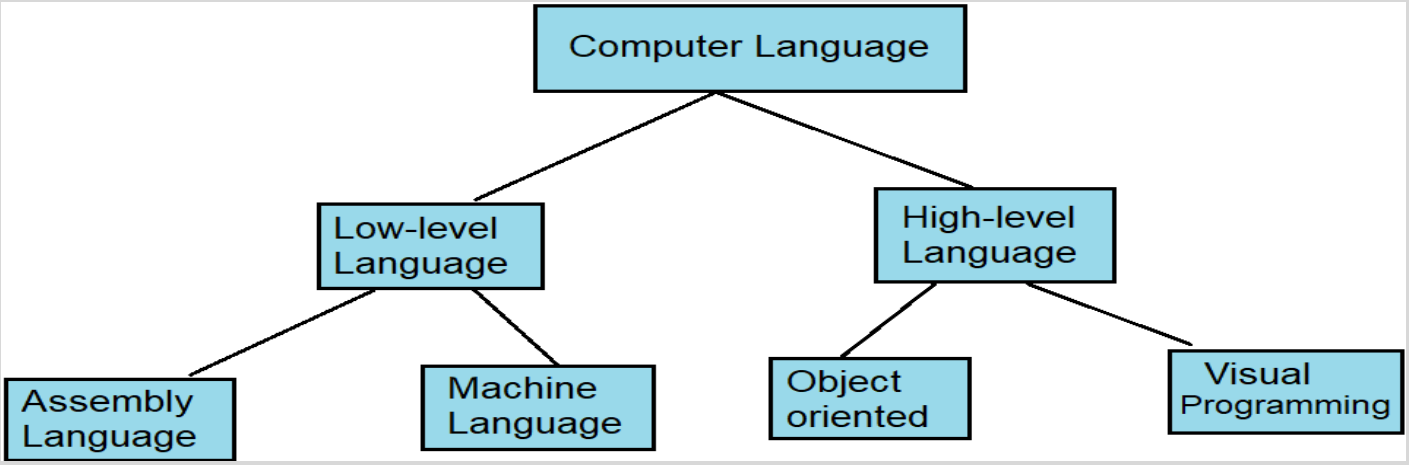
Programming Language – कंप्यूटर लैंग्वेज क्या है?

- Computer language एक ऐसी लैंग्वेज होती है जिसके द्वारा कंप्यूटरों के साथ communicate किया जाता है।
- जिस तरह हम इंसान आपस में एक दुसरे से communicate (बात-चीत) करने के लिए किसी ना किसी language (भाषा) का इस्तेमाल करते है। ठीक उसी तरह computer भी communicate करने के लिए language का use करते है। **जिसे हम computer language कहते है।**
- कंप्यूटर लैंग्वेज instructions (निर्देशों) का एक समूह होता है जिसके द्वारा किसी विशेष कार्य को पूरा किया जाता है।
- दूसरें शब्दों में कहें तो, "Computer language एक code होता है जिसके द्वारा programs को लिखा जाता है।"
- कंप्यूटर लैंग्वेज का प्रयोग desktop applications, mobile applications और websites को बनाने के लिए किया जाता है। जैसे कि- Java, C++ का प्रयोग करके app बनाई जाती है।
- Computer language को programming language भी कहा जाता है और इसका प्रयोग programmers के द्वारा कंप्यूटर प्रोग्राम को create करने के लिए किया जाता है।
- कुछ कंप्यूटर लैंग्वेज इंसान की समझ में नही आती हैं इन्हें केवल कंप्यूटर ही समझ सकता है।

Types of computer language – कंप्यूटर लैंग्वेज के प्रकार

Computer language मुख्य रूप से दो प्रकार की होती हैं जो कि निम्नलिखित है।

1. High Level Language
2. Low Level Language



Low Level Language – लो लेवल लैंग्वेज क्या है?

- Low Level Language एक ऐसी भाषा होती है जिसे हम इंसान नही समझ सकते इसे केवल कंप्यूटर के द्वारा ही समझा जा सकता है।
- Computer इस language को बड़ी ही आसानी से समझ सकते है। ये language high level language के बिलकुल विपरीत है।
- यह भाषा machine dependent होती है इसका मतलब यह कि यह भाषा कुछ ही कंप्यूटर पर run होती है।
- इस लैंग्वेज में प्रोग्राम को run करने के लिए compiler और interpreter की जरूरत नही पड़ती है।
- इस language का program काफी तेज execute होता है। इसके अलावा output भी काफी जल्दी देता है।
- यह लैंग्वेज high level language के मुकाबले काफी ज्यादा मुश्किल होती है। अर्थात इसे सीखना मुश्किल होता है।

Types of Low level language – लो लेवल लैंग्वेज के प्रकार

1. **Machine Language**
2. **Assembly Language**

Machine Language क्या है ?

Machine language वह भाषा होती है जिसमें केवल binary (0 और 1) अंको का ही प्रयोग होता है।

दूसरे शब्दों में कहें तो, "जिस भाषा को computer बिना किसी technology के समझ लेता है। उसे हम machine language कहते है।"

computer केवल Binary (0 और1) को ही समझ पाता है। Binary digit का use computer हर काम के लिए करता है। जो program 0 से लेकर 1 digit के होते है। उन्हें हम machine language program कहते है।

Advantages of Machine Language – मशीन लैंग्वेज के फायदे

1:- इस भाषा में लिखे गये programs को computer आसानी से समझ लेता है।

2:- इस language को computer जल्दी execute कर लेते है। क्योंकि इसमें compiler और इंटरप्रेटर की जरूरत नही पड़ती।

3:- यह output जल्दी दे देता है।

Disadvantages of Machine Language – मशीन लैंग्वेज के नुकसान

1:- मशीनी भाषा के लिखे गये program किसी इंसान को समझ नहीं आते।

2:- machine भाषा के लिखे गए program लम्बे होते है।

3:- program लिखते समय गलतियां ज्यादा होती है।

Assembly Language क्या है ?

Assembly language एक low-level प्रोग्रामिंग लैंग्वेज है. इसको मशीन लैंग्वेज में बदलने के लिए एक सॉफ्टवेर की आवश्यकता होती है जिसे assembler कहते है.

असेंबली लैंग्वेज का प्रयोग माइक्रोप्रोसेसर पर आधारित डिवाइस में, और real time systems में किया जाता है.

इस language में 0 और 1 digit के स्थान पर alphanumeric का use किया जाता है। जैसे कि- A-Z, 0-9 .

Assembly भाषा में लिखे गए program किसी दुसरे computer पर execute नहीं हो सकते है। इस भाषा को लिखने और समझने के लिए computer hardware की knowledge होनी चाहिए।

असेंबली भाषा के फायदे

1. assembly भाषा को समझना machine language की तुलना में काफी ज्यादा आसान है।
2. इस भाषा में गलती होने के chances काफी कम होते है।
3. इसमें program को modify करना आसान है।

असेंबली भाषा के नुकसान

1. assembly भाषा के प्रोग्राम machine पर depend होते है।
2. इस भाषा का use करने के लिए hardware की knowledge ज़रूरी है।
3. इस भाषा में काफी ज्यादा time waste होता है।

High Level Language– हाई लेवल लैंग्वेज क्या है?

- High Level Language एक ऐसी भाषा है , जिसकी मदद से कंप्यूटर प्रोग्राम को आसानी से समझा और लिखा जा सकता है।
- High level language अंग्रेजी की तरह होती है इसलिए इस भाषा को इंसान आसानी से समझ सकते हैं।
- हाई लेवल लैंग्वेज का इस्तेमाल user-friendly सॉफ्टवेयर और वेबसाइट बनाने में किया जाता है।
- ये एक ऐसी भाषा है जिसका syntax पहले से ही निर्धारित होता है।
- हाई लेवल लैंग्वेज में compiler और interpreter की आवश्यकता होती है जिससे कि program को machine language में बदला जा सके।
- High-level language के उदाहरण हैं – Python, Java, JavaScript, PHP, C#, C++, Cobol, Perl, Pascal, और FORTRAN आदि।

Types of High level language– हाई लेवल लैंग्वेज के प्रकार

अलग अलग जरूरतों को पूरा करने के लिए अलग अलग प्रकार के high level language होते है। चलिए उनको भी एक बार देख लेते है।

1:- Object-Oriented Programming Language – इस language में दुनिया की समस्याओ को observe किया जाता है। उसके बाद इन समस्याओ को solve किया जाता है। example के लिए C++ , Java.

2:- Visual Programming Language – इस language का इस्तेमाल window application को बनाने के लिए और design करने के लिए किया जाता है। example के लिए Visual Basic, Visual Java, Visual C.

Compiler – कम्पाइलर क्या है ?

- Compiler एक ऐसा प्रोग्राम होता है जो high level language में लिखे गए code को low level language में बदलता है जिससे कि प्रोग्राम को आसानी से execute किया जा सके।
- दूसरे शब्दों में कहें तो, "कम्पाइलर एक language translator होता है जो high level language के code को machine language में translate करता है।"
- Compiler पूरे हाई लेवल लैंग्वेज के कोड को एक ही बार में मशीन लैंग्वेज में बदल देता है। जबकि interpreter एक-एक line को मशीन लैंग्वेज में बदलता है।
- Compiler के द्वारा source code को object code में convert (बदला) जाता है। अगर source code में कोई error (गलती) होती है तो कम्पाइलर source code को object code में नहीं बदल सकता।
- कम्पाइलर, Interpreter से ज्यादा बुद्धिमान (intelligent) होता है क्योंकि यह एक ही बार में पूरे source code को translate कर देता है।
- प्रोग्रामिंग लैंग्वेज जैसे कि – C++, JAVA, C, C# आदि कम्पाइलर का प्रयोग अपने programs को translate करने के लिए करती है।

Advantage of compiler– कम्पाइलर के फायदे

कम्पाइलर के फायदे निम्नलिखित हैं-

1. एक समय में कम्पाइलर पूरे प्रोग्राम या source code को एक साथ ही execute कर सकता है।
2. compile किया हुआ प्रोग्राम बहुत तेजी से run होता है।
3. compiler में गलतियां होने की संभावना कम होती है।
4. कम्पाइलर program को optimize करता है इसलिए code को कम मैमोरी की जरूरत पड़ती है।
5. कम्पाइलर programmer को error दिखाता है।
6. इसमें programs सुरक्षित रहते हैं और कोई हैकर इन programs को हैक नही कर सकता।
7. इसमें CPU का utilization ज्यादा होता है।
8. इसे बहुत सारीं high-level language के द्वारा support किया जाता है जैसे कि- C++, JAVA, C# आदि।

Disadvantages of compiler – कम्पाइलर के नुकसान

Compiler के नुकसान नीचे दिए गए हैं-

1. कम्पाइलर flexible नही होता है।
2. इसमें प्रोग्राम को debug करना मुश्किल होता है।
3. इसमें errors को ढूढना मुश्किल होता है।
4. अगर प्रोग्राम में कोई बदलाव होता है तो पुरे प्रोग्राम को फिर से compile करना पड़ता है |

Types of compiler– कम्पाइलर के प्रकार

इसके दो प्रकार होते है-

1. One Pass compiler – यह कम्पाइलर केवल एक बार ही code को read करता है और उसे ट्रांसलेट कर देता है।
2. Multi pass compiler – यह कम्पाइलर code को बहुत बार read और translate करता है।

Interpreter– इंटरप्रेटर क्या है?

- Interpreter एक कंप्यूटर प्रोग्राम होता है जो high-level लैंग्वेज में लिखे गए code को machine लैंग्वेज में बदल देता है।
- यह code को एक-एक लाइन करके मशीन लैंग्वेज में बदलता है। अगर किसी लाइन में कोई error आती है तो जब तक उस error को ठीक नही कर लिया जाता है तब तक यह आगे कोड को ट्रांसलेट नही करता है।
- Interpreter कोड की प्रत्येक लाइन को ध्यान से check करता है। लाइन के सही होने पर वह उसे सीधे मशीन लैंग्वेज में बदल देता है।
- इंटरप्रेटर का सबसे पहले प्रयोग 1952 में किया गया था।
- High level language को केवल इंसानो के द्वारा ही समझा जा सकता है। जिन्हे हम source code भी कहते है। दूसरी और देखे तो computer केवल binary भाषा में लिखे हुए program को ही समझ पाता है। इसलिए interpreter और compiler की ज़रूरत पड़ती है।

Interpreter कैसे काम करता है ?

किसी भी program को लिखने से पहले Interpreter में memory को load कर दिया जाता है। इसके बाद हम program लिखने की शुरुआत करते है। जैसे जैसे हम high level language में code लिखने लगते है। वैसे वैसे interpreter उस कोड को check करता है।

अगर लिखते समय आपसे कोई गलती हो गई हो। तो interpreter उस line को bold कर देता है। ताकि आप उस कोड में सुधार कर सके।

जब कोई mistake नहीं होती तब interpreter उसको मशीन लैंग्वेज में convert कर देता है।

Advantages of Interpreter (इंटरप्रेटर के फायदे)

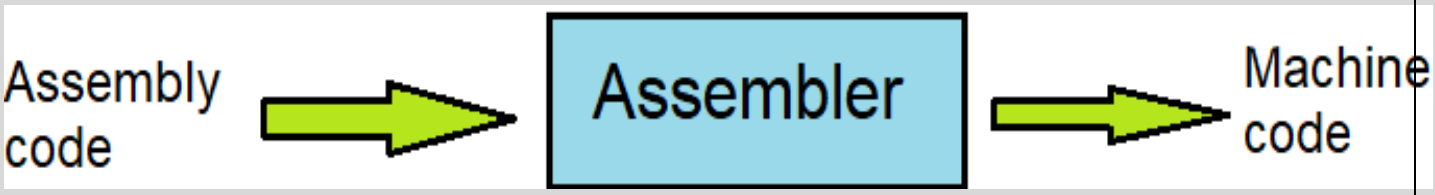
1. इसके द्वारा कोड को debug करना काफी ज्यादा आसान है। क्योकि यह कोड को line by line पढ़ता है।
2. Line by line check करने के कारण गलतियों को ढूढ़ना काफी ज्यादा आसान हो जाता है।
3. यह compiler की तरह अलग file नहीं बनाते। जिसके कारण memory का use बहुत कम करना पड़ता है।
4. Interpreter को आप कभी भी रोक कर code को सही कर सकते है।
5. इसमें mistake होने की संभावना बहुत कम होती है।

Disadvantages of Interpreter – इंटरप्रेटर के नुकसान

1. यह compiler की तुलना में काफी ज्यादा slow काम करता है।
2. इसकी security बहुत ही कमजोर होती है।
3. कंप्यूटर में source code को execute करने के लिए हमें Interpreter को install करना पड़ता है।

Assembler – असेम्बलर क्या है?

- Assembler एक ऐसा computer program है जो assembly भाषा में लिखे गए code को machine भाषा में convert करने में मदद करता है।
- दूसरे शब्दों में कहें तो, "असेम्बलर एक प्रोग्राम है जो असेंबली भाषा को मशीन भाषा में बदल देता है।"
- असेम्बलर को कभी-कभी assembly language का compiler भी कहते है।



Types of Assembler – असेम्बलर के प्रकार

1:- **One-Pass Assembler**– यह एक प्रकार का Load-and-go assembler है। जो एक ही बार में assembly code को machine code में convert कर देता है। इसमें बार बार काम करने की ज़रूरत नहीं पड़ती।

2:- Multi-Pass/Two-Pass Assembler – इसमें बहुत बार असेंबली कोड को मशीन कोड में बदला जाता है। इसमें बार बार काम करने की जरूरत होती है।

Advantages of Assembler – असेम्बलर के फायदे

1. इसके द्वारा कठिन काम को आसन तरीके से पूरा कर सकते है।
2. इसमें कम memory space की ज़रूरत पड़ती है।
3. इसके काम करने की speed तेज है। क्योकि इसमें execution time कम है।
4. इसका इस्तेमाल important काम को समय में पूरा करने के लिए किया जाता है।
5. इसमें Memory location को track करने की ज़रूरत नहीं पड़ती।
6. यह flexible होता है।

Disadvantage of Assembler – असेम्बलर के नुकसान

1. इसके अंदर code लिखने में काफी ज्यादा समय और मेहनत लगती है।
2. असेम्बलर आसानी से समझ में नहीं आते। यह काफी ज्यादा complex (कठिन) होते है।
3. इसमें Syntax को याद करके रखना काफी ज्यादा मुश्किल है।
4. असेंबली भाषा के लम्बे program को run करने के लिए ज्यादा memory space का इस्तेमाल करना पड़ता है।

Difference between compiler, assembler and interpreter – कम्पाइलर, असेम्बलर और इंटरप्रेटर के बीच अंतर

Compiler	Assembler	Interpreter
यह पुरे program को scan करता है। और एक ही बार में पुरे program को machine code में convert करता है।	यह program को phases में scan करता है।	यह एक एक line करके program को scan करता है।
इसको source code को analys करने में काफी ज्यादा समय लगता है।	इसमें source code को scan करने में काफी कम समय लगता है।	इसमें भी source code को analyze करने में काफी ज्यादा समय लगता है। लेकिन process को जल्दी execute कर दिया जाता है।
इसमें memory space की ज्यादा जरूरत पड़ती है।	इसमें ज्यादा memory space की ज़रूरत नहीं पड़ती।	इसमें ज्यादा memory space की ज़रूरत नहीं पड़ती।
यह पुरे program को scan करने के बाद गलतियों को ढूढता है।	असेम्बलर phases में गलतियों को ढूढता है।	यह line by line गलतियों को ढूढता है।
इसके साथ काम करते समय debug करने में काफी ज्यादा समस्याओं का सामना करना पड़ता है।	debugging करने में परेशानियों का सामना करना पड़ता है।	इसके साथ debugging करना काफी आसान होता है।
Compiler के कुछ example है C and C++ etc .	इसके कुछ उदाहरण – GAS, GNU etc.	इसके कुछ example – Ruby and Python etc .