূহ নির্মাতারা কম্পিউটার তৈরি করার সময় ইলেকট্রনিক ডিভাইসের মধ্যে সরবরাহ করে থাকে তাকে ফার্মওয়্যার বলে। এ প্রোগ্রামগুলো মেমরীতে সংরক্ষিত থাকে এবং সাধারণভাব পরিবর্তন করা সম্ভব নয়। কম্পিউটার অন-অফ, স্টার্ট-রিস্টার্ট, ইনপুট-আউটপুটের যে কার্যধারা তা ফার্মওয়্যারের দ্বারা নিয়ন্ত্রিত হয়। ফার্মওয়্যারকে মনিটর প্রোগ্রামও বলা হয়। ফার্মওয়্যার ও সফ্‌টওয়্যারের মধ্যে মূল পার্থক্য হচ্ছে সফ্‌টওয়্যারকে মুছা যায় কিন্তু ফার্মওয়্যারকে মুছা যায় না। যেমন– ROM, BIOS-এ তথ্যসমূহ এবং কার্য নির্দেশের তালিকা।

1–A.20. হিউম্যানওয়্যার (Humanware)

আমরা জানি হার্ডওয়্যার এবং সফ্‌টওয়্যার একে অপরের পরিপূরক। যেসব কম্পিউটার দিয়ে আমরা কাজ করে থাকি তার পূর্ণাঙ্গ রূপ দিতে বিভিন্ন পর্যায়ের মানুষের অবদান রয়েছে। প্রথমতঃ একটা কম্পিউটারের বিভিন্ন অংশ তৈরি করা হয়েছে বিভিন্ন ফ্যাক্টরীতে। তারপর এর বিভিন্ন অংশ বিভিন্ন মানুষের হাতে সংযোজিত হয়েছে। এভাবে একটি কম্পিউটারের পূর্ণাঙ্গ আকৃতি প্রদানের পর উহাতে যেসব সফ্‌টওয়্যার সন্নিবেশিত হয়েছে তার ডিজাইন, ডাটা সংরক্ষণ ও পরীক্ষাকরণ, প্রোগ্রাম লিখা, সফ্‌টওয়্যার ও হার্ডওয়্যার এর মধ্যে সমন্বয় সাধান ইত্যাদি কাজগুলো সম্পন্ন করতে বিভিন্ন ব্যাক্তি বর্গ পারস্পারিকভাবে জড়িত রয়েছেন। এভাবে জড়িত ব্যাক্তিদের সবাইকে একত্রে হিউম্যানওয়্যার বলে।

1–A.21. প্রোগ্রামিং সফটওয়্যার (Programming Software)

যে সকল কম্পিউটার সফটওয়্যার কম্পিউটার অ্যাপ্লিকেশন অথবা প্রোগ্রাম সফটওয়্যার তৈরি, ডিবাগ, নিয়ন্ত্রণ, রক্ষনাবেক্ষন এবং অন্য প্রোগ্রাম বা অ্যাপ্লিকেশনগুলোকে সহযোগিতা করতে কম্পিউটার অ্যাপ্লিকেশন অথবা প্রোগ্রাম উন্নয়নকারীগণ ব্যবহার করে থাকেন সে সকল কম্পিউটার সফটওয়্যার কে প্রোগ্রামিং সফটওয়্যার বলে। যেমন–

Fortran, C, C++, C#, Java.

1–A.22. অ্যাপ্লিকেশন সফটওয়্যার বা প্রায়োগিক সফটওয়্যার (Appli-cation Software)

[NU(Pass)–2011]

যে সকল কম্পিউটার সফটওয়্যার মানুষকে কোনো বিশেষ ধরণের কাজ সম্পাদন করতে সাহায্য করে সে সকল কম্পিউটার সফটওয়্যারকে অ্যাপ্লিকেশন সফটওয়্যার বলে। অ্যাপ্লিকেশন সফটওয়্যারকে কম্পিউটার অ্যাপ্লিকেশন বলা হয় যা বিশেষ ধরণের কম্পিউটার প্রোগ্রাম। যেমন– মাইক্রোসফট অফিস, ফটোশপ, পিডিএফ রিডার ইত্যাদি।

1–A.23. কাল্পনিক স্মৃতি (Virtual memory)

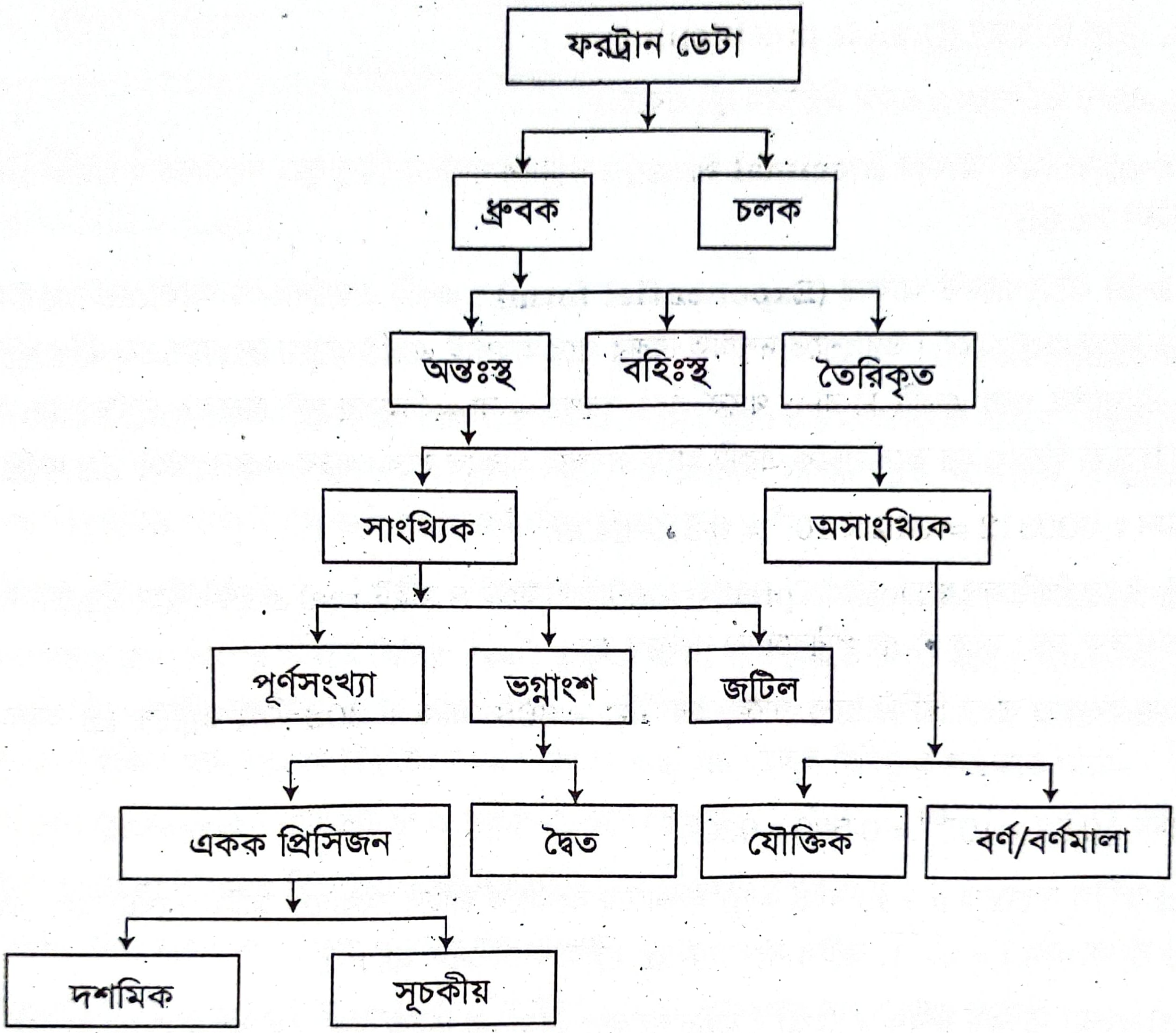
কম্পিউটারের সকল তথ্য র‍্যামে এসে জমা হওয়ার পরে কার্যক্রম সম্পাদন হয় বলে র‍্যামের জায়গা কম হলে কম্পিউটারের কাজের বিঘ্ন ঘটে। এই সমস্যা সমাধানের জন্য উইন্ডোজ বা ম্যাক অপারেটিং সিস্টেম হার্ডডিস্কের কিছু অংশকে সাময়িকভাবে র‍্যামের পরিপূরক হিসাবে ব্যবহার করে। হার্ডডিস্কের এই স্মৃতিকে কাল্পনিক স্মৃতি বলে।

1–A.24. হার্ডওয়্যার ও সফটওয়্যার এর পার্থক্য (Difference between hardware and software) [NU(Pre)–2003]

হার্ডওয়্যার	সফটওয়্যার
১. কম্পিউটারের ভৌত অংশকে হার্ডওয়্যার বলে।	১. সফটওয়্যার বলতে প্রোগ্রামসমূহকে বুঝায়, যা একগুচ্ছ নির্দেশনা।
২. এটি দৃশ্যমান।	২. এটি অদৃশ্যভাবে কাজ করে।
৩. হার্ডওয়্যার বা যন্ত্রাংশ পরিবর্তনযোগ্য।	৩. সফটওয়্যারের কোড পরিবর্তনযোগ্য।
৪. এটি ধাতু বা প্লাস্টিক দিয়ে তৈরি।	৫. এটি প্রোগ্রামিং কোড লিখে তৈরি করা হয়।
৫. স্পর্শযোগ্য।	৬. স্পর্শ করা যায় না।
৬. এটি ভঙ্গুর।	৭. ভাইরাস বা অন্যকোনো কারণে নষ্ট হতে পারে।
৭. উদাহরণ ঃ কীবোর্ড, মাউস, মনিটর।	৭. উদাহরণ ঃ উইন্ডোজ, এমএস অফিস, লিনাক্স।

2.7. ডেটা টাইপ (Data Types)

কম্পিউটার যে সকল বর্ণ, সংখ্যা প্রতীক ইত্যাদি নিয়ে কাজ করে সেগুলো ফরট্রান ডেটা বলে। বিভিন্ন প্রকার ডেটা নিচের চিত্রের মাধ্যমে দেখানো হলো–



2.8. ধ্রুবক (Constant) [NUH-2018, DUH(AC)-2017]

ফরট্রানে যে সকল ডেটার মান প্রোগ্রাম চলাকালীন সময়ে অপরিবর্তিত থাকে তাকে ধ্রুবক বলে। ধ্রুবক আবার তিন রকম–

(ক) অন্তঃস্থ (Intrinsic) (খ) ব্যক্ত (Explicit) (গ) উৎপন্ন (Derived)

(ক) অন্তঃস্থ বা (Intrinsic) : এটি আবার দুই রকম। যথা–

(i) সংখ্যা বিষয়ক (Numeric) (ii) সংখ্যা ব্যতীত (Non-Numeric)

(i) সংখ্যা বিষয়ক ঃ যে সকল ধ্রুবককে বিভিন্ন প্রকার সংখ্যা দ্বারা প্রকাশ করা হয় তাদের সংখ্যা বিষয়ক ধ্রুবক বলে। এটি কয়েক রকম হয়–

(a) পূর্ণ সংখ্যা বা স্থির বিন্দু ধ্রুবক (Integer or fixed constant)

(b) বাস্তব সংখ্যা বা ভাসমান বিন্দু ধ্রুবক (Real or Floating point constant)

(c) জটিল সংখ্যা ধ্রুবক (Complex constant)

2.10. চলক (Variable) [NUH–2010, 2013, 2017, 2019, DUH(AC)–2019]

ডেটা সংরক্ষণের জন্য বিশেষ নিয়মে লিখিত মেমরি লোকেশন বা স্মৃতিস্থানকে চলক বলে। সাধারণতঃ র‍্যামের কোনো cell এর মধ্যে ডেটা সংরক্ষণ করার জন্য ঐ cell এর নামকেই variable name বলে। variable name এর মধ্যে সংরক্ষিত মান প্রোগ্রাম নির্বাহের সময় পরিবর্তন করা যায়। চলক বিভিন্ন প্রকার হতে পারে–

১। **পূর্ণসংখ্যা (Integer) :** যে সকল চলক পূর্ণসংখ্যা ধারণ করে।

২। **বাস্তব সংখ্যা বা ভাসমান বিন্দু (Real Number or Floating point) :**
যে সকল চলক বাস্তব সংখ্যা তথা সাধারণ ভগ্নাংশ ধারণ করে।

৩। **জটিল সংখ্যা (Complex Number) :** যে সকল a + ib আকারের সংখ্যা ধারণ করে।

৪। **যৌক্তিক চলক (Logical Variable) :** যে সকল চলক সত্য বা মিথ্যা অর্থাৎ true বা false ধারণ করে।

৫। **বর্ণ চলক (Charater Variable) :** যে সকল চলক যেকোনো বর্ণ বা প্রতীক ধারণ করে।

2.11. Fortran ভাষায় চলক লেখার নিয়ম (Rules for Writting Fortran Variable Name) [NUH–2010, 2013, 2017, 2019, DUH(AC)–2019, BSc(Pass)–2012]

Fortran ভাষায় চলক লিখার নিয়ম নিম্নের দেওয়া হলো ঃ

১। চলকের নামের দৈর্ঘ্য 32 বিটে সর্বোচ্চ 31 বর্ণ/সংখ্যা এবং 64 বিটে 63 বর্ণ/সংখ্যা বিশিষ্ট হতে পারে (পুরাতন ভার্সনে 6 বর্ণ)।

২। চলকের নামের প্রথম বর্ণ (Character) অবশ্যই অক্ষর (A – Z বা a – z) হতে হবে। তবে Fortran 90/95 ভার্সনে–(Underscore) ব্যবহার করা যাবে। কোনো সংখ্যা দিয়ে শুরু করা যাবে না।

৩। ফরট্রান কোনো রিজার্ভ ওয়ার্ড বা সংরক্ষিত শব্দ না থাকায় যে কোনো শব্দকে চলক নাম হিসেবে ব্যবহার করা যায় তবে অর্থপূর্ণ নাম ব্যবহার করাই শ্রেয়। [2008 এর পূর্ববর্তী ভার্সনসমূহে এ নিয়ম প্রয়োজ্য নয়।]

৪। চলকের নাম i, j, k, l, m, n ইত্যাদি দিয়ে শুরু হলে তার ডেটা টাইপ হবে পূর্ণ সংখ্যা (Integer) টাইপ। অন্য বর্ণ দিয়ে শুরু হলে তার ডেটা টাইপ হবে real type.

৫। যে কোনো চলকের নামের পূর্বে identifier ব্যবহার করে তার ডেটা টাইপ নির্দিষ্ট করে দেয়া যাবে। যেমন– integer sum, real netvalue.

৬। নামের মাঝে ফাঁকা স্থান ব্যবহার করা যাবে না। যেমন– net value ব্যবহার করা যাবে না।

৭। কোনো নামের একাধিক শব্দ থাকলে সেখানে camel case ব্যবহার করা যেতে পারে। যেমন– net Value, Net Value.

৮। ফরট্রানের কোনো কীওয়ার্ড (Keyword) বা লাইব্রেরী ফাংশন (Library Function) চলকের নাম হিসাবে ব্যবহার করা যাবে না।

উদাহরণ–1. নিম্নে কিছু Fortran চলকের নাম শুদ্ধ নয়, সেগুলি বের কর এবং কেন শুদ্ধ নয় ব্যাখ্যা কর। (Some of the Fortran variable names are not correct. Find them and explain why they are not correct.) **[NUH–2018]**

(i)	X–RAY	:	অশুদ্ধ। কারণ এতে হাইপেন (–) ব্যবহার করা হয়েছে, যেটি fortran variable name লিখার নিয়ম বহির্ভূত।
(ii)	STUDENTS	:	অশুদ্ধ। কারণ variable name এর দৈর্ঘ্য Fortran–77 এ অবশ্যই ছয় (6) এর বেশী হতে পারবে না। এখানে variable টির দৈর্ঘ্য 8। ইহা Fortran–90/95 এর ক্ষেত্রে শুদ্ধ।
(iii)	b. b. c.	:	অশুদ্ধ। কারণ ডট (.) চিহ্ন ব্যবহার করা হয়েছে, যেটি variable name লিখার নিয়ম বহির্ভূত।
(iv)	x(100)	:	অশুদ্ধ। কারণ parenthesis '( )' বন্ধনী ব্যবহার করা হয়েছে, যেটি Fortran variable name লিখার নিয়ম বহির্ভূত। কিন্তু ইহা array name হলে শুদ্ধ।
(v)	2ND	:	অশুদ্ধ। কারণ প্রথম Character অবশ্যই Letter (A to Z or a to z) হতে হবে।
(vi)	A(7)	:	অশুদ্ধ। কারণ parenthesis '( )' বন্ধনী ব্যবহার করা হয়েছে, যেটি Fortran variable name লিখার নিয়ম বহির্ভূত। কিন্তু ইহা array name হলে শুদ্ধ।
(vii)	set–code	:	অশুদ্ধ। কারণ এতে Special Character হাইপেন '–' ব্যবহার করা হয়েছে, যেটি fortran variable name লিখার নিয়ম বহির্ভূত।
(viii)	Stop, real	:	অশুদ্ধ। কারণ এটি একটি Fortran statement. তবে 2008 বা পরবর্তী ভার্সনের জন্য ব্যবহার করা যাবে।
(ix)	N. U.	:	অশুদ্ধ। কারণ ডট (.) চিহ্ন ব্যবহার করা হয়েছে, যেটি variable name লিখার নিয়ম বহির্ভূত।
(x)	2 MAX	:	অশুদ্ধ। কারণ প্রথম Character অবশ্যই Letter (A to Z or a to z) হতে হবে।
(xi)	READ	:	ফরট্রানের নিজস্ব কীওয়ার্ড বলে চলক হবে না। তবে 2003 সংস্করণে ব্যবহার করা যাবে।
(xii)	NU + JU	:	অশুদ্ধ। কারণ variable name লিখার জন্য (+) প্লাস চিহ্ন ব্যবহার করার অনুমতি নাই।
(xiii)	SUM/3	:	শুদ্ধ নয়। (/) shash ফরট্রান চলকে নাম লেখার নিয়ম বহির্ভূত।
(xiv)	Num/2	:	শুদ্ধ নয়। Fortran–iv পর্যন্ত কিন্তু Fortran–77 ও পরবর্তী ভার্সনের জন্য শুদ্ধ।

(xv) 1∗5 : শুদ্ধ নয়। Asterisk (∗) চিহ্ন Fortran চলকে ব্যবহারের অনুমতি নেই।

(xvi) $2 : শুদ্ধ নয়। এখানে ডলার ($) ব্যবহার করা হয়েছে যা নিয়ম বহির্ভূত।

(xvii) A3.9 : শুদ্ধ নয়। ডেসিম্যাল পয়েন্ট (.) ব্যবহার করা হয়েছে।

(xviii) 09 MAY : শুদ্ধ নয়। প্রথম অক্ষর অবশ্যই A – Z বা a – z হতে হবে।

(xix) A∗∗B : শুদ্ধ নয়। চলকে (∗) চিহ্ন গ্রহণযোগ্য নয়।

(xx) MATHEMATICS : শুদ্ধ নয়। Fortran–77 ভার্সনে, তবে Fortran–90/95 পরবর্তী ভার্সনে সঠিক।

(xxi) BB, AA শুদ্ধ নয়। কমা ব্যবহৃত হয়েছে।

উদাহরণ–2. নীচে প্রদত্ত শব্দগুলির কোন্ কোন্‌টি Fortran চলক হিসাবে গ্রহণযোগ্য নয়, তা চিহ্নিত কর এবং কেন ঐগুলি বৈধ নয় তা ব্যাখ্যা কর [Some of the words listed below are not valid in Fortran variable names. Identify them and explain why they are not legal]

KJ–26	ATITUDE	LXYX∗Z	A58.2
25 MAY	IN-OUT	XN6	(X61)

সমাধান ঃ KJ–26 গ্রহণযোগ্য নয়। কারণ এতে Special Character হাইপেন '–' ব্যবহার করা হয়েছে, যেটি fortran variable name লিখার নিয়ম বহির্ভূত।

Atitude : KJ-26 : গ্রহণযোগ্য নয়। কারণ variable name এর দৈর্ঘ্য Fortran–77 এ সর্বোচ্চ ছয় (৬) ক্যারেক্টার হতে পারবে, কিন্তু Atitude variable টির দৈর্ঘ্য ৬ এর বেশী। ইহা Fortran–90/95 এর ক্ষেত্রে শুদ্ধ।

LXYX∗Z : গ্রহণযোগ্য নয়। কারণ Fortran variable নাম লিখার জন্য asterisk '∗' চিহ্ন ব্যবহারের অনুমতি নাই।]

A58.2 : গ্রহণযোগ্য নয়। কারণ ডেসিম্যাল পয়েন্ট ব্যবহার করা হয়েছে, যেটি ফরট্রান চলক লিখার নিয়ম বহির্ভূত।

25 MAY : গ্রহণযোগ্য নয়। কারণ প্রথম Character অবশ্যই Letter (A to Z or a to z) হতে হবে।

IN-OUT : গ্রহণযোগ্য নয়। কারণ এতে Special Character হাইপেন '–' ব্যবহার করা হয়েছে, যেটি fortran variable name লিখার নিয়ম বহির্ভূত।

XN6 : গ্রহণযোগ্য

(X61) : গ্রহণযোগ্য নয়। কারণ parenthesis '( )' বন্ধনী ব্যবহার করা হয়েছে, যেটি Fortran variable name লিখার নিয়ম বহির্ভূত।

2.4. ফরট্রান Statement লেখার নিয়ম বা কার্ড পাঞ্চিং সিস্টেম (Rules for writting fortran statement or Card Punching System)

বর্তমানের বিভিন্ন ইনপুট যন্ত্র যেমন কিবোর্ড, মাউস বা পেনড্রাইভ, ফ্লপি ডিস্ক এর পূর্বে Punch Card ব্যবহার করে বিভিন্ন ইনপুট দেয়া হত। যাতে 80 কলামে ডেটা থাকত। আবার বর্তমানে মনিটরে ফিক্সড ফরম্যাট সিস্টেমে 80 কলামের ভিতর ডেটা ইনপুট দেয়া হয়, যার কলামগুলোর কাজ নিম্নরূপ–

1 (কমেন্ট ফিল্ড)	2345 (লেবেল ফিল্ড)	6 (কন্টিনিউশন ফিল্ড)	789···72 (স্টেটমেন্ট ফিল্ড)	73 – 80 (ইগনোরেশন ফিল্ড)
মন্তব্য শুরুর জন্য	লাইনের বামপাশে সংখ্যা লিখে লেবেলিং করার জন্য।	কোনো লাইনের কিছু অংশ পরের লাইনে গেল অবিচ্ছিন্নতা রক্ষার জন্য।	মূল প্রোগ্রামের কোড বা লাইনসমূহ লেখার জন্য।	মন্তব্য, ব্যাখ্যা, তথ্য ইত্যাদি মন্তব্য আকারে লেখার জন্য।

1. প্রথম কলামে C, c, ✳ থাকলে কম্পাইলার এই লাইনটি অনুবাদ করবে না। অর্থাৎ প্রোগ্রাম লেখক নিজের মত কোনো মন্তব্য লিখতে পারে। তাই 1 নং কলামকে মন্তব্য (Comment) ফিল্ড বলে।
2. 1 – 5 কলামে স্টেটমেন্টের লেবেল নম্বর লেখার কাজে ব্যবহার করা হয়। লেবেলটি 1 থেকে 99999 এর মধ্যে যে কোনো চিহ্নহীন পূর্ণসংখ্যা হবে। তাই 1 – 5 নং কলামকে Label ফিল্ড বলে।
3. যদি ফরট্রান স্টেটমেন্ট এক লাইনের (7 – 72 কলাম) চেয়ে বেশী লম্বা হয় অথবা স্টেটমেন্টের কিছু ক্যারেকটার পরবর্তী লাইনে লিখতে হয় তবে স্টেটমেন্টটির অবশিষ্ট অংশ পরবর্তী লাইনে লিখবার পূর্বে 6 নম্বর কলামে শূন্য (0), ফাকা (space) বা <tab> ব্যতীত যেকোনো একটি ফরট্রান ক্যারেকটার লিখতে হবে। তাই 6 নম্বর কলামকে Continuation ফিল্ড বলে।
4 7 – 72 কলামের মধ্যে ফরট্রান স্টেটমেন্ট লিখতে হয়। এখানে 67টি ক্যারেকটার আছে। তাই 7 — 72 কলামকে Statement ফিল্ড বলে।
5. ফরট্রান কম্পাইলার 73 – 80 কলামগুলোকে অনুবাদ করে না। তাই এই কলামগুলিকে প্রোগাম আইডেনটিফিকেশন মার্ক লিখতে ব্যবহার করা যায়। এই 73 – 80 কলামকে Ignoration ফিল্ড বলে।

2.5. ফরট্রান প্রোগ্রামের গঠন (Structure of Frotran Program) **[NUH(AC)–2017]**

একটি আদর্শ ফরট্রান প্রোগ্রামের গঠন নিম্নরূপ ঃ

PROGAM name

Opening documentation

Variable declarations

Program statements ad comments

END