



இந்த அத்தியாயத்தைக் கற்பதன் மூலம்

- கணினி மொழிகளின் வளர்ச்சி
- கணினியின் உதவியுடன் பிரசினங்களைத் தீர்த்தல்
- நெறிமுறையை வகைகுறித்தல்
- விஷுவல் பேசிக்கைப் பயன்படுத்தி மூலமாக செய்நிரல்களைத் தயாரித்தல்

ஆகியன பற்றிய விளக்கத்தைப் பெற்றுக் கொள்ளலாம்.

தற்போது, கணினி ஓர் இன்றியமையாத சாதனமாக விளங்குகிறது. கணினியைப் பயன்படுத்தி அனேக கருமங்களை ஆற்றக்கூடியதாக உள்ளது. பிரசினங்களைத் தீர்ப்பதற்கான கருவியாக கணினியை எவ்வாறு பயன்படுத்தலா மென இந்த அத்தியாயத்தில் ஆராயவுள்ளோம்.

கணினியின் துணையுடன் பிரசினமொன்றைத் தீர்ப்பதற்கு, கணினியால் விளங்கிக் கொள்ளத்தக்க மொழியில் அந்தப் பிரசினத்தைத் தீர்க்கும் முறையை கட்டளைகளின் தொகுப்பாக கணினிக்கு வழங்க வேண்டும். கணினியால் நேரடியாக விளங்கிக் கொள்ளத்தக்க மொழிகள் இயந்திர மொழிகள் (Machine Languages) எனப்படும். பொதுவாக இயந்திர மொழிகளில் அடங்கியுள்ள கட்டளைகள் மற்றும் அக்கட்டளைகளின் தன்மை ஆகியன கணினி முறைவழியாக்கியின் (Processor) வகையில் தங்கியுள்ளது. அதாவது, பல்வேறு வகைப்பட்ட முறைவழியாக்கிகளினால் விளங்கிக் கொள்ளத்தக்க இயந்திர மொழிகளின் தன்மை வேறுபட்டதாகும்.

உதாரணமாக 'Intel' குடும்ப முறைவழியாக்கிகளிற்கெனத் தயாரிக்கப்பட்ட இயந்திர மொழியானது 'SPARC' குடும்ப முறைவழியாக்கிகளின் இயந்திர மொழியைவிட மாறுபட்டதாகும். எனினும், பொதுவான ஒற்றுமை கொண்ட இயந்திர மொழிகளைப் பயன்படுத்திச் செயற்படுத்தத்தக்கதாகத் தயாரிக்கப்பட்ட பல்வேறு முறைவழியாக்கிகள் தற்போது சந்தையில் உள்ளன. 'Pentium', 'AMD' ஆகிய முறைவழியாக்கிகளை இதற்கு உதாரணமாகக் குறிப்பிடலாம்.

பல்வேறு முறைவழியாக்கிகளுக்கென வெவ்வேறு இயந்திர மொழிகள் அறிமுகஞ் செய்யப்பட்டுள்ளபோதும் எல்லா இயந்திர மொழிகளுக்கும் பொதுவான இயல்பொன்று உள்ளது. எல்லா இயந்திர மொழிகளுக்கும்மான கட்டளைகள் 0 அல்லது 1 எனும் குறியீடுகளை மட்டுமே அடிப்படையாகக் கொண்டிருப்பதே அந்தப் பொது இயல்பாகும். கணினி, இரண்டு குறிகளை மட்டும் கொண்ட இரும் மொழியை (Binary Language) மட்டுமே விளங்கிக்கொள்ளக் கூடியதாக இருப்பதே அதற்கான காரணமாகும். இவ்வாறான இரும் குறிமுறையை மட்டும் கொண்ட கணினி மொழிகள் கீழ்மட்ட மொழிகள் (Low Level Languages) என அழைக்கப்படும்.

அன்றாடம் கணினிகளைச் செயற்படுத்துவதற்கென இந்த கீழ்மட்ட மொழிகளைப் பயன்படுத்துவது சிக்கலானதாக இருக்கும். எனினும், இம்மொழிகள் கணினிகளினால் நேரடியாக விளங்கிக் கொள்ளத்தக்க மொழி வகைகளாக அமைந்துள்ளமை சிறப்பம்சமாகும். இதன் காரணமாகவே இயந்திரமொழி என அழைக்கப்படும் இந்த கீழ்மட்ட மொழி கணினிக்கு மிக அண்மையானதெனவும் மனிதனுக்கு மிக சேய்மையானதெனவும் கருதப்படுகிறது.

இயந்திர மொழியைப் பயன்படுத்தி கட்டளைகளை வழங்குவது மிக கடினமாக அமைந்துள்ளதால் இதற்குப் பதிலாக ஆங்கில மொழிக்கு நெருக்கமான பல்வேறு கணினி மொழிகள் உருவாக்கப்பட்டுள்ளன. இவை உயர்மட்ட மொழிகள் என அழைக்கப்படுகின்றன. 'C','VB', 'PHP', Java ஆகியன இவற்றுக்கான சில உதாரணங்களாகும்.

கணினிமூலமான தொழிற்பாட்டை மேற்கொள்வதற்கென, கணினி மொழியின் துணையுடன் நிரற்படுத்தப்பட்டுள்ள கூற்றுகளின் தொகுதி கணினிச் செய்நிரல் (Computer Programme) எனப்படும். இக்கூற்றுகளில் பெரும்பாலானவை கணணிக்கான கட்டளைகளாகவும் ஏனையவை விளக்கக் குறிப்புக்களாகவும் இருக்கும் இவ்வாறான கட்டளைகள் அல்லாதவை செயல் நிரல்களை இலகுவாக விளங்கிக் கொள்ளத் தேவையான குறிப்புரைகளாகவோ (comments) செயல்நிரல்களை இயந்திர மொழிக்கு மாற்றுவதற்கு தேவையான தகவல்களாகவோ இருக்கலாம். இவ்வாறான கட்டளைகள் அல்லாத கூற்றுகள் பற்றிய அறிமுகம் பின்னர் வழங்கப்படும்.

'C', 'VB', 'PHP', Java போன்ற கணினி மொழிகளை எம்மால் இலகுவாக விளங்கிக்கொள்ளக் கூடியதாக உள்ளபோதும் இவற்றைக் கணினியால் இலகுவாக விளங்கிக் கொள்ள முடியாது. ஆகையால், இவ்வாறான உயர்மட்ட மொழி களைப் பயன்படுத்தி தயாரிக்கப்பட்ட செய்நிரல்கள் கணினியால் செயற்படச் செய்ய முன்பதாக கணினியால் விளங்கிக் கொள்ளக் கூடிய கீழ்மட்ட மொழிகளாக மொழிமாற்றம் செய்யப்பட வேண்டும். இச்செயன்முறை மொழிபெயர்ப்பு (Language translation), அல்லது தொகுப்பாக்கம் (compilative) எனப்படும். இதற்கெனவுள்ள பல்வேறு முறைகள் பற்றிப் பின்னர் கற்போம்.



## 1.1 கணினி மொழிகளின் வளர்ச்சி

ஆரம்ப காலந்தொட்டு இன்றுவரை உருவாகியுள்ள கணினி மொழிகளை அவற்றின் கட்டளைகளின் தன்மைக்கேற்ப நான்கு பிரதான தொகுதிகளாக வகைப்படுத்தலாம்.

1. முதலாம் சந்ததிக் கணினி மொழிகள் அல்லது இயந்திர மொழிகள் (Machine Languages)
2. இரண்டாம் சந்ததிக் கணினி மொழிகள் அல்லது ஒருங்கு சேர்ப்பு மொழிகள் (Assembly Languages)
3. மூன்றாம் சந்ததிக் கணினி மொழிகள் அல்லது உயர்மட்ட மொழிகள் (High Level Languages)
4. நான்காம் சந்ததிக் கணினி மொழிகள்

கணினி மொழிகள் எம்மால் இலகுவாக விளங்கிக் கொள்ளக் கூடியவாறும் இலகுவாகப் பயன்படுத்தக் கூடியவாறும் படிப்படியாக விருத்தியடைந்தன.

### முதலாம் சந்ததிக் கணினி மொழிகள்

எல்லா இயந்திர மொழிகளும் இதிலடங்கும்.

**இவ்வகை மொழிகளின் பொது இயல்புகள்:**

- ❖ எல்லாக் கட்டளைகளும் இரும் குறியீட்டு முறைமூலம் உள்ளிடப்பட வேண்டும்.
- ❖ செய்நிரலாக்கக் கட்டளைகளை கணினியில் வழங்குதல் மிகச் சிரமமானதாகும்.
- ❖ இது இயந்திரத்தின் வகையில் தங்கியுள்ள மொழியாகும். அதாவது ஒரு வகை இயந்திர முறைவழியாக்கியிற்காக தயாரிக்கப்பட்ட செய்நிரலை வேறு முறைவழியாக்கிகளைப் பயன்படுத்தி செயற்படச் செய்யமுடியாது.
- ❖ கணினியின் பல்வேறு வன்பொருட்கள் பற்றிய அறிவு அவசியமாகும்.

## இரண்டாம் சந்ததிக் கணினி மொழிகள்

0, 1 ஆகிய இரும குறியீடுகளை மட்டும் பயன்படுத்தி செய்நிரலை தயாரிப்பதிலுள்ள சிக்கல்களை நீக்குவதற்கென உருவாக்கப்பட்ட கணினி மொழி வகைகளே இவையாகும்.

**இவ்வகை மொழிகளின் பொது இயல்புகள்:**

- ❖ கட்டளைகள் 'Add', 'Sub' போன்ற எளிய குறிப்பீட்டுப் பெயர்களில் இருத்தல்.
- ❖ எளிய குறிப்பீட்டுப் பெயர்களைப் பயன்படுத்தி கணினி நினைவகத்தில் தரவுகளைச் சேமிக்கவும் மீண்டும் பெற்றுக் கொள்ளவும் இயலுமானதாக விருத்தல்.
- ❖ செய்நிரல்களை தவறு நீக்குதல் (Debugging) இயந்திர மொழியை விட இலகுவானதாகவிருத்தல்.
- ❖ இயந்திரத்தின் வகையில் தங்கியிருக்கும் மொழியாகக் காணப்படல்
- ❖ செய்நிரல்களை எழுதுவதற்கு வன்பொருட்கள் பற்றிய ஆழமான அறிவு தேவைப்படல்.

இம்முறைகளைப் பயன்படுத்தி உருவாக்கப்பட்ட செய்நிரல்களை கணினிகளினால் நேரடியாகத் செயற்படச் செய்ய முடியாது. அதற்கென ஒருங்கு சேர்ப்பி (Assembler) எனப்படும் மென்பொருள் முறை மூலம் அந்த செய் நிரல்கள் அல்லது கட்டளைகள் உரிய இயந்திரமொழிக்கு மாற்றம் செய்யப்பட வேண்டும்.

## மூன்றாம் சந்ததிக் கணினி மொழிகள்

தற்சமயம் பயன்பாட்டிலுள்ள C, VB ஆகிய மொழிகள் இவ் வகையில் அடங்கும்.

**இவ்வகை மொழிகளின் பொது இயல்புகள்**

- ❖ ஆங்கிலம் போன்று எமக்குப் பரிட்சயமான மொழிகளில் அமைந்திருத்தல்.
- ❖ இயந்திரத்தின் வகையில் தங்கியிருக்கும் மொழியாக அமையாமை.
- ❖ செய்நிரலாக்கம் மற்றும் தவறு திருத்தல் ஆகியன இலகுவாக அமைந்திருத்தல்.

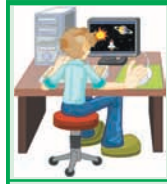
இம்மொழிகளையும் கணினியால் நேரடியாக புரிந்து கொள்ள முடியாது. ஆகவே, மொழிபெயர்ப்பு மென்பொருள்கள் மூலம் இம்மொழியில் எழுதப்பட்ட செய்நிரல்கள் இயந்திரமொழிக்கு மொழிபெயர்ப்புச் செய்யப்படும்.

## நான்காம் சந்ததிக் கணினி மொழிகள்

செய்நிரலாக்கத்திற்கு எடுக்கும் காலத்தைக் குறைக்கும் நோக்கில் தயாரிக்கப்பட்ட கணினி மொழிகளே இவையாகும்.

**இவ்வகை மொழிகளின் பொது இயல்புகள்:**

- ❖ ஆங்கிலம் போன்ற எம்மால் பயன்படுத்தும் மொழிக்கு நெருக்கமான தன்மையைக் கொண்டிருத்தல்.
- ❖ மக்களால் குறுகிய காலத்தில் கற்கக்கூடியதாகவிருத்தல்.
- ❖ மிகக் குறைந்த எண்ணிக்கையான கட்டளைகளின் மூலமாக கருமங்களை மேற்கொள்ளச் செய்யக் கூடியதாயிருத்தல்.



### செயற்பாடு 1.1

- (1) இயந்திர மொழி, ஒருங்கு சேர்ப்பி மொழி ஆகியவற்றுக் கிடையிலான வேறுபாடுகளை இனங்காண்க.
- (2) ஒருங்குசேர்ப்பி மொழி, உயர்மட்ட மொழி ஆகியவற்றுக் கிடையிலான அடிப்படை வேறுபாடுகளை விளக்குக.



### செய்நிரலாக்க மொழிகளை மொழிபெயர்த்தல் (Programming Language Translation)

கணினிகளை இரும் குறியீட்டு முறையைக் கொண்ட இயந்திரமொழி மூலமாக மட்டுமே நேரடியாகத் தொழிற்படச் செய்ய முடியுமென நாம் கற்றுள்ளோம். ஆகவே, இயந்திரமொழி தவிர்ந்த இரண்டாம் மற்றும் மூன்றாம் நான்காம் சந்ததிக்குரிய மொழிகளில் எழுதப்பட்ட செய்நிரல்களை அல்லது கட்டளைகளை கணினிகளில் செயற்படுத்த முன்னர் கணினியினால் விளங்கிக் கொள்ளத்தக்க இயந்திர மொழியாக கணினி மென்பொருட்களின் மூலமாக மாற்றம் செய்யப்பட வேண்டும். இவ்வாறு ஒரு கணினி மொழியை மற்றுமோர் கணினிமொழியாக மாற்றுதல் கணினி மொழிபெயர்ப்பு (Translation) எனப்படும். இதற்கான மூன்று முறைகள் உள்ளன. அவையாவன :

- (1) ஒருங்குசேர்ப்பியைப் (Assemblers) பயன்படுத்தல்
- (2) தொகுப்பிகளைப் (Compilers) பயன்படுத்தல்
- (3) வரிமொழி மாற்றிகளைப் (பொருள் கோடலிகளைப்)(Interpreters) பயன்படுத்தல்

## ஒருங்குசேர்ப்பி (Assembler)

ஒருங்குசேர்ப்பு மொழிமூலமாக தயார்படுத்தப்பட்ட கணினிச் செய்நிரலை கணினியால் விளங்கிக் கொள்ளத்தக்கதாக இயந்திரமொழிக்கு (Machine Language) மாற்றும் செய்நிரலை இதுவாகும்.

## தொகுப்பி (Compiler)

தொகுப்பி மென்பொருள்கள் மூலமாக செய்நிரலில் உள்ள எல்லா கட்டளைகளும் இயந்திர மொழியாக மாற்றப்படும். இவ்வாறு பெறப்படும் செய்நிரல் பல்வேறு உள்ளீடு தரவுகளுக்கென மீண்டும் மீண்டும் செயற்படுத்த முடியும். C, FORTRAN ஆகிய கணினி மொழிகளின் தொகுப்பிகள் இதற்கு உதாரணங்களாகும்.

தொகுப்பிகள் மூலம் மொழிபெயர்ப்புச் செய்யப்படும்போது பயன்படுத்தப்படும் ஆரம்ப செய்நிரல்கள் ஆதார அல்லது மூல செய்நிரல் (Source Code Programme) எனப்படும். அவ்வாறே மொழிபெயர்ப்பு மூலம் கிடைக்கும் செய்நிரல், இலக்குப் பொருள் செய்நிரல் (Object Code Programme) எனப்படும். மூல செய்நிரல்களை செய்நிரல்களாக மாற்றுவதே மொழித்தொகுப்பிகளின் பணியாகும். ஆகவே, செய்நிரலிலுள்ள எல்லாத் தவறுகளையும் இனங்காணல். மற்றும் அவற்றைத் திருத்துதல் ஆகியன இச்செயல்முறையின் போது மேற்கொள்ளப்பட வேண்டும். செயல்நிரல்களில் உள்ள தவறுகளை இனங்காணல், மற்றும் அத்தவறுகளை திருத்துதல் ஆகியன தவறுநீக்கல் (Debugging) எனப்படும். செய்நிரலில் உள்ள பிழைகள் தவறுகள் 'Bugs' என அழைக்கப்படும்.



## வரிமொழிமாற்றி (பொருள் கோடலி) (Interpreter)

வரிமொழிமாற்றி மென்பொருட்கள் மூலமாக ஒரே தடவையில் முழுமையான செய்நிரலும் இயந்திர மொழியாக மொழி பெயர்க்கப்படுவதில்லை. இங்கு செய்நிரல் செயற்படுத்தப்படும்போது அதில் அடங்கியுள்ள கட்டளைகள் ஒவ்வொன்றும் அதற்குரிய இயந்திரமொழி கட்டளைத் தொகுப்புகளாக மொழிபெயர்ப்புச் செய்யப்பட்டு செயற்படுத்தப்படும். இவ்வகை செயல்நிரல்களைச் செயற்படுத்தும் எல்லா சந்தர்ப்பங்களிலும் இந்த மொழிப்பெயர்ப்புச் செயல்முறை மீண்டும் மீண்டும் நடைபெறும். ஆகவே, இவ்வகைச் செய்நிரல்களைச்

செயற்படுத்துவதற்கு அதிக நேரம் தேவையாகும். VB, Python ஆகியன இவ்வகைக்குரிய கணினி மொழிகளாகும். தொகுப்பிகள், பொருள்கோடலிகள் ஆகியன வற்றை இணைத்தும் மொழிப் பெயர்ப்புக் காசு பயன்படுத்தப்படுகின்றன. இங்கு முதலில் செய்நிரலில் உள்ள கட்டளைகள் இடைநிலைக் கணினி மொழியாக மொழிபெயர்க்கப்படும். பின்னர் இந்த இடைநிலைமொழி கொண்ட செய்நிரலானது பொருள்கோடலி மூலமாக கணினியில் செயற்படுத்தப்படும். 'JAVA' கணினி மொழி இதற்கு உதாரணமாகும்.



## 1.2 கணினியின் உதவியுடன் பிரச்சினைகளைத் தீர்க்கும் கட்டள்கள்

பிரச்சினைத் தீர்ப்பதற்கென கணினியைப் பயன்படுத்துவது தொடர்பான கட்டள்கள் பல உள்ளன.

- 1) பிரச்சினைத் தீர்ப்புக்குத் தயார்.
- 2) பிரச்சினைத் தீர்ப்பதற்குப் பொருத்தமான நெறிமுறையை (Algorithm) உருவாக்கல்.
- 3) உருவாக்கப்பட்ட நெறிமுறையை கணினி செய்நிரலாக மாற்றுதல்.
- 4) செய்நிரலில் உள்ள பிழைகளை இனங்காணலும் தவறு நீக்கலும்.
- 5) செய்நிரல்களை பொருத்தமான தரவுகளுடன் செயற்படுத்தல்.

### 1) பிரச்சினைப் பகுப்பாய்வு

பிரச்சினைகளைத் தீர்ப்பதற்கான அடிப்படைக் கூறுகளான உள்ளீடு, வருவிளைவு மற்றும் உள்ளீடுகளை வருவிளைவுகளாக மாற்றீடு செய்யத் தேவையான செயற்பாடுகளை இனங்காண்பதே பிரச்சினைப் பகுப்பாய்வின் அடிப்படை நோக்கமாகும்.

உதாரணம் : செவ்வகம் ஒன்றின் பரப்பளவைத் துணிய வேண்டுமெனக் கொள்க. இதற்கான உள்ளீடு, வருவிளைவு, செயன்முறை ஆகியன வருமாறு

உள்ளீடு : செவ்வகத்தின் நீளம், அகலம்

செயன்முறை : நீளம்  $\times$  அகலம்

வருவிளைவு : மேற்கூறப்பட்ட செயன்முறைமூலம் கிடைக்கும் பேறு



## 2) பிரசினத் தீர்வுக்கான நெறிமுறையை உருவாக்குதல்

குறித்தவொரு பிரசினத்தைத் தீர்ப்பதற்கென கைக்கொள்ளவேண்டிய செயல்முறைகள் அனைத்தையும் ஒழுங்குமுறையில் கொண்ட ஒரு வெளிப்படுத்தலே நெறிமுறை எனப்படும்.

மேலே தரப்பட்ட உதாரணத்திற்கான ஒரு பின்வரும் நெறிமுறையாக பின்வருவதை கொள்ளலாம்:

- செவ்வகத்தின் நீளத்தைக் பெறுதல்
- செவ்வகத்தின் அகலத்தைக் பெறுதல்
- பரப்பளவு = நீளம்  $\times$  அகலம்
- பரப்பளவை எடுத்துக்கூறுதல்

## 3) நெறிமுறையை செய்நிரலாக மாற்றுதல்

கணினியின் உதவியுடன் பிரசினமொன்றைத் தீர்ப்பதற்கென உருவாக்கப்பட்ட நெறிமுறை மிகச் சரியாக கணினியினால் புரிந்துகொள்ளக் கூடிய மொழியில் கணினியில் உள்ளிடப்பட வேண்டும். தற்போது இதற்கென உருவாக்கப்பட்டுள்ள கணினி மொழிகள் அதிகம் உள்ளன. இந்த மொழிகள் அனைத்தும் கணினியின் துணையுடன் பிரசினங்களைத் தீர்ப்பதற்காக உருவாக்கப்பட்ட செயற்கை மொழிகள் ஆகும். C, Python, Php, VB, FORTRAN ஆகியன இதற்கான சில உதாரணங்களாகும். நெறிமுறையை கணினி மொழி செய்நிரலாக மாற்றும் செயல்முறை குறிமுறையாக்கம் (Coding) எனப்படும். இப்பணியைச் செய்பவர் குறிமுறையாக்குநர் (Programmer / Coder) என அழைக்கப்படுவார்.

## 4) செய்நிரலைத் தவறு நீக்கல்

இங்கு செய்நிரலிலுள்ள பிழைகள் இனங்காணப்பட்டு அப்பிழைகள் தவறு நீக்கப்படும். இது தவறுநீக்கல் (Debugging) செயல்முறை எனப்படும். இதன்போது முன்னரே தயார் செய்யப்பட்ட மாதிரித் தரவிற்கு (Test Data)] செய்நிரலை செயற்படுத்தி உரிய வருவிளைவுகள் கிடைக்கின்றனவா என பரீட்சிக்கப்படும்.



## 5) செய்நிரலைச் செயற்படச் செய்தல்

இதன்போது தயாரிக்கப்பட்ட செய்நிரல் உண்மையான உள்ளீடுகளின் அடிப்படையில் செயற்படுத்தப்பட்டு வருவிளைவுகள் பெற்றுக்கொள்ளப்படும்.



### செயற்பாடு 1.2

வட்டமொன்றின் பரப்பளவைக் காண்பதற்கான செய்நிரலை உருவாக்க அவசியமான உள்ளீடு, வருவிளைவு, மற்றும் செயன்முறை ஆகியவற்றை இனங்காணுங்கள்.



### செயற்பாடு 1.3

நீங்கள் ஆண்டிறுதிப் பரீட்சையில் எல்லாப் பாடங்களிலும் பெற்ற புள்ளிகளின் மொத்தம், இடை மற்றும் இடையம் ஆகியவற்றைப் பெறுவதற்கான செய்நிரலொன்றை உருவாக்குவதற்குத் தேவைப்படும் உள்ளீடுகள் செயன்முறை மற்றும் வருவிளைவுகள் ஆகியவற்றை இனங்காணுங்கள்.


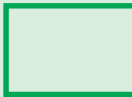

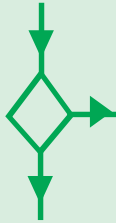

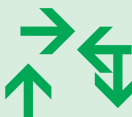


## 1.3 நெறிமுறையை வெளிப்படுத்தல்

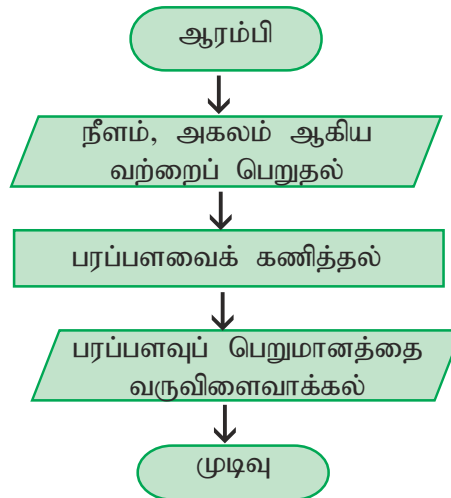
பிரசினத்தைத் தீர்ப்பதற்கான நெறிமுறையொன்றை உருவாக்கும்போது அதற்கான செயன்முறை மற்றும் அச்செயன்முறையினை செயற்படுத்தும் ஒழுங்குமுறை ஆகியவற்றை வெளிப்படுத்துவதற்கான குறித்தவொரு நுட்பமுறை அவசியமாகும். இதற்கென வரைவியலில் (Graphical) அல்லதுஎழுத்துருவில் (Textual) வெளிப்படுத்தல்களைப் பயன்படுத்தலாம். பாய்ச்சற் படங்கள் (Flow charts) இதற்காக அதிகளவில் பயன்படுத்தப்படும். இதுதலுவிர பேர்லிக்குறிமுறைகள் (Pseudo codes) மூலமாகவும் நெறிமுறை எழுத்தணதுருவில் வெளிப்படுத்தப்படும்.

## பாய்ச்சற் படங்கள் (Flow Charts)

நெறிமுறைத் தொழிற்படு விதத்தை ஒழுங்கு முறையில் வரையவியல், மற்றும் குறிமுறை ஆகியன மூலமாக வெளிப்படுத்துதலே பாய்ச்சற் படங்கள் எனப்படும். பாய்ச்சற் படங்களை உருவாக்குவதற்குரிய நியமக் குறியீடுகள் உள்ளன. இதன்போது இவ்வாறான குறியீடுகள் அதிகளவில் பயன்படுத்தப்படும். இவ்வாறான சில குறியீடுகளும் அவற்றின் பயன்பாடும் பின்வரும் அட்டவணையில் தரப்பட்டுள்ளது.

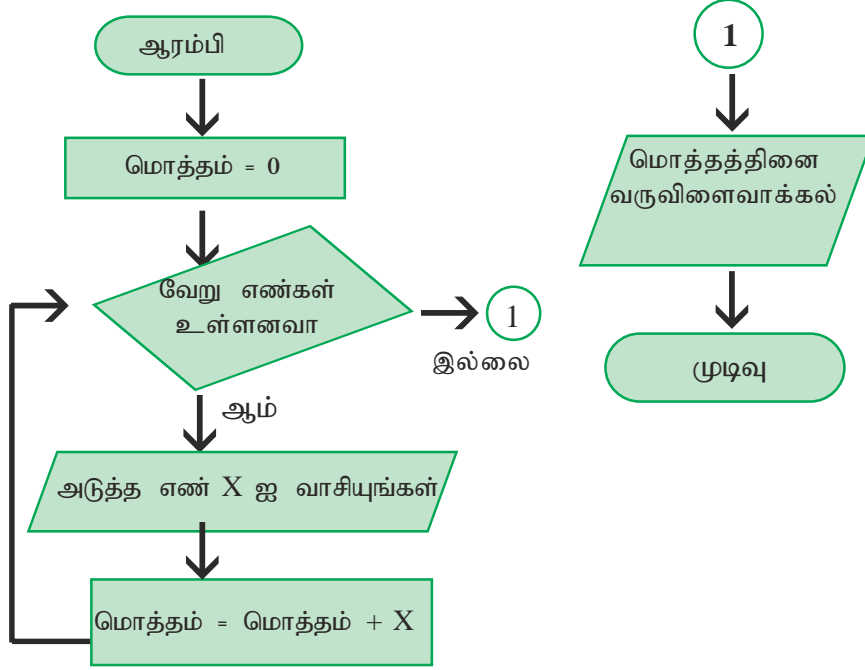
குறியீடு	பெயர்	பயன்பாடு
	தொடக்கி/ முடிப்பான் Terminator	நெறிமுறையின் தொடக்கம்/முடிவு (Start/Stop)னை காண்பிக்க.
	செயன்முறை Process	செயன்முறையிலுள்ள ஒரு படியினை அல்லது பல படிகளை காண்க
	உள்ளீடு / வருவிளைவு Data Input / Output	தரவு உள்ளீடு / வருவிளைவினை காண்பிக்க
	தீர்மானம் Decision	தீர்மான செயற்பாட்டினை காண்பிக்க
	தொடுப்பி Connector	வெவ்வேறு பாய்ச்சற்பட பகுதிகளை தொடுப்பதற்கு
	பாய்ச்சல் கோடு Flow Lines	தரவுகளின் பாய்ச்சல் திசைகளை காண்பிப்பதற்கு

**உதாரணம் :** செவ்வகத்தின் பரப்பளவைப் பெறுவதற்கான நெறிமுறையினை பாய்ச்சற் படம் மூலம் காட்டுக.



பாய்ச்சற் படமானது பல பக்கங்களுக்கு நீண்டு செல்லும் சந்தர்ப்பங்களில் பாய்ச்சலின் திசையை சிறிய கோடொன்றின் மூலம் தெளிவாகக் காட்டுவற்கு தொடுப்பி (Connector) குறியீடு பயன்படுத்தப்படும்.

**உதாரணம்:** தரப்பட்ட எண்கள் சிலவற்றின் மொத்தப் பெறுமானத்தைக் கணிப்பதற்கான நெறிமுறையினைப் பாய்ச்சற் படத்தின் மூலம் எடுத்துக்காட்டுக.



**செயற்பாடு 1.4**

மேற்கூறப்பட்ட உதாரணத்தில் தரப்பட்ட பாய்ச்சற் படத்தை எண்கள் சிலவற்றின் மொத்தம், இடையம் ஆகியவற்றை கணிப்பதற்கு ஏற்ற வகையில் நீடிப்புச் செய்க.

**போலிக்குறிமுறை (Pseudocode)**

நெறிமுறை ஒன்றினை கணினி மொழியில் தங்கியிராத சொற்களின் மூலமாக எளிமையாக எடுத்துரைத்தல் [போலிக்குறிமுறை (Pseudocodes) ] எனப்படும். இவ்வாறு எடுத்துரைக்கப் பயன்படுத்தக்கூடிய சொற்கள் மற்றும் அச்சொற்களின் உள்ளடக்கம் ஆகியவற்றுக்கான நியம உடன்பாடுகள் சில உள்ளன. இதனால் நெறிமுறையொன்றைப் போலிக்குறிமுறை மூலமாக

எடுத்துக்காட்டும்போது செய்நிரலர் அல்லாதோரும் புரிந்துகொள்ளக் கூடியதாக அவை எளிமையாக எடுத்துக் காட்டப்படுவதனை உறுதிப்படுத்திக் கொள்ள வேண்டும்.

**உதாரணம் :** செவ்வகத்தின் பரப்பளவைத் துணியும் செயல்முறையைப் பின்வருமாறு போலிக்குறிமுறை மூலமாகக் காட்டமுடியும்:

- Start ஆரம்பம்
- Get the length and width of the rectangle. செவ்வகத்தின் நீளம், அகலம் என்பவற்றைப் பெறுக.
- Area = length x width பரப்பளவு = நீளம் x அகலம்
- Cutput Area பரப்பளவை வருவிளைவாக்குக.
- End முடிவு



## 1.4 விஷுவல் பேசிக்கைப் (VB) பயன்படுத்தி செய்நிரல்களைத் தயாரித்தல்

இனி பிரசினங்களைத் தீர்ப்பதற்கென விஷுவல் பேசிக் (Visual Basic) கணினி மொழியைப் பயன்படுத்தி செய்நிரல்களைத் தயாரிப்பது பற்றிக் கற்போம்



### விஷுவல் பேசிக்

'Visual Basic' அல்லது 'VB' என அழைக்கப்படுவது மூன்றாம் சந்ததிக்குரிய ஒரு கணினி மொழியாகும். 'VB' யினைப் பயன்படுத்தி செய்நிரல்களைத் செயற்படுத்துவதற்கு வரிமொழிமாற்றி (பொருள்கோடலி) பயன்படுத்தப்பட வேண்டும். 'VB' மொழியானது பிரதானமாக 'Windows' பணிசெயல்முறையில் செயற்படுத்துவதற்கான செயல்நிரல்களை தயாரிப்பதற்கு உருவாக்கப்பட்ட கணினி மொழியாகும். ஆகவே, 'VB'யில் எழுதப்பட்ட செய்நிரல்களை 'Linux' போன்ற பணிசெயல் முறைமையில் செயற்படுத்த முடியாது. இது 'VB' மொழியின் அடிப்படைக் குறைபாடாக உள்ளது.

'VB' மொழியைப் பயன்படுத்தி செய்நிரல்களைத் தயாரிப்பது ஏனைய கணினி மொழிகளைப் பயன்படுத்தி செய்நிரல்களைத் தயாரிக்கும் நடைமுறைகளை விட ஓரளவு வேறுபட்டிருக்கும். 'VB' மொழியானது செயற்றிட்டங்களைத் தயாரிக்கும் கருவிகளுடன் இணைந்தவாறு காணப்படுவதே இதற்கான அடிப்படைக் காரணமாகும். இவ்வாறான செயற்றிட்ட வடிவமைப்புக் கருவிகள் ஒருங்கிணைந்த செயற்றிட்ட வடிவமைப்புச் சூழல் 'Integrated Development Environments' (IDE) என அழைக்கப்படும்.

செய்நிரலை இலகுவாக அமைப்பதற்குத் தேவையான எளிமையான வழிமுறைகளை அளிப்பதே இவ்வாறான செயற்றிட்டத் தயாரிப்புக் கருவிகளின் தொழிற்பாடாகும். இவற்றின்மூலமாக செயற்றிட்டங்கள் தொடர்பான தெளிவான அறிவு பெறப்படாவிடினும் கூட எளிமையாக செயற்றிட்டங்களை மேற்கொள்ளக் கூடிய அறிவு கிடைக்கும். எனவே, 'VB' பற்றியும் 'IDE' பற்றியும் தெளிவான விளக்கம் பெறுதல் அவசியமாகும்.

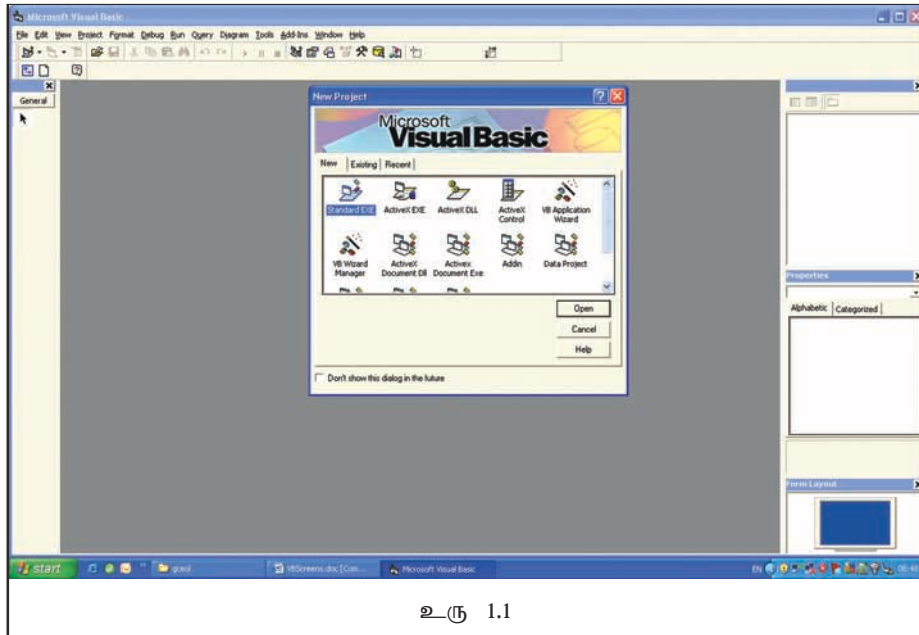


### விஷுவல் பேசிக் செயற்றிட்டத் தயாரிப்பும் IDE முகப்புக்குள் பிரவேசித்தலும்

'VB' செயற்றிட்டங்களை ஆக்குவதற்கான கணிமத்தினை செயற்படுத்தி அதற்குரிய முகப்பினுள் பிரவேசிக்கும் முறை பின்வருமாறு அமையும்.

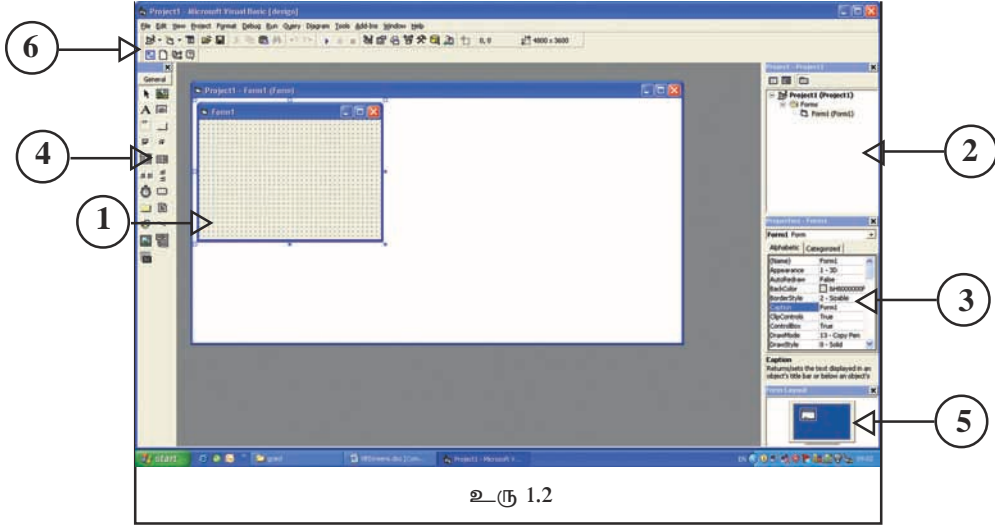
- 1) Start
- 2) All Programs
- 3) Visual Studio 6.0
- 4) Microsoft Visual Basic 6.0

மேற்கூறப்பட்ட கட்டளைகளைச் செயற்படுத்திய பின்னர் பின்வருமாறான முகப்பு திரையில் தோன்றும்:



உரு 1.1

விஷுவல் பேசிக் மூலமாக பல்வேறு வகைப்பட்ட செயற்றிட்டங்களை தயாரிக்க முடியும். இவ்வாறான பல்வேறு செயற்றிட்டங்களைத் தயாரிப்பதற்கென செயற்படுத்தவேண்டிய கட்டளைகளுடன் தொடர்புடைய படவருக்கள் (Icons) திரையில் தென்படும். Standard. EXE வகை செய்நிரல் ஒன்றினை உருவாக்குவதற்கு Standard. EXE படவருவினை சொடக்குவதன் மூலம் அதற்குரிய முகப்பினுள் பிரவேசிக்க முடியும். அவ்வாறான முகப்பு பின்வரும் உருவில் காட்டப்பட்டுள்ளது.



- ① படிவத்தைத் திட்டமிடுவதற்கான சாளரம் (Form Design Window)
- ② செயற்றிட்ட ஆய்வுப் பலகணி (Project Explorer Window)
- ③ சாளரப் பண்புகள் (Window Properties)
- ④ கருவிப் பெட்டி (Tool Box)
- ⑤ படிவ அமைவுச் சாளரம் (Form Layout Window)
- ⑥ பிரதான பட்டி (Main Menu)

**கவனத்திற் கொள்ள வேண்டியது :** உங்கள் கணினியில் நிறுவப்பட்டுள்ள கணிமத்திற்கமைய செயற்படுத்தப்பட வேண்டிய கட்டளைகள் மற்றும் அக் கட்டளைகளைச் செயற்படுத்தப்பட வேண்டிய விதம் ஆகியன மாறுபடலாம்.

'VB' செயற்றிட்டமொன்றைத் தயாரிப்பதற்குத் தேவையான எல்லா கருவிகளும் மேற்கூறப்பட்ட நிகழ்ச்சித்திட்ட வடிவமைப்புச்சூளல் (IDE) முகப்பில் அடங்கியிருக்கும்.

'VB' நிகழ்ச்சித்திட்டம் பல்வேறு விதமான கோப்புகளைக் கொண்டிருக்கலாம். குறித்தவொரு நிகழ்ச்சித்திட்டத்துக்குரிய கோப்புகள் அனைத்தும் ஒருங்கே 'VB' செயற்றிட்டம் (VB Project) என அழைக்கப்படும்.

மேலே காட்டப்பட்டுள்ள இடை முகத்தில் 1 எனும் இலக்கத்தினால் காட்டப்பட்டுள்ள படிவ வடிவமைப்புச் சாளரமானது (Form Design Window) நிகழ்ச்சித் திட்டத்தின் பயன்படுத்தப்படும். 2 எனும் இலக்கத்தினால் காட்டப்பட்டுள்ள பாகத்தில் நீங்கள் மேற்கொள்ளும் எல்லாச் செயற்றிட்டங்களின் (Project) பெயர்களும் அவ்வொரு செயற்பாட்டிற்குமுரிய எல்லா கோப்புகளின் பெயர்களும் பட்டியலிடப்பட்டிருக்கும்



#### செயற்பாடு 1.5

உரு 1.2 இல் காட்டப்பட்டுள்ள எல்லா சாளரங்களும் உங்கள் கணினித் திரையில் உள்ளனவாவென அவதானிக்கவும். ஏதேனும் சாளரம் காணப்படவில்லையெனின் அதனைத் திரையில் பெறும் முறையை அறிந்து கொள்ளுங்கள்.

மேலே உருவில் 3 எனும் இலக்கத்தினால் குறிப்பிடப்பட்டுள்ள பகுதியின் மூலம் 'VB' செயற்றிட்டத்தில் அடங்கியுள்ள எல்லா துணை உறுப்புகளினதும் இயல்புகள் (Properties) காண்பிக்கப்படும். மேலும், அவ் இயல்புகளின் பெறுமானங்களை மாற்றியமைக்கவும் முடியும்.



#### செயற்பாடு 1.6

ஆய்வுப்பலகணியில் தற்போதைய செயற்றிட்டத்தின் பெயரின் மீது சொடக்கி செய்து செயற்றிட்ட இயல்புகள் எனும் பகுதிக்குச் சென்று செயற்றிட்டத்தின் பெயரினை First Project என மாற்றுக.

உருவில் 1.2 இல் 4 எனக் குறிப்பிடப்பட்டுள்ள பகுதி கருவிப் பெட்டி (Tool Box) எனப்படும். 'VB' நிகழ்ச்சித்திட்டமொன்றைத் தயாரிப்பதற்கு அதிகளவில் பயன்படுத்தப்படும் கட்டுப்பாட்டுத் தொகுதிகள் (Controls) எனும் துணைப்பாகங்கள் இங்கு காணப்படும்.


உரு 1.2 இல் 5 எனக் காட்டப்பட்ட பாகத்தினால் உங்கள் நிகழ்ச்சித் திட்டத்தைச் செயற்படுத்தும்போது அது திரையில் தென்படும் விதம் காட்டப்படும்.



#### செயற்பாடு 1.7

படிவ வடிவமைப்புச் சாளரத்தில் உள்ள Form-1 இல் உள்ள 'Caption' எனும் இயல்பை 'First Form' என மாற்றுக. இந்த படிவத்தின் அகலம் (Width) எவ்வளவாகும்?

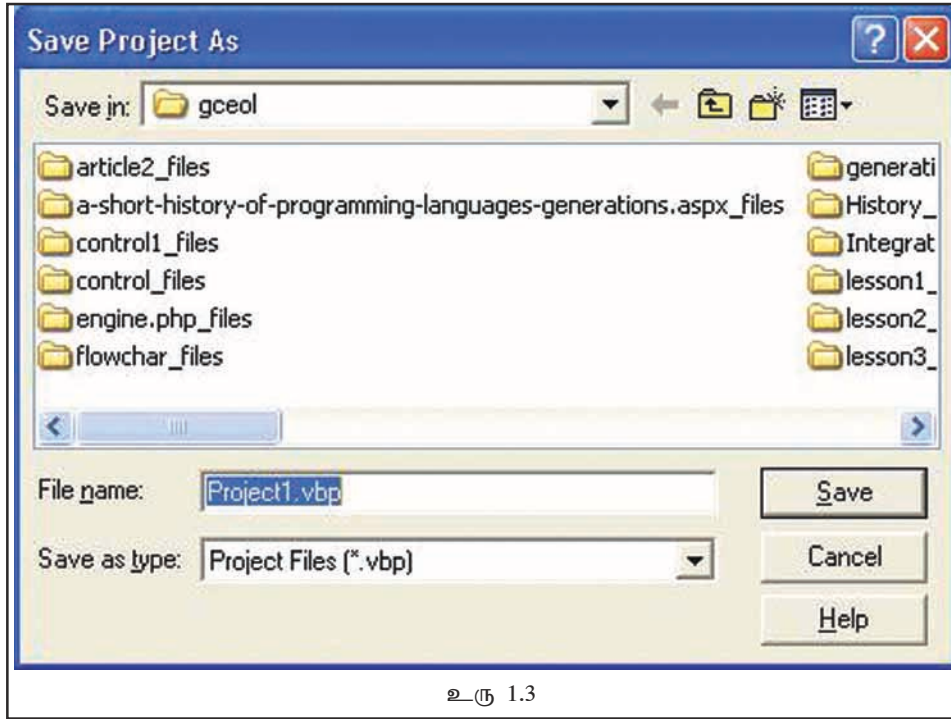




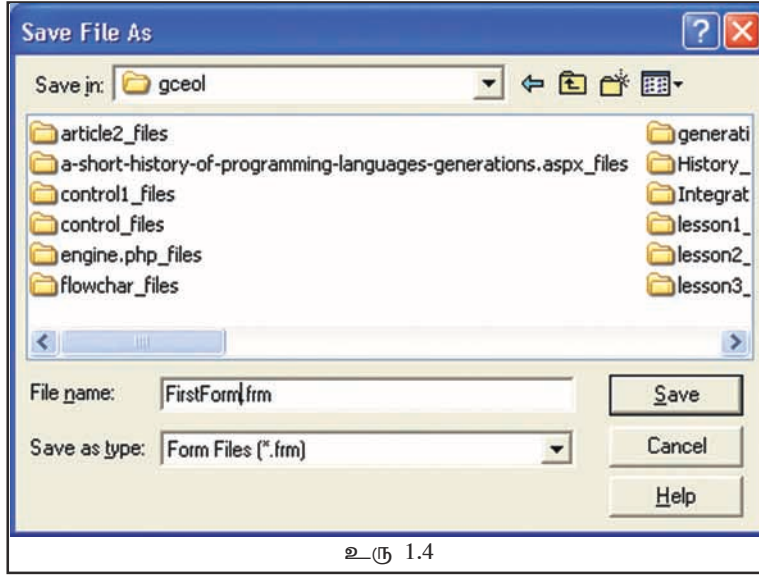
**செயற்பாடு 1.8**

உருவில் 6 எனும் இலக்கத்தினால் காட்டப்படும் பிரதான பட்டியில் (Main Menu) Run → Start என்பதனைச் செயற்படுத்தி ஏற்படும் விளைவை அவதானிக்குக. மீண்டும் Run → End எனும் கட்டளைகளைச் செயற்படுத்தி கிடைக்கும் பெறுபேறுகளை அவதானிக்க


நீங்கள் இதுவரை தயாரித்தவற்றை File → Save Project எனும் கட்டளைகள் மூலமாக நீங்கள் விரும்பிய பெயரில் சேமிக்கவும் (Save).



இப்போது செயற்றிட்ட சாளரத்திலுள்ள உங்களது செயற்றிட்டத்துடன் தொடர்ச்சியாக எல்லா கோப்புகளையும் சேமிப்பதற்கு தேவையான பெயர்களை நீங்கள் வழங்க வேண்டும்.




பிரதான பட்டியிலுள்ள File → Exit எனும் கட்டளைகளைச் செயற்படுத்தி உங்களால் செயற்றிட்ட வடிவமைப்புச் சூழலிலிருந்து (VB IDE ) வெளியேற (Exit) முடியும்.



**செயற்பாடு 1.9**

முன்னர் நீங்கள் சேமித்த 'First Application.Vbp' என காட்டப்பட்டுள்ள படவுருவின்மீது (Icon) இருதடவைகள் சொடக்கி ஏற்படும் விளைவை அவதானியுங்கள்.



**செயற்பாடு 1.10**

File → Open Project ஆகிய கட்டளைகளைச் செயற்படுத்துவதன் மூலமாக முன்னர் சேமிக்கப்பட்ட செயற்றிட்டம் (Project) ஒன்றை மீண்டும் திறக்கக்கூடிய முறையை அறிந்து கொள்ளுங்கள்.



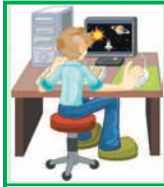
## 1.5 விஷுவல் பேசிக் நிகழ்ச்சித்திட்டமொன்றை உருவாக்கும் படிமுறைகள்

'VB' கணினிமொழியானது நிகழ்வுச் செலுத்துகை (Event Driven) வகை நிகழ்ச்சித்திட்டங்களைத் தயாரிப்பதற்கென தயாரிக்கப்பட்ட மொழியாகும். இவ்வாறான மொழியைப் பயன்படுத்தி தயாரிக்கப்பட்ட நிகழ்ச்சித்திட்டங்களின் செயற்பாடானது நிகழ்ச்சித்திட்ட முகத்தில் தோன்றும் படவுருக்களின் (ICON) மீது பயன்படுத்துபவரினால் மேற்கொள்ளப்படும் இடைவினைகளில் [(சொடக்கி) 'Click', (நகர்வு) 'Move', (இழு) 'Drag'.] தங்கியுள்ளது.

'VB' நிகழ்ச்சித்திட்டத்தின் இடை முகத்தில் ஏற்படும் இடைவினைகளினை நிகழ்ச்சிகளாக (Events) உருவாக்கக்கூடிய சாவிகள் கட்டுப்படுத்திகள் (Controls) என அழைக்கப்படும். 'VB' நிகழ்ச்சித் திட்டத்தைத் தயாரிக்கும்போது இடைமுகத்தில் தோன்றும் படிவமும் அவ்வாறானதொரு கட்டுப்படுத்திச் சாவி ஆகும்.

'VB' நிகழ்ச்சித் திட்டங்களைத் தயாரிப்பதற்கெனப் பயன்படுத்தக்கூடிய கட்டுப்பாட்டுச்சாவிகள் ஏராளம் உள்ளன. இவற்றுள் அதிகளவில் பயன்படுத்தப்படும் பிரதான கட்டுப்பாட்டு சாவிகள் 'IDE' முகப்பில் உள்ள கருவிப் பெட்டியில் காணப்படும்.

சுட்டிச் சுட்டுவாணை (Mouse Pointer) விஷுவல் பேசிக் கருவிப் பெட்டியின் மீது உள்ள படவுருவின் மீது கொண்டு செல்லும்போது தோன்றும் செய்தியிலிருந்து அப்படவுரு எதுவென எம்மால் இனங்கண்டு கொள்ள முடியும்.



### செயற்பாடு 1.11

1. கருவிப்பெட்டியிலுள்ள படவுருக்களின் மீது சுட்டிச் சுட்டு வாணைக் கொண்டு சென்று அப் படவுரு எதற்குரியதென இனங்காணுங்கள்.
2. 'IDE' முகத்திலுள்ள படிவத்தின் அளவுகள் (நீளம், அகலம்) ஆகியவற்றை எவ்வாறு மாற்றலாம் எனக் கண்டறியுங்கள். இவ்வாறு மாற்றம் செய்யும் போது 'Properties' பகுதியிலுள்ள உயரம், அகலம் ஆகிய இயல்புகள் மாற்றமடையும் விதத்தை அவதானியுங்கள்.
3. கருவிப் பெட்டியின் வலது மூலையில் உள்ள  எனும் படவுருவை சொடக்கி கிடைக்கும் விளைவை அவதானியுங்கள். பிரதான பெட்டியிலுள்ள View → Tool Box இனை சொடக்கி மீண்டும் கருவிப் பெட்டியை திரையின் மீது பெற்றுக்கொள்க.



## படிவத்தில் கட்டுப்பாட்டு (Controls) சாவிகளை இடுதல்

இதனை பிரதானமாக இருமுறைகளில் செய்ய முடியும்.

- 1) தேவையான கட்டுப்பாட்டுச் சாவியின் மீது இடைவிடாது இரு தடவைகள் சொடுக்குதல் இதன்போது அக் கட்டுப்பாட்டுச் சாவியை படிவத்தின் மத்தியில் நிலைப்படுத்தலாம். பின்னர் இக்கட்டுப்பாட்டுச் சாவியை சுட்டியைப் பயன்படுத்தி படிவத்தின் தேவையான இடத்துக்குக் கொண்டு செல்ல முடியும்.
- 2) தேவையான கட்டுப்பாட்டுச் சாவியினை சுட்டியினால் தெரிவு செய்து சுட்டுவானை படிவத்தின் உரிய இடத்துக்குக் கொண்டு சென்று சுட்டியைச் சொடக்கி தேவையான அளவுக்கு பருமனை மாற்றிக் கொள்ள முடியும்.

படிவத்தின் மீது கட்டுப்பாட்டுச் சாவியை உட்புகுத்தியதும் 'IDE' மூலமாக அக் கட்டுப்பாட்டுச் சாவியின் அநேக இயல்புகளுக்கு (Properties) பொருத்தமான பெறுமானங்கள் தன்னியக்கமாகவே வழங்கப்படும். பின்னர் இப் பெறுமானங்களை தேவைக்கேற்ப Properties window இனுடாக மாற்றியமைக்கலாம்.

'VB' நிகழ்ச்சித்திட்டமொன்றைத் தயாரிக்கும்போது மூன்று படிமுறைகள் பின்பற்றப்பட வேண்டும்.

- 1) படிவத்தில் கட்டுப்பாட்டுச் சாவிகளை இடல்.
- 2) கட்டுப்பாட்டுச் சாவிகளின் இயல்புகளை (Properties) தேவைக்கேற்ப மாற்றியமைத்தல்.
- 3) கட்டுப்பாட்டுச் சாவிகளின் தொழிற்பாட்டை கட்டுப்படுத்துவதற்கான கட்டளைகளை உப செய்நிரலாகத் (Sub program) தொடர்புபடுத்தல்.

செயற்பாடு 1.12

1 → First Name

2 → Last Name

3 → Full Name

4 → First Name

5 → Last Name

6 → Full Name

7 → Display Full Name

செயற்பாடு 1.12 இல் உள்ள உருவில் காட்டப்பட்டவாறு ஏழு கட்டுப்படுத்திகளை (Control) உட்படுத்துங்கள். இக்கட்டுப்படுத்தி(Control) களின் வகைகளையும் இயல்புகளையும் பின்வரும் அட்டவணையில் காட்டப்பட்டுள்ள பெறுமானங்களுக்கு ஏற்ப மாற்றுக.

உங்களால் தயாரிக்கப்பட்ட நிகழ்ச்சியை பிரதான பட்டியிலுள்ள (Menu) File → Save Project கட்டளை மூலமாக 'My Project.Vdb' எனும் பெயரில் சேமிக்க.

a) Form Object Properties

Name : first frm  
Caption : First Form  
Width : 10000  
Height : 6000

b) Object 1

Type : Label  
Name : fname1bl  
Caption : First Name  
Width : 1000  
Left : 450  
Top : 400  
Height : 200

c) Object 2

Type : Label  
Name : lname1bl  
Caption : Last Name  
Width : 1000  
Left : 450  
Top : 950  
Height : 200

d) Object 3

Type : Label  
Name : fullname1bl  
Caption : Full Name  
Width : 1000  
Left : 450  
Top : 1500  
Height : 200

e) Object 4

Type : Text Box  
Name : fname.txt  
Width : 3000  
Left : 1600  
Top : 400  
Height : 200  
Text :

f) Object 5

Type : Text Box  
Name : lname.txt  
Width : 1600  
Left : 1600  
Top : 950  
Height : 200  
Text :

g) Object 6

Type : Text Box  
Name : fullname.txt  
Width : 5000  
Left : 1600  
Top : 1500  
Height : 200  
Text :

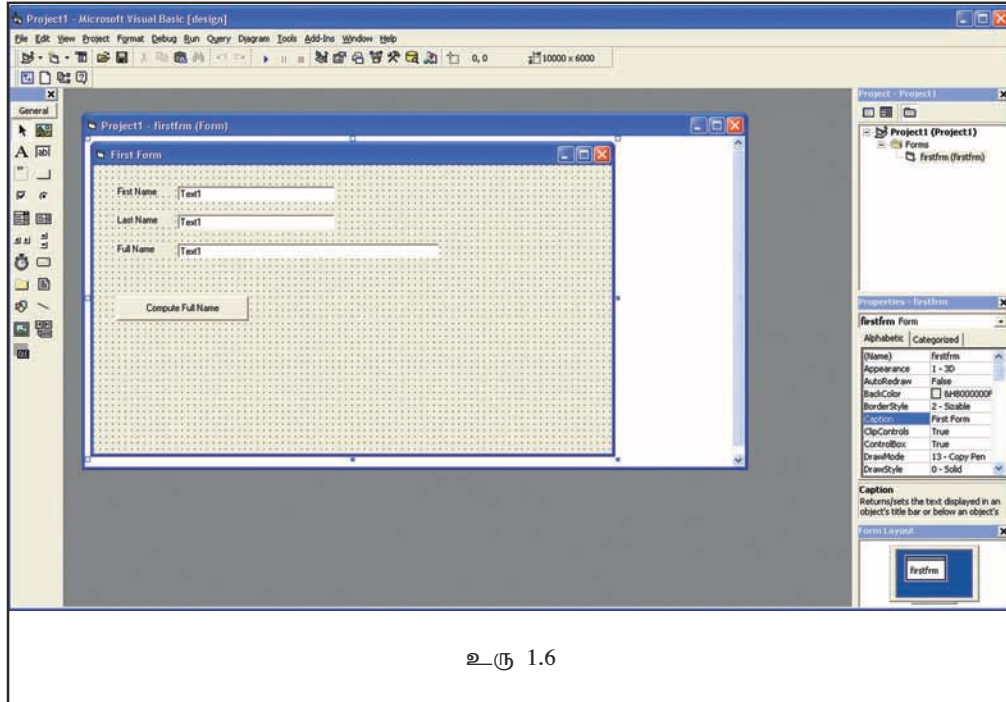
h) Object 7

Type : Command Button  
Name : fullnamecmd  
Caption : Compute Full Name  
Width : 2500  
Left : 450  
Top : 2500  
Height : 200



## 1.6 கட்டுப்படுத்தியின் (Control) செயற்பாட்டுக்கான குறிமுறையை (Code) எழுதுதல்

இதற்காக முன்னைய செயற்பாட்டில் உருவாக்கப்பட்ட கீழுள்ள முகப்புத் திரையில் தோன்றும் செயற்றிட்டத்தினை மீண்டும் திறந்து கொள்ளுங்கள்.



உரு 1.6

இந்த முகத்திலுள்ள "Display Full Name" எனக் குறிப்பிடப்பட்டுள்ள கட்டுப்பாட்டுச் சாவியின் (Command Button) மீது இரு தடவைகள் சொடக்குவதனால் (Double click) பின்வருமாறான முகத்தைப் பெற்றுக்கொள்ள முடியும்.





'VB' நிகழ்ச்சித் திட்டமானது பல 'VB' உப செய்நிரல்களின் (sub Programmes) தொகுப்பாகக் கருதப்படலாம். இவ்வாறான 'VB' உபநிகழ்ச்சித் திட்டங்களை எழுதும்போது கவனத்திற்கொள்ளவேண்டிய விடயங்கள் பற்றி அறிந்து கொள்ள வேண்டியது அவசியமாகும்.



### 'VB' செயற்றிட்டத்தின் கட்டமைப்பு

'VB' செயற்றிட்டமானது பிரதானமாக ஒன்று அல்லது பல படிவங்களைக் (form) கொண்டிருக்கலாம். 'VB' யிலுள்ள 'IDE' இனை செயற்படுத்தியதும் தன்னியக்கமாக உருவாக்கப்பட்டிருக்கும் படிவத்திற்கு மேலதிகமாக படிவங்கள் தேவைப்படும்போது Project → Add Form கட்டளைகளைச் செயற்படுத்தி தேவையான படிவங்களினை உருவாக்கிக் கொள்ளமுடியும்.

ஒவ்வொரு கட்டுப்பாட்டுச்சாவிடனுமும் இணைந்ததாக பல இடைவினை நிகழ்ச்சிகள் அமைந்திருக்கும். உதாரணமாக : பொத்தான் மீது சொடக்குதல் (Mouse click), பொத்தானின் மீது சுட்டானினை கொண்டு செல்லல் (Mouse Over) போன்றன. இவ் இடைவினைகளுடன் ஒரு தொகுதி கட்டளைகள் அடங்கிய செய்நிரல் ஒன்று தொடர்புபடுத்தப்பட்டிருக்கும். இச் செயல்நிரலானது ஒரு செயல்முறையின் (Procedure) கட்டமைப்பினை கொண்டிருக்கும்.

'VB' மென்பொருளினை செயற்படுத்தியதும் அதற்குரிய படிவமும் கட்டுப்பாட்டுச் சாவிகளும் திரையில் தென்படும். பயனர் (User) இக்கட்டுப்படுத்திகளில் இடைவினை செயற்பாடுகளை மேற்கொள்வதன் மூலம் (Click, Move, Drag) அதனுடன் தொடர்பான செய்நிரலை செயற்படுத்த முடியும்.

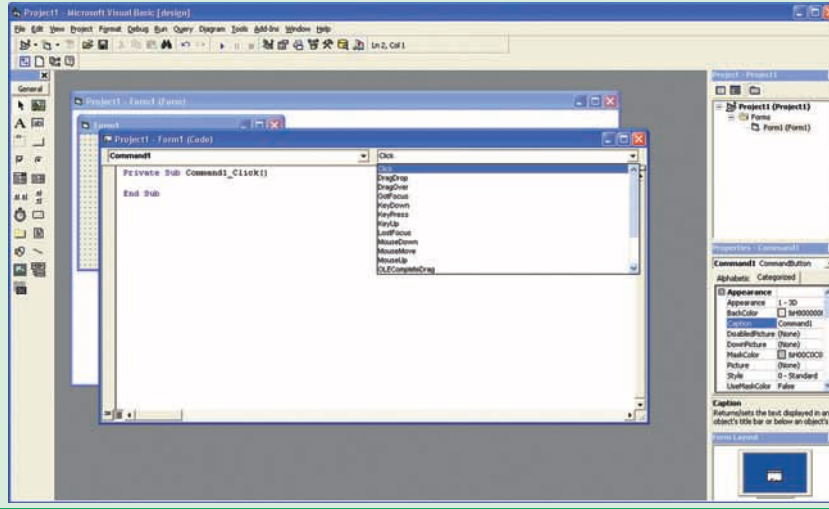
#### மாறிகள் :

செய்நிரலை தரவுகளுடன் செயற்படுத்தும்போது உரிய தரவுகளை பிரதான நினைவகத்தில் சேமித்து வைத்துக் கொள்வதற்கு மாறிகள் பயன்படுத்தப்படகின்றன. இவை பற்றி பின்னர் விரிவாக பார்ப்போம்.



## செயற்பாடு 1.14

'VB', இன் படிவம் ஒன்றின் இன் மீது Command Button Control ஒன்றை உருவாக்குக. இக் கட்டுப்படுத்தியின் மீது இரட்டைச் சொடுக்கல் செய்து குறிமுறை சாளரத்தைப் பெற்றுக்கொள்க. இந்த குறிமுறைச் சாரளத்திலுள்ள நிகழ்வுச் சாரளத்தினுள் பிரவேசித்து (நிழற்றப்பட்டுள்ள உருவில் காட்டப்பட்டுள்ளது) Command button மூலம் உருவாக்கத்தக்க நிகழ்வுகளை அவதானியுங்கள். இந்த நிகழ்வுகளை தெரிவு செய்ததும் குறிமுறைச் சாரளத்தில் ஏற்படும் மாற்றங்களை அவதானியுங்கள்.



## VB மொழியின் சிறப்புச் சொற்கள் (Reserved Words / KeyWords)

'VB' கணினி மொழியிலும் ஏனைய அனேக கணினி மொழிகளில் உள்ளது போன்றவாறான சிறப்புச் சொற்களின் பட்டியல் உள்ளது. 'Private', 'Sub' 'End', 'If', 'End If' ஆகியன இதற்கான சில உதாரணங்களாகும். 'VB' செய்நிரலை உருவாக்கும்போது இச்சொற்களின் பயன்பாடு அவற்றின் பொருளுக்குப் பொருந்தும் வகையில் மேற்கொள்ளப்பட வேண்டும். உதாரணமாக 'Sub' எனப்படும் சொல் 'VB' செய்நிரலில் செயல்முறைய ஒன்றினை ஆரம்பிப்பதற்காவும் 'End Sub' எனப்படும் சொல் செயல்முறையின் முடிவைக் காட்டுவதற்காகவும் பயன்படுத்தப்படும். இவை பொதுவாக குறிமுறைச் சாளரத்தில் நீலநிறத்தில் தோன்றும்.



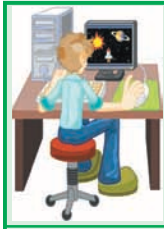
## VB செயல் முறைகள்

'VB' செய்நிரலானது செயல்முறைகள் பலவற்றின் தொகுப்பென இதற்கு முன்னர் கற்றுள்ளோம். 'VB' செயல்முறைகள் மூன்று வகைப்படும். அவையாவன :-

- 1) நிகழ்வுச் செயல்முறைகள் (Event Procedures)
- 2) பொதுவான செயல்முறைகள் (General Procedures)
- 3) செயற்கூற்று செயல்முறைகள் (Function Procedures)

படிவம் ஒன்றிலுள்ள கட்டுப்படுத்தியுடன் இணைந்த செயல்முறை நிகழ்வுச் செயல்முறை (Event Procedures) என அழைக்கப்படும். இச் செயல்முறைகளின் செய்நிரலில் 'Private Sub' எனும் சொற்களினால் ஆரம்பிக்கப்படும். இவ்வாறான செயல்முறைகள் அது சார்ந்துள்ள படிவத்திற்கு மட்டுமே தனித்துவமாக அமைவதுடன் 'VB' செய்நிரலை வேறொரு படிவத்திலுள்ள கட்டுப்படுத்தி மூலமாக செயற்படுத்த முடியாது.

கட்டுப்படுத்தியுடன் இணைந்திராத செயல்முறைகள் பொதுவான செயல் முறைகள் (General Procedures) என அழைக்கப்படும். இவ்வாறான செயல்முறையின் ஆரம்பத்தில் 'Private Sub' அல்லது 'Public Sub' எனும் சிறப்புச் சொற்கள் பயன்படுத்தப்படும். 'Private Sub' என ஆரம்பிக்கப்படும் செயல்முறை குறிப்பிட்ட படிவத்துக்கு மட்டுமே தனித்துவமாக அமைவதுடன் 'Public Sub' என ஆரம்பிக்கும் செயல்முறை VB நிகழ்ச்சித்திட்டத்தில் அடங்கியுள்ள எந்தவொரு படிவத்தின் மூலமாகவும் செயற்படுத்த முடியும்.



### செயற்பாடு 1.15

செயற்கூற்று செயல்முறை (Function procedures) ஏனைய வகைப்பட்ட செயல்முறைகளை விட எவ்வாறு வேறுபடுகின்றனவென ஆராய்ந்தறிக. செயற்கூற்று செயல்முறையில் பயன்படுத்தக்கூடிய சிறப்புச் சொற்கள் எவையென அறிந்துகொள்ளுங்கள்.

செவ்வகத்தின் நீளம், அகலம் தரப்படுமிடத்து அதன் பரப்பளவைத் துணிவதற்காக 'VB' செயல்முறையினை உருவாக்கும் முறையை அறிந்துகொள்வோம். இதற்கு இச்செய்நிரலுக்கு அவசியமான இடைமுகத்தை (Interface) முதலில் உருவாக்கிக் கொள்ள வேண்டும். இப்பிரசினத்தைத் தீர்ப்பதற்கு பின்வருமாறான இடைமுகத்தைப் பயன்படுத்தலாம்.

உரு 1.8

இங்கு 'Length', 'Height' 'Area' ஆகிய கட்டுப்படுத்திகள் முகப்பு அடையாள கட்டுப்படுத்தி ( Label Control) எனப்படுவதுடன் அவற்றுக்கு எதிரே உள்ளவை பாடப் பெட்டி கட்டுப்படுத்தி (Text Box Control) ஆகும். 'Compute' எனப்படுவது கட்டளை பெட்டி கட்டுப்படுத்தி (Command Box Control) ஆகும்.

இம்முகத்தில் 'Length' 'Height' ஆகியவற்றுக்குப் பெறுமானங்களை வழங்கி 'Compute' எனும் கட்டுப்படுத்தியை சொடுக்கும் செயல்முறை ஒன்றின் மூலம் செவ்வகத்தின் பரப்பளவை கணித்து அதன் பெறுமானத்தை 'Area' எனும் பகுதியில் வெளிப்படுத்த வேண்டும். இதனை மேற்கொள்ள 'Compute' எனும் கட்டுப்படுத்திக்கு குறித்த செயல்முறையினை தொடர்புபடுத்த வேண்டும்.

இச் செய்கையை மேற்கொள்வதற்கு Command box இல் Click எனும் Event உடன் தொடர்புபடுத்த வேண்டிய செய்நிரல் வருமாறு:

**Private Sub Compute\_Click()**

```

length      = Val (length.Text)
height      = Val (height.Text)
Area        = length*height
Area.Text   = Str (Area)
End Sub.

```

இச்செய்நிரலில் பின்வரும் இயல்புகளை அவதானிக்க முடியும்.

1. செய்நிரல் 'Private Sub' எனும் சிறப்புச் சொல்லுடன் ஆரம்பிக்கப்பட்டுள்ளதாலும் 'End Sub' என முடிக்கப்பட்டிருப்பதனாலும் இதுவொரு செயல்முறை ஆகும். இதில் Private என்னும் சிறப்புச் சொல் காணப்படுவதால் இது அதற்குரிய படிவத்தினால் மட்டுமே செயற்படுத்தக்கூடியதாகும். 'Private Sub Compute - Click ()' எனும் கூற்று கட்டளை அன்று இது மொழிபெயர்ப்பு செய்நிரலிற்கு உபசெய்நிரலின் (Sub - Programme) ஆரம்பத்தினை காட்டுவதற்காக உள்ளது.

2. இக்கருமத்தை இனங்கண்டு கொள்ளவென 'Compute\_click()' எனும் பெயர் பயன்படுத்தப்பட்டுள்ளது. இதில் 'Compute' என்பது கட்டளைப் பெட்டியின் பெயராக உள்ளதுடன் '\_click' எனும் பகுதி VB யின் சொடுக்கல் நிகழ்விற்காக சுயமாக உருவாக்கப்பட்ட பாகமாகும்.
3. செயல்முறை ஒன்றின் உள்ளடக்கமானது (Body) கட்டளைகள் சிலவற்றின் தொகுப்பாகும். இவை ஒரு வரியில் ஒரு கட்டளை என்றவாறு பல வரிகளில் அமையப்பெற்றிருக்கும். இவ்வாறு ஒரு வரியில் தரப்பட்டுள்ள கட்டளை கூற்று (Statement) எனப்படும்.
4. தரப்பட்டுள்ள செயல்முறையின் முதற்கட்டளை  $Length = Val (length.text)$  ஆகும். இங்கு குறிப்பிடப்பட்டுள்ள '=' எனும் அடையாளம் மதிப்பளி வினைக்குறி (Assignment Operator) எனப்படும். வினைக்குறி என்பது 'VB' யின் மூலமாக குறிப்பிட்டவொரு செய்பணியை (Operation) மேற்கொள்ள குறிப்பிடும் குறியீடாகும். இங்கு இடப்பக்கத்திலுள்ள 'Length' எனப்படுவது மாறியாகும். இது கணினியின் பிரதான நினைவகத்திலுள்ள ஒரு பகுதியை அணுகுவதற்கு (Access) பயன்படுத்தும் பெயராகும். இங்கு '=' எனும் வினைக்குறியின் அடிப்படைக் கருமமாக அமைவது வலது பக்கத்தில் தரப்பட்டுள்ள கருமத்தை மேற்கொண்டு அதன் பெறுபேற்றை இடப்பக்க நினைவகப் பகுதியில் சேமிக்கவும் என்பதாகும். இங்கு 'Val (Length.text)' எனும் பாகத்தில் 'Length.text' என்பதால் காட்டப்படுவது 'Length' எனப் பெயரிடப்பட்டுள்ள கட்டுப்படுத்தியில் text எனும் இயல்பின் பெறுமானத்தை எடுக்கவும் என்பதாகும். ஏதேனுமொரு கட்டுப்படுத்தியின் இயல்பிற்குரிய பெறுமானத்தை
 

Control Name. Property Name

 எனும் மாறியைப் பயன்படுத்தி பெற்றுக்கொள்ளலாம். இங்கு 'Val' எனப்படுவது 'VB' யினால் அறிமுகம் செய்யப்பட்ட மற்றொரு செய்கையாகும். இச்செய்கையின் மூலமாக 'length.text' இன் பெறுமானத்தை கணிதச் செய்கைக்குப் பயன்படுத்தக் கூடிய பெறுமானமாக மாற்ற முடியும்.
5.  $Area = length * height$  என்பதில் '\*' எனும் குறியீடு பெருக்கல் எனும் கணிதச் செய்கையை காட்டும். இதன்போது 'Length', 'Height' ஆகிய மாறிகள் பெறுமானங்களை நினைவகத்தில் பெற்று அப்பெறுமானங்களின் பெருக்கத்தை மேற்கொண்டு 'Area' எனும் மாறியில் சேமிக்கும்.

6. இறுதிக் கட்டளைக் கூற்றாகிய Area.Text = str (Area) இலுள்ள 'str' என்பது 'VB' யில் உள்ள மேலுமொரு செய்கையாகும். இதன்மூலமாக 'Area' எனும் மாறியின் எண் பெறுமானத்தை சரம் (string) எனும் தரவு வகைக்கு (Data type) மாற்ற முடியும். இவ்வாறு செய்யப்பட்ட பின்னர் கிடைக்கும் பெறுமானம் 'Area' எனும் கட்டுப்படுத்தியின் 'Text' எனும் இயல்புக்கு பிரதியிடப்படும்.

மேற்குறிப்பிடப்பட்டவற்றை பகுப்பாய்வு செய்து பின்வரும் பிரதான விடயங்களை அறிந்து கொள்ள முடியும்:

1. செயல்முறையானது கூற்றுக்கள் சிலவற்றின் தொகுப்பாகும்.
2. செயல்முறையில் பெறுமானங்களை தற்காலிகமாக தேக்கிவைக்க மாறிகள் பயன்படுத்தப்படும்.
3. 'VB' மூலமாக செயற்படுத்தப்பட வேண்டிய கருமங்களைக் காட்டுவதற்கு கூற்றுகள் பயன்படுத்தப்படும். இவ்வாறான கூற்றுகள், மாறிகள் மற்றும் பல்வேறு வகைப்பட்ட 'VB' வினைக் குறிகளினால் (Operators) ஆக்கப்பட்டிருக்கும்.
4. 'VB' யில் படிவமொன்றிலுள்ள கட்டுப்படுத்தியின் குறிப்பிட்ட இயல்பொன்றின் பெறுமானத்தை 'ControlName.Property Name' எனும் மாறியின் மூலமாக பெற்றுக்கொள்ளவோ அல்லது மாற்றிக் கொள்ளவோ முடியும்.

மேலே காட்டப்பட்ட நிகழ்வுச் செயல்முறை (Event Procedure) பின்வருமாறு பொதுச் செயல்முறை (General Procedure), நிகழ்வுச் செயல்முறை ஆகியவற்றின் தொகுப்பாகவும் அமைக்கப்படலாம்.

```
Private Sub Compute _ Click ()
    ComputeArea (Val (length.Text),
                Val (height.Text))
End Sub
```

```
Sub ComputeArea (length, height)
    Area = length*height
    Area.Text = str(Area)
End Sub
```



இங்கு 'ComputeArea' எனப்படுவது கட்டுப்படுத்தியுடன் தொடர்பான நிகழ்வுச் செயல்முறை அன்று ஆகையால் அது பொதுச் செயல்முறை என அழைக்கப்படும். ஆகவே, அது நிகழ்ச்சித்திட்டத்திலுள்ள போது எந்தவொரு கட்டுப்படுத்தி மூலமாகவும் அதனை செயற்படுத்தத்தக்கதாக இருக்கும்.



#### செயற்பாடு 1.16

மேலே விளக்கப்பட்ட 'VB' நிகழ்ச்சித்திட்டத்தைத் தயாரியுங்கள். அந்நிகழ்ச்சித் திட்டத்தை செயற்படுத்திக் கிடைக்கும் பெறுமானங்களை அவதானியுங்கள்.



#### குறிப்புரைகள் (Comments)

செவ்வையான செய்நிரலாக்கம் செய்யும் போது குறிப்புரைகள் பயன்படுத்தப்படுல் அவசியமாகும். குறிப்புரைகளை செய்நிரலில் உள்ள ஏனைய கட்டளைகளைப் போன்று செயற்படுத்த முடியாது. செய்நிரலினுள் குறிப்புரைகளை உட்படுத்துவதன் முக்கிய நோக்கம் அச்செய்நிரலை இலகுவாக விளங்கிக் கொள்ளத் தேவையான விவரங்களை வழங்குவதாகும். 'VB' குறிப்புரைகளை உட்படுத்தவென (') அல்லது 'rem' ஆகிய விசேட குறியீடுகள் பயன்படுத்தப்படும். இக்குறியீடு இடப்பட்ட இடத்திற்கு அப்பால் தொடங்கிய அவ்வரியின் இறுதி வரைக்குமுள்ள விடயங்கள் குறிப்புரைகளாகும். ஏதேனுமொரு விபரமானது ஏனைய வரிகளுக்கும் தொடர்ந்து செல்லும் சந்தர்ப்பங்களில் அவ் ஒவ்வொரு வரியும், '\' 'rem' எனும் விசேட குறியீட்டுடன் ஆரம்பிக்கப்பட வேண்டும். 'VB IDE' யின் குறிப்புரைகள் பச்சை நிறத்தில் காணப்படும்.

#### உதாரணம் : Program 8

```
'
rem'Sub Program to add two numbers
'
Private Sub fullnamecmd_Click()
total = 0
total = Val(firsttxt.Text) + Val(secondtxt.Text) 'add two numbers
totaltxt.Text = total 'print the result
End Sub
```



### செயற்பாடு 1.17

1. மேலே தரப்பட்ட உதாரணத்திலுள்ள குறிப்புரைகளை இனங்காணுங்கள்.

2.  $Sum = Val(txtFirst.txt) + Val(txtSecond.txt)$  எனும் வரிசையை  
' $Sum = Val(txtFirst.txt) + Val(txtSecond.txt)$ ' எனும் விதத்தில் மாற்றமுறச் செய்தால் கிடைக்கும் பெறுபேறுகளை கலந்துரையாடுக.



### விஷுவல் பேசிக் தரவு வகைகள்

'VB' யில் பயன்படுத்தக்கூடிய தரவுகளை, அவற்றின் தன்மை, அவற்றைச் சேமிக்கத் தேவையான இடவசதி, அவற்றினால் செயற்படுத்தக்கூடிய செய்கைகள் ஆகியவற்றின் அடிப்படையில் வகைப்படுத்தலாம். இவ்வாறான பல்வேறு தரவு வகைகள் 'Data Types' என அழைக்கப்படும்.

'VB' யின் மூலம் அறிமுகஞ் செய்யப்பட்டுள்ள தரவு வகைகளை ('Data Types') சில அடிப்படைப் பகுதிகளாகப் பிரிக்கலாம்: அவையாவன Number, String, Currency, date, மற்றும் Variant

நிகழ்ச்சித்திட்டமொன்றில் பயன்படுத்த எதிர்பார்க்கும் எண் பெறுமானங்களின் பருமனுக்கேற்ப எண்சார்ந்த தரவுகளின் தொகுதிகளை ஐந்து உப தொகுதிகளாகப் பிரிக்கலாம்.

1. Byte
2. Integer
3. Long
4. Single
5. Double

இவற்றுள் 'Byte' 'Integer' 'Long' ஆகிய தரவு வகுப்புக்களை முழு எண்களை தேக்கி வைக்கவும் 'Single' 'Double' ஆகிய தரவு வகைகளை தசமதானங்கள் கொண்ட மெய்யெண்களை தேக்கி வைக்கவும் பயன்படுத்தலாம்.

3. சரம் (String) : எழுத்துக்கள், இலக்கங்கள் மற்றும் குறியீடுகள் ஆகியவற்றைக் கொண்ட தரவுகள் இத்தொகுதியில் அடக்கப்படும். "Sri Lanka", இதற்கான உதாரணங்களாகும். இவ்வாறான தரவுகள் எப்போதும் "page 32" ஆகியன எனும் குறியீட்டினால் உள்ளக்கப்பட்டிருக்கும். எண்களை சரவகை மாறியில் தேக்கிவைக்கும்போது (String) அதனை எண்சார்ந்த தரவுகளாகப் பயன்படுத்தி செய்கைகளுக்கெனப் பயன்படுத்த முடியாது. சரவகை (String) தரவுகளுடன் மேற்கொள்ளக்கூடிய செய்கைகளுள் சரஇணைப்பு (String Concatenation) பிரதான இடம் வகிக்கின்றது. இதற்கென '&' எனும் குறியீடு பயன்படுத்தப்படும். இதன்மூலம் ஒன்றிற்கு மேற்பட்ட சரங்களை ஒன்றுடனொன்று இணைக்க முடியும்.

**உதாரணம்** : "My name is" & "Saman" எனும் கூற்றில் & செய்கையை மேற்கொள்ளும்போது "My name is Saman" எனும் சரம் (string) கிடைக்கும்.

- |                 |  |
|-----------------|--|
| <b>Currency</b> | - இது நிதிசார்ந்த பெறுமானங்களை முகாமைத்துவம் செய்வதற்கென தரப்பட்டுள்ளது. இவ்வகைப் பெறுமானங்களில் தசமதானத்திற்குப் பின்னர் நான்கு இலக்கங்களைப் பயன்படுத்த வேண்டும். |
| <b>Date</b>     | - இது திகதி, நேரம் ஆகியவற்றை முகாமைத்துவம் செய்வதற்கான தரவு வகையாகும்.   |
| <b>Variant</b>  | - மேலே கூறப்பட்ட சகல விதமான தரவுகளும் இவ்வகைக்குள் அடங்கும்.   |



## 1.7 மாறிகள் (Variables)

கணினி மொழி தொடர்பாகக் கற்பதற்கு மாறிகள் (Variables) பற்றிய அறிவு இன்றியமையாதாகும். செய்நிரலை தரவுகளுடன் செயற்படுத்தும்போது உரிய தரவுகளை பிரதான நினைவகத்தில் (RAM) தற்காலிகமாக நினைவில் வைத்துக் கொள்வதற்கென மாறிகளைப் பயன்படுத்தலாம். என முன்னர் பார்த்தோம். கணினியின் பிரதான நினைவகத்தின் குறிப்பிட்ட பகுதியினை அணுகுவதற்குப் பயன்படுத்தக்கூடிய குறியீட்டுப் பெயர்களே மாறிகள் எனப்படும். இவ்வாறு மாறிகளை நினைவகத்தின் குறிப்பிட்ட பாகத்துடன் இணைப்புச் செய்வதன் மூலம் அம்மாறிகளைப் பயன்படுத்தி அதற்குரிய நினைவகப் பகுதியில் பல்வேறு தரவுகளைக் களஞ்சியப்படுத்தவும் மீண்டும் அவற்றைப் பெற்றுக் கொள்ளவும் முடியும்.

### உதாரணம் 1

$A = 25$  எனும் கூற்றைக் கருத்திற் கொள்வோம். இவ்வாறான கூற்றொன்றை செய்நிரலில் உட்படுத்தும்போது அக்கணினிக்கு குறித்தவொரு கட்டளை வழங்கப்படும். அக்கட்டளையைச் செயற்படுத்தும்போது நடைபெறும் பணிகள் வருமாறு:

01. கணினியின் பிரதான நினைவகத்தின் ஒரு பகுதி  $A$  எனப்படும் மாறிக்கெனப் பெற்றுக் கொள்ளப்படல்.
02. இப்பகுதியுடன்  $A$  எனும் குறியீட்டுப் பெயரை தற்காலிகமாக இணைப்புச் செய்தல்
03.  $A$  என இனங்காணப்பட்ட நினைவகப் பகுதியில் 25 எனும் பெறுமானம் களஞ்சியப்படுத்தப்படல்

### உதாரணம் 2

$$\begin{aligned} A &= 10 \\ B &= 20 \\ C &= A + B \end{aligned}$$

இக்கூற்றுகளை கணினிச் செய்நிரல்களாக செயற்படுத்தும்போது நடைபெறும் செயன்முறை வருமாறு:

இலவச விநியோகத்திற்காக

01. மேலே காட்டப்பட்டவாறு கணினியின் பிரதான நினைவகத்தில் ஒரு பகுதி ஒதுக்கப்பட்டு அதனுடன் A எனும் குறியீட்டுப் பெயர் இணைப்புச் செய்யப்பட்டு அப் பகுதியில் 10 எனும் பெறுமானம் சேமிக்கப்படல்.
02. இவ்வாறே நினைவகத்தின் பிறிதொரு பகுதி ஒதுக்கப்பட்டு அதனுடன் B எனும் குறியீட்டுப் பெயர் இணைப்புச் செய்யப்பட்டு அப்பகுதியில் 20 எனும் பெறுமானம் சேமிக்கப்படல்.
03. பகுதி A இல் சேமிக்கப்பட்ட பெறுமானத்தை பகுதி B யில் சேமிக்கப்பட்டுள்ள பெறுமானத்துடன் கூட்ட வரும் மொத்தத்தை நினைவகத்தில் பிறிதொரு பகுதியில் சேமித்தலும் அப்பகுதிக்கு C எனும் குறியீட்டுப் பெயரை இணைப்புச் செய்தலும்.

மேற்கூறப்பட்ட கட்டளைகளைச் செயற்படுத்தும்போது கணினியின் பிரதான நினைவகத்தில் ஏற்படும் முக்கிய மாற்றங்கள் பின்வரும் வரிப்படத்தில் காட்டப்பட்டுள்ளன.

A	=	10	A 10
B	=	20	B 20
C	=	A+B	C 30



**மாறிகள் தொடர்பாக நினைவில் வைத்திருக்க வேண்டிய அடிப்படைத் தகவல்கள்**

1. கணினியின் பிரதான நினைவகத்தில் தரவுகளைக் களஞ்சியப் படுத்துவதற்காக ஒதுக்கப்பட்ட இடத்தின் ஒரு குறியீட்டுப் பெயராக கருதப்படலாம்.
2. மாறியுடன் இணைந்த நினைவகப் பகுதியில் பல்வேறு சந்தர்ப்பங்களில் பல்வேறு பெறுமானங்கள் களஞ்சியப்படுத்தப்படலாம். இவ்வாறு பல்வேறு சந்தர்ப்பங்களில் பல்வேறு பெறுமானங்களைக் களஞ்சியப்படுத்தும் போது

அதற்கு முன்னர் களஞ்சியப்படுத்தப்பட்ட பெறுமானங்கள் மாற்றீடு செய்யப்படும். அதற்கான உதாரணம் வருமாறு.

$$A = 10$$

$$A = 20$$

இச்செய்நிரலைச் செயற்படுத்தும்போது முதலில் நினைவகத்தில் ஒரு பகுதிக்கு A எனும் குறியீட்டுப் பெயர் இணைப்புச் செய்யப்பட்டு அப்பகுதியில் 10 எனும் பெறுமானம் களஞ்சியப்படுத்தப்படும். இரண்டாவது கட்டளையைச் செயற்படுத்தும்போது 10 எனும் பெறுமானம் 20 இனால் மாற்றீடு செய்யப்படும்.

3. நினைவகத்தின் ஒரு பகுதி மாறிகளுக்கென ஒதுக்கப்படுதல் ஒரு தற்காலிக செயன்முறையாகும். செய்நிரல் செயற்படுத்தப்பட்டு முடிந்ததும் அச்செய்நிரலில் மாறிக்கென எடுக்கப்பட்ட நினைவகத்தின் அனைத்துப் பகுதிகளையும் மீண்டும் கணினி பெற்றுக்கொள்ளும்.



### மாறிகளுக்குப் பெயரிடல் (Naming Variables)

மாறியொன்றுக்கு குறியீட்டுப் பெயரை இடும்போது கைக்கொள்ளப்பட வேண்டிய விதிமுறைகள் சில உள்ளன. இவ்விதிமுறைகள் செயல்நிரல் எழுதும்போது கடைப்பிடிக்கப்படும் இவ் விதிமுறைகள் கணினி மொழியில் தங்கியிருக்கும்.

விஷுவல்பேசிக் கில் உள்ள மாறிகளின் பெயர்கள் பின்வரும் விதிமுறைகளுக்கு அமைய இடப்பட வேண்டும்.

1. ஆங்கில அரிச்சுவடி எழுத்தைப் பயன்படுத்தி மாறியின் பெயர் ஆரம்பிக்கப்பட வேண்டும்.
2. முதலாம் எழுத்துக்குப் பின்னர் வரும் எழுத்துக்களுக்கு ஆங்கில அரிச்சுவடியிலுள்ள எழுத்துகள் அல்லது இலக்கங்கள் அல்லது “-” (Under score) ஆகிய குறியீடுகளைப் பயன்படுத்தலாம்.
3. மாறியின் பெயரில் இருக்கத்தக்க ஆகக்கூடிய எழுத்துக்களின் எண்ணிக்கை 255 ஆகும்.
4. விஷுவல்பேசிக்கிலுள்ள சிறப்புச் சொற்கள் (Reserved words) மாறிகளின் பெயர்களாக இருத்தல் கூடாது.



### செயற்பாடு 1.18

பின்வருவனவற்றில் எப்பெயர்களை 'VB' க்கான மாறிகளுக்குரிய பெயர்களாக பயன்படுத்தலாம். ஏனையவை அவ்வாறு பயன்படுத்த முடியாமைக்கான காரணம் யாதென விளக்குக.

- |            |          |
|------------|----------|
| 1) A5      | 6) Name? |
| 2) MyName  | 7) abc   |
| 3) My_Name | 8) lf    |
| 4) My Name | 9) 5A    |
| 5) _Name   | 10) A_5  |



### மாறிகளை பிரகடனப்படுத்தல்

செய்நிரலாக்கத்தின்போது தேவைப்படும் மாறிகளின் தரவு வகைகளை (Data types) நிகழ்ச்சித்திட்டத்தில் எடுத்துரைப்பதன் மூலம் வரையறுக்கமுடியும். இவ்வாறான கூற்றுக்கள் மாறிகளை பிரகடனப்படுத்தும் கூற்றுக்கள் (Variable Declarations) எனப்படும்.

'VB' மாறிகளில் தரவுவகையை எடுத்துரைப்பதற்கு 'Dim' எனும் சிறப்புச் சொல் 'As' எனும் சிறப்புச் சொல்லுடன் பயன்படுத்தப்படும்.

உதாரணம் : Dim Value1, As Boolean  
Dim Value2, As Boolean  
Dim Name As String

இவ் உதாரணத்தில் 'Value1' 'Value2' ஆகிய மாறிகளின் தரவு வகுப்பு 'Boolean' ஆகவும் 'Name' மாறி 'String' வகைக்குரிய மாறியாகவும் பிரகடனப்படுத்தப்பட்டுள்ளது.

'VB' நிகழ்ச்சித்திட்டத்தில் எடுத்துரைக்கப்படாத எல்லா மாறிகளும் 'Variant' வகை மாறிகள் எனப்படும்.





### செயற்பாடு 1.19

'VB' மாறிகளைப் பயன்படுத்த முன்னர் எடுத்துரைக்கப்படுவது அவசியமன்று. எனினும் மாறிகளின் தரவு வகுப்பை எடுத்துரைப்பதன் மூலமாக பல பயன்கள் கிட்டும். இப்பயன்கள் எவையென கலந்துரையாடுக.



### செயற்பாடு 1.20

'Numeric' மற்றும் 'String' எனும் தரவு வகைகளில் மேற்கொள்ளக் கூடிய பல்வேறு செய்கைகள் எவையெனக் கண்டறிய முயலுங்கள். இவ்வாறான செய்கைகளை 'VB' செய்நிரலாக்கத்தில் பயன்படுத்தும்போது கைக்கொள்ளப்படவேண்டிய ஒழுங்கு விதிகள் எவையென அறிந்துகொள்ளுங்கள்.



## கணித வினைக்குறிகள் (Mathematical Operators)

+	கூட்டல்	(Addition)
-	கழித்தல்	(Subtraction)
*	பெருக்கல்	(Multiplication)
/	பிரித்தல்	(Division)
^	அடுக்கேற்றம்	(Exponention)
\	முழு எண் பிரித்தல்	(Integer division)
Mod	மீதி	(Remainder)



## 1.8 செய்கை முன்நிகழ்வு (Operator Precedence)

பின்வரும் கணிதச் சார்பினை கருத்திற் கொள்க.

$$5 - 3 \times 2$$

இதனை இரண்டு முறைகளில் சுருக்க முடியும்.

$$5 - (3 \times 2) = -1 \quad (5 - 3) \times 2 = 4$$

இவ்வாறான கூற்றுகளை பல்வேறு முறைகளில் சுருக்கும்போது கிடைக்கும் பெறுபெறுகள் அநேகமாக ஒன்றுக்கொன்று வேறுபட்டனவாகும். ஆகவே, கூற்றில்

பல்வேறு கணிதச் செய்கைகள் உள்ளபோது அவை செயற்படுத்தப்பட வேண்டிய ஒழுங்கு அச்செய்கைகளின் முன்னுரிமை (முன்னிகழ்வு) அடிப்படையில் அமைந்திருக்கும்.

செய்கை	முன்னுரிமை	ஒழுங்கு
(1)		()
(2)		^
(3)		*, /
(4)		+, ., -


**உதாரணம் :**

$$5 + 2 * 3 - 5 ^ 2$$

செய்கை முன்னுரிமை (முன்னிகழ்வின்) அடிப்படையில் பின்வரும் கூற்று கருதப்பட வேண்டிய முறை அடைப்புக்குறி இடப்பட்டு காட்டப்பட்டுள்ளது.

$$\begin{aligned} & ((5 + (2 * 3) - (5 ^ 2))) \\ & \quad \uparrow \quad \uparrow \quad \uparrow \quad \uparrow \\ & \quad (3) \quad (2) \quad (4) \quad (1) \\ & = ((5 + (2 * 3) - 25)) \\ & = ((5 + 6) - 25) \\ & = 11 - 25 \\ & = -14 \end{aligned}$$

அடைப்புக்குறிகளை இடுவதன் மூலம் கூற்றிலுள்ள செய்கைகளுக்கென வழங்கப்படும் முன்னுரிமையை மாற்றியமைக்கலாம்.



**செயற்பாடு 1.21**

அடைப்புக்குறியை இடுவதனால் கூற்றில் அடங்கியுள்ள கணித செய்கையின் முன்னுரிமை (முன்னிகழ்வு) மாறுபடும் விதத்தைக் கலந்துரையாடுக.



## நிபந்தனை வினைக்குறிகள் (செயலிகள்) (Conditional Operators)

நிபந்தனை வினைக்குறிகள் கணியம் ஒன்றினை (எண்,சரம்) இன்னோர் கணியத்துடன் ஒப்பிடுவதற்கு பயன்படும். இவ்வாறான செய்கைகளை செயற்கூற்றில் செயற்படுத்தும்போது கிடைக்கும் பேறுபேறு பூலியன் மெய் அல்லது பொய் (True or False) பெறுமானங்களாக அமைந்திருக்கும்.

செய்கை	விளக்கம்
= சமன்	(Equal to)
> பெரியது	(Geater than)
< சிறியது	(Less than)
>= பெரிது அல்லது சமன்	(Greater than or equal to)
<= சிறிது அல்லது சமன்	(Less than or equal to)
<> சமனிலி	(Not equal to)

உதாரணம் :-

கூற்று

பெறுபேறு

1)  $5 = 6$

False (பொய்)

2)  $3 + 5 = 4 + 4$

True (மெய்)

3)  $5 < 6$

True (மெய்)

4)  $6 < 5$

False (பொய்)

5)  $5 < > 6$

True (மெய்)

6)  $5 < = 6$

True (மெய்)

7)  $6 < = 5$

False (பொய்)

நிபந்தனைச் செய்கைக் கூற்று ஒன்றில் நிபந்தனை வினைக்குறியின் வலது பக்கத்தில் ஒரு சார்பும் இடது பக்கத்தில் ஒரு சார்புமாக அமைந்திருக்கும். இச் சார்புகள் சிக்கலானவையாக இருக்கும் பட்சத்தில் அவை முதலில் சுருக்கப்பட்டு பின்னர் நிபந்தனை வினைக்குறி பிரயோகிக்கப்படும்.

உதாரணம் :-

$$(5 * 3) + 2 > (6 / 2) * 5$$

முதலில் இக்கூற்று பின்வருமாறு சுருக்கப்படும்.

$$17 > 15$$

17 எனும் எண் 15 இலும் பெரிதானதால் இக்கூற்றிற்குரிய பெறுபேறு மெய் (True )என கிடைக்கும்.



## தர்க்க வினைக்குறிகள் (Logical Operators)

செயற்கூற்று ஒன்றில் தர்க்க ரீதியான செய்கைகளை மேற்கொள்ள இவ்வினைக்குறிகள் பயன்படுத்தப்படும். இவ்வகைச் செய்கை தர்க்கத் (Logical) தரவு வகையில் மட்டுமே மேற்கொள்ளப்படலாம். அடிப்படைச் செய்கைகள் மூன்று வினைக்குறிகளால் மேற்கொள்ளப்படும். 'AND', 'OR' 'NOT' ஆகியவையே அவையாகும். அவற்றிற்குரிய விளக்கம் பின்வரும் உண்மை அட்டவணைகள் (Truth Table) மூலம் தரப்பட்டுள்ளது.

(பூலியன் அட்சரகணிதம் பற்றிய விளக்கம் தரம் 10 இன் நான்காம் அத்தியாயத்தில் விளக்கப்பட்டுள்ளது.)

பின்வரும் உண்மை அட்டவணையில் 'a' 'b' ஆகியன பூலியன் மாறிகளாகும்.

	a	b	a AND b
AND	True	True	True
	True	False	False
	False	True	False
	False	False	False

**உதாரணம் :-**

$(2 > 3) \text{ AND } (3 > 2)$  எனும் கூற்றைக் கருத்திற் கொள்க. தர்க்கச் செய்கையை பூலியன் வகைப் பெறுமானங்களுக்கு மட்டும் செயற்படுத்தமுடியுமாயால் முதலில் வினைக்குறியின் கூற்றின் இடது மற்றும் வலது பக்கங்களிலுள்ள சார்புகள் கருதப்பட்டு பூலியன் பெறுமானங்களைப் பெறவேண்டும்.  $2 > 3$  என்பது 'False' ஆக அமைவதுடன் அவற்றின்  $3 > 2$  'True' ஆகும். அதற்கடுத்ததாக 'False AND True' எனும் கூற்றின் பெறுபேறு மேற்கூறப்பட்ட உண்மை அட்டவணைக்கமைய 'False' ஆக அமையும். ஆகவே, மேற்கூறப்பட்ட கூற்றின் இறுதிப் பெறுமானம் 'False' எனும் பூலியன் வகைப்பெறுமானமாக அமையும்.

	a	b	a OR b
OR	True	True	True
	True	False	True
	False	True	True
	False	False	False

NOT

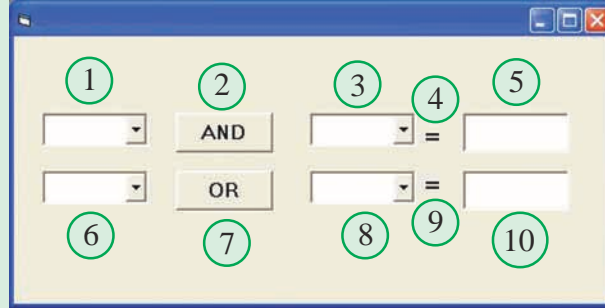
a	NOT a
True	False
False	True

மேற்கூறப்பட்ட அட்டவணையில் காட்டப்பட்டவாறு 'AND' மற்றும் 'OR' ஆகியன துவித வினைக்குறிகளாக அமைவதுடன் 'NOT' ஏக வினைக்குறியாகும்.



### செயற்பாடு 1.22

AND, OR ஆகிய பூலியன் வினைக்குறிகளின் (செயலிகளின்) தொழிற்பாட்டைக் காட்டுவதற்கென பின் வருமாறான இடைமுகத்தைக் கொண்ட நிகழ்ச்சித் திட்டத்தைத் தயாரிக்க.



Control Number	Type	Name
1	Combo Box	cboAndLeft
2	Command Button	cmdAnd
3	Combo Box	cboAndRight
4	Label	lblAndEqual
5	Text Box	txtAnd
6	Combo Box	cboOrLeft
7	Command Button	cmdOr
8	Combo Box	cboOrRight
9	Label	lblOrEqual
10	Text Box	txtOr

1,3,6,8 ஆகிய கட்டுப்படுத்திகளின் 'Data Format' இயல்பினை 'List' இயல்பாகவும் 'True' மற்றும் 'False' ஆகியன அவற்றின் பெறுமானங்களாகவும் அமைக்க 'Combo Box' இல் 'List' இயல்புக்கான பெறுமானங்கள் சிலவற்றை வழங்கும்போது ஒரு பெறுமானத்தை உள்ளிட்ட பின்னர் 'Ctrl - Enter' சாவியைச் செயற்படுத்தி அடுத்த பெறுமானத்தையும் உள்ளிட முடியும்.

மேலேயுள்ள cmdAnd மற்றும் cmdOr ஆகிய கட்டளைப் பொத்தான்களில் சொடுக்கல் இடைவினைச்செயல் (Click events) களுக்கு பின்வரும் செயல்முறைகளை உட்படுத்தி செய்நிரலை செயற்படச் செய்க.

### Program 9

```
Private Sub cmdAnd_Click()  
    Dim LeftAndValue As Boolean  
    Dim RightAnd Value As Boolean  
    Dim Result And As Boolean  
    LeftAndValue = cboAndLeft.List(cboAndLeft.ListIndex)  
    RightAndValue = cboAndRight.List(cboAndRight.ListIndex)  
    ResultAnd = LeftAndValue And RightAndValue  
    txtAnd.Text = ResultAnd  
End Sub  
Private Sub cmdOr_Click()  
    Dim LeftOrValue As Boolean  
    Dim RightOrValue As Boolean  
    Dim ResultOr As Boolean  
    LeftOrValue = cboOrLeft.List(cboOrLeft.ListIndex)  
    RightOrValue = cboOrRight.List(cboOrRight.ListIndex)  
    ResultOr = LeftOrValue Or RightOrValue  
    txtOr.Text = ResultOr  
End Sub
```

செய்நிரலைக் கட்டுப்படுத்தும் And அல்லது Or கட்டளைப் பொத்தான் மீது சொடுக்கி நடைபெறுவனவற்றையும் அவ்வாறு ஏன் நடைபெறுகின்றதெனவும் குறித்துக் கொள்க.

'Combo Boxes' இலுள்ள பெறுமானங்களை And மற்றும் Or பொத்தான்களின் மீது சொடுக்கும்போது கிடைக்கும் விடைகளை குறித்துக் காட்டுக.

இதன்போது பின்வரும் செயல்முறைப் பகுதியைச் சேர்த்து மீண்டும் செய்நிரலை மேலே கூறப்பட்டவாறு செயற்படுத்தி ஏற்படும் மாற்றங்களை குறித்துக் கொள்க.

```
Private Sub Form_Load()  
    cboAndLeft.ListIndex = 0  
    cboAndRight.ListIndex = 0  
    cboOrLeft.ListIndex = 0  
    cboOrRight.ListIndex = 0  
End Sub
```



### பாய்ச்சற் கட்டுப்பாடு (Flow Control)

செய்நிரலிலுள்ள கூற்றுக்களைச் செயற்படுத்தும் ஒழுங்குமுறையினை பாய்ச்சற் படங்கள் (Flow Chart) மூலம் வெளிப்படுத்தலாம். பொதுவாக செய்நிரல் ஒன்றிலுள்ள கூற்றுக்களில் முதலாவது கூற்றிலிருந்து ஆரம்பித்து ஒன்றின் பின் ஒன்றாக செய்நிரல் கூற்றுக்கள் ஒழுங்குமுறையில் செயற்படுத்தப்படும். இவ் ஒழுங்குமுறையை மாற்றியமைப்பதற்கென பல்வேறு கட்டுப்பாட்டு அமைப்புக்கள் (Control Structures) பயன்படுத்தப்படும்.

இக் கட்டுப்பாட்டு அமைப்புகளின் பிரயோகத்தின்போது அவை தொடர்பாக சில விதிகள் கைக்கொள்ளப்பட வேண்டியது அவசியமாகும். இவ்விதிகள் அக்கட்டமைப்புக்களின் தொடரியல் (Syntax) எனப்படும்.





## IF...Then.....Else கட்டமைப்பு

IF Condition THEN

VB Statements

1

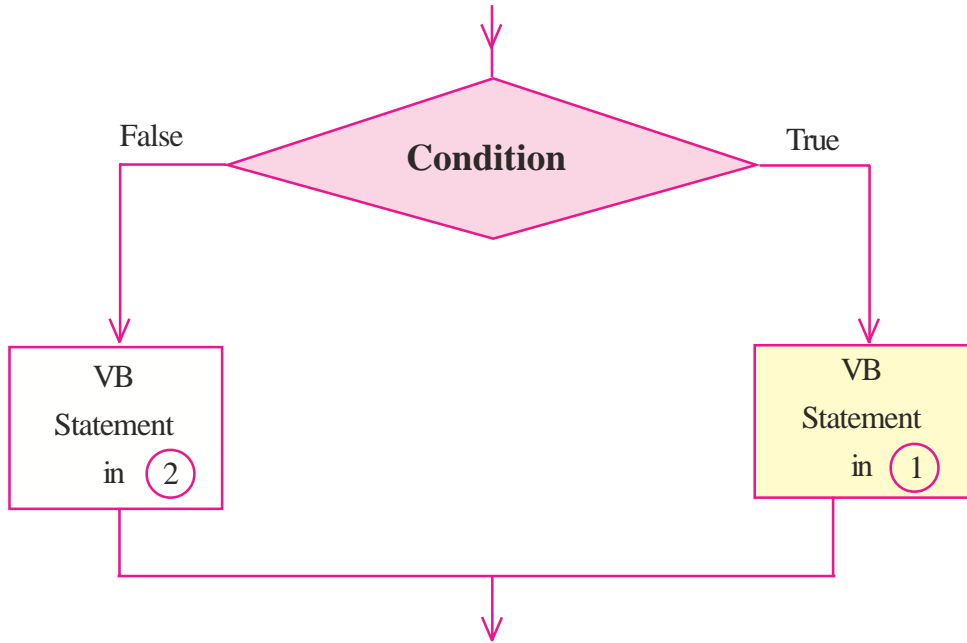
Else

VB Statements

2

End IF

பாய்ச்சற் படம்



இம் மாதிரியக்கத்துக்காக பயன்படுத்தப்படும் If, Then, Else, End If ஆகிய சொற்கள் 'VB', இற்குரிய சிறப்புச் சொற்களாகும்.

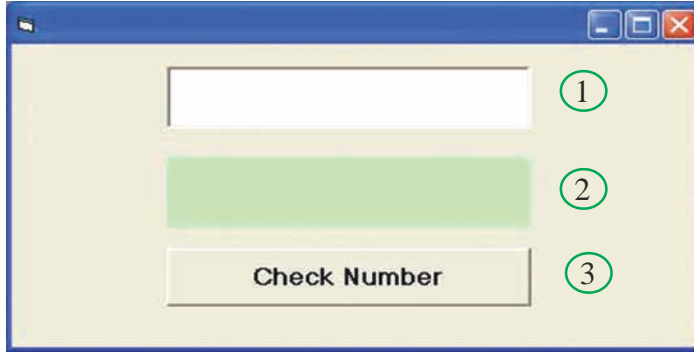
'Condition' என தரப்பட்டுள்ள 'VB' கூற்றின் இறுதிப் பெறுபேறு பூலியன் பெறுமானமாக இருக்க வேண்டும்.

'VB' Statements எனப்படுவது 'VB' கூற்றுகளின் தொகுப்பாகும்.

இந்த அமைப்பின் தொழிற்பாட்டைப் பின்வருமாறு காட்டலாம்.

- 'Condition' கூற்றினைச் சுருக்கி இறுதி பூலியன் பெறுமானமாகப் பெற்றுக் கொள்க.
- இந்தப் பெறுமானம் 'True' எனில் பகுதி 1 இல் குறிப்பிடப்பட்டுள்ள 'VB'கூற்றுகளை இதில் தரப்பட்டுள்ள ஒழுங்கில் செயற்படுத்துங்கள்.
- 'Condition' கூற்றின் பெறுமானங்கள் 'False' எனின் பகுதி 2 இல் தரப்பட்டுள்ள கூற்றுக்களை அதில் தரப்பட்டுள்ள ஒழுங்கில் செயற்படுத்துக.

**Program - 10 :-** எண்ணானது ஒன்றையெண்ணா இரட்டையெண்ணா என அறிந்து கொள்ள பொருத்தமான செய்நிரலை அமைக்குக.



Control Number	Type	Name
1.	Text Box	txtNo
2.	Label	lblDisplay
3.	Command Button	cmdCheckNo

```
Private Sub cmdCheckNo_Click()
```

```
    Dim No As Integer
```

```
    No = Val(txtNo.Text)
```

```
    If (No Mod 2) = 0 Then
```

```
lblDisplay.Caption = "Even Number"
```

```
Else
```

```
lblDisplay.Caption = "Odd Number"
```

```
End If
```

```
End Sub
```



## 1.10 Select.. Case

இக்கட்டமைப்பை IF...Then...Else ஆகிய கட்டளைகளின் தொகுப்பாகும்.

Select Case expression

Case Value 1

a

VB Statements

Case Value 2

1

VB Statements

2

Case Else

VB Statements

End Select

n

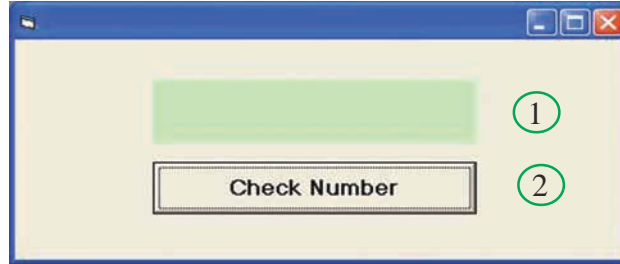
செயற்கூற்று

இங்கு 'Select Case', 'Case', 'Case Else', 'End Select' ஆகியன சிறப்புச் சொற்களாகும். 'Expression' இன் தரவு வகை (Data type) 'Value1', 'Value2'... தரவு வகைகளுடன் ஒத்திருக்கவேண்டும். இங்கு 'a' பகுதியில் குறிப்பிடப்பட்டுள்ள சார்பினை சுருக்கும்போது கிடைக்கப்பெறும் பெறுமானத்தை 'Value 1' உடன் ஒப்பிட்டு நோக்கலாம். அது சமனாகும் போது 1 வரையான பகுதி செயற்படும். இல்லாவிடின் அது 'Value 2', 'Value 3'..... ஆகியவற்றின் ஒழுங்கு முறையில்

ஓப்பிட்டு நோக்கப்படும். ஏதேனுமொரு 'Value' இன் பெறுமானம் சார்பின் ('ex-pression') இன் இறுதிப்பெறுமானத்துக்குச் சமனாகும் போது அவ் 'Value' பகுதிக்கு கீழே உள்ள 'VB' செயற்கூற்றுகள் செயற்படுத்தப்படும்.

சார்பின் பெறுமானமானது, ஏதேனுமொரு 'Value 1', 'Value2', ..... பெறுமானத்துக்கு சமனாக அமையாதவிடத்து 'Case Else' பகுதிக்கு கீழே 'n' என குறிப்பிடப்பட்டுள்ள பகுதியிலுள்ள 'VB' கூற்று செயற்படும்.

**Program 11** - எண்ணானது நேர் பெறுமானமா, மறைப்பெறுமானமா அல்லது பூச்சியமா என அறிந்து கொள்ள செய்நிரலொன்றை தயாரிப்போம்.



Control Number	Type	Name
1.	Label	lblDisplay
2.	Command Button	cmdCheckNo

```

Private Sub cmdCheckNo_Click()
    Dim N As Single
    N = InputBox("Enter Number")
    Select Case N
        Case Is < 0
            lblDisplay.Caption = "Negative Number"
        Case Is > 0
            lblDisplay.Caption = "Positive Number"
        Case Else
            lblDisplay.Caption = "Zero"
        End Select
    End Sub

```



## செயற்பாடு 1.24

Select .... Case இன் செயற்கூற்றைக் காட்டுவதற்கு பாய்ச்சற் கோட்டுப் படம் வரைக.



## 1.10 தடமாக்கல் (Looping)

செய்நிரலிலுள்ள அடுத்தடுத்த கூற்றுக்கள் சிலவற்றினை மீண்டும் மீண்டும் செயற்படுத்த பயன்படுத்தப்படும் அமைப்புகள் (Looping Structures) தடமாக்கல் கட்டுமானத்தில் அடங்கும்.

### For ... Next தடமாக்கல் மாதிரி

*For* மாறி = ஆரம்பபெறுமானம் *To* இறுதிப் பெறுமானம் (*Step* குறைக்கும் அல்லது கூட்டும் பெறுமானம்)

*உரிய குறிமுறைப்பகுதி*

*Next* மாறி

இதிலுள்ள [*step* குறைக்கும் அல்லது கூட்டும் பெறுமானம்] ஆகிய பகுதிகள் தேவையான சந்தர்ப்பங்களில் மட்டும் பயன்படுத்தக்கூடிய மாற்றீடாகும்.

### For ... Next தடமாக்கல் தொழிற்படும் விதம்

For ... Next தடமாக்கலில் முதலில் அதில் குறிக்கப்பட்டுள்ள மாறிக்குரிய பெறுமானம் பிரதியிடப்படும். இதன்போது முதலாம் பெறுமானத்திலிருந்து இறுதிப் பெறுமானம் வரை அதிகரித்துச் செல்லும் ஒழுங்கில் அவை பிரதியிடப்படும். Step உடன் சேர்த்து குறிக்கப்பட்ட பெறுமானமொன்று வழங்கப்பட்டிருப்பின் அடுத்தடுத்த இரண்டு பெறுமானங்களுக்கிடையிலான வித்தியாசத்தை சீர்செய்வதற்கு இப் பெறுமானம் பயன்படுத்தப்படும். அதாவது பொதுவாக For... Next தடமாக்கலின்போது பெறுமான அதிகரிப்பொன்று ஏறுவசரிசையில் ஒன்றுக்கொன்று அதிகரித்துக் காணப்படும். பெறுமானங்களின் பெறுமதியை மாறுபட்ட விதத்தில் அதிகரிப்பதற்கு அல்லது குறைப்பதற்கென step மாற்றீட்டைப் பயன்படுத்தலாம். இறங்குவரிசையில் தடமாக்கலை மேற்கொள்ளும்போது இந்த மறைப் பெறுமானம் பயன்படுத்தப்பட வேண்டும்.

மாறிக்கென பிரதியிடப்பட்ட பெறுமானத்துடன் முன்னோக்கி செல்லும் தடமாக்கலில் அதிலுள்ள கூற்றுக்கள் செயற்படுத்தப்படுவதுடன் Next என்னும் சொல் / வரிசை அண்மித்ததும் மீண்டும் For சொல் உள்ள வரிசையை நோக்கி செல்லும். அதன்போது உரிய மாறியின் பெறுமானம் இறுதிப் பெறுமானமா என பரீட்சப்படுத்தப்படுவதுடன் அது இறுதிப் பெறுமானமாயின் தடமாக்கல் பூர்த்தி செய்யப்பட்டு Next சொல் / வரிசைக்கு பின்னர் உள்ள கூற்றுக்கள் செயற்படுத்தப்படும். அவ்வாறு நிகழாதவிடத்து மாறியின் பெறுமானத்தை உரியவாறு மாற்றியமைத்து தடமாக்கலினுள் மீண்டும் செல்லும். அதாவது மீண்டும் தடமாக்கலிலுள்ள கூற்றுக்களை செயற்படுத்தி முன்னோக்கி செல்லும். Next சொல்லினை எட்டியதும் மீண்டும் For வரிசைக்கு சென்று மாறியின் பெறுமானம் பரிசீலிக்கப்படும். இவ்வாறான செயற்பாடு மாறியின் பெறுமானம் இறுதிப் பெறுமானம் ஆகும் வரை நடைபெறும்.

குறித்த தடவைகள் குறிப்பிட்ட கூற்றுக்களை செயற்படுத்துவதற்கு For .... Next தடமாக்கல் ஒரு பொருத்தமான கட்டமைப்பாகும்.

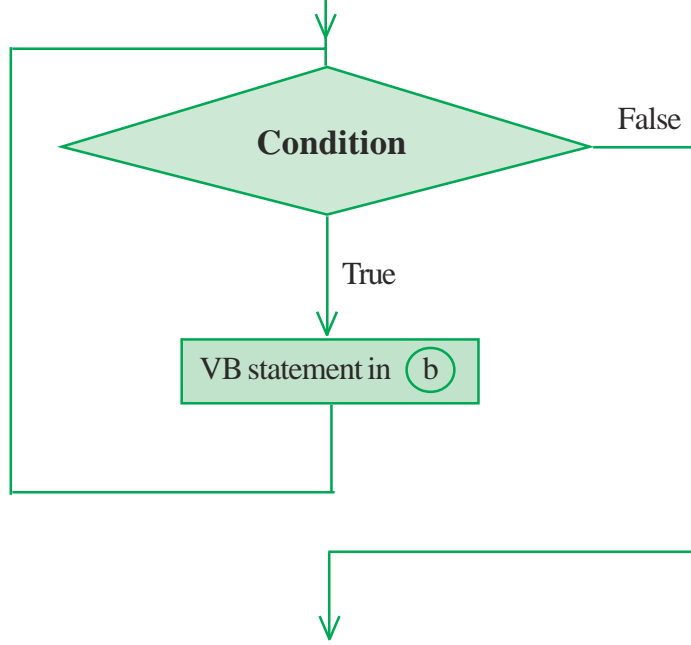
**உதாரணம் :-**

```
a) Sum=0
   For i= 0 to 10
       Sum= Sum + i
       Print i
   Next
   Print Sum
```


இங்கு 1 இலிருந்து 10 வரை பெறுமானங்கள் ஒன்றிலிருந்து மற்றையது அதிகரிக்கத் தக்கதாக (ஒழுங்கில்) படிவத்தில் கீழ்நோக்கிப் பதிப்புச் செய்யப்படுவதுடன் இறுதியில் அப்பெறுமானங்களின் கூட்டுத் தொகையாகிய 55 என்னும் பெறுமானமும் பதிப்பு செய்யப்படுவதைக் காணலாம்.







இங்கு 'Do,'while', 'மற்றும்' Loop' எனும் சொற்கள் சிறப்புச் சொற்களாகும். 'Condition' எனும் சார்பு பூலியன் பெறுமானம் ஒன்றினை தரக்கூடியதாக இருக்கும்.



**செயற்பாடு**  
1.24

புதிய செயற்றிட்டமொன்றை ஆரம்பித்து படிவமொன்றில் கட்டளைப் பொத்தானொன்றை இணைத்து Click event இற்கு பின் வரும் செயற் கூற்றுக்களை இட்டு அதனைச் செயற்படுத்தும்போது கிடைக்கும் பெறுமானத்தை அவதானிக்குக.

**உதாரணம்**

```

counter= 1
Do while counter<100
    Debug.print counter
    counter=counter+1
Loop
          
```

உரிய கட்டளைப் பொத்தான் மீது (Command button) சொடுக்கும் போது அதன் பெறுபேறு Immediate Window மீது வெளிக்காட்டப்படும். பின்வரும் செயற்பாடுகளின்போது ஏற்படும் மாற்றத்தை அவதானிக்குக.

- debug.print counter என்பதை print counter ஆக மாற்றவும்
- debug.print counter  
counter = counter + 1 என்பதை  
counter = counter + 1 debug.print counter  
ஆக மாற்றவும்



## 1.11 அணிகள் (Arrays)

'VB' கணினி மொழியிலும் ஏனைய கணினி மொழிகளைப் போன்று தரவுகளை முறைவழியாக்கம் செய்யும்போது அத்தரவுகளை தற்காலிகமாக சேமிக்கக்கூடிய முறை உள்ளடக்கப்பட்டுள்ளது. இவ்வாறு தற்காலிகமாக செய்நிரலில் தரவுகளை முறைவழியாக்க தயார் செய்யப்படும் தரவுக் கட்டமைப்புகள் 'Data Structures' என அழைக்கப்படும். அணி (Arrays), அடுக்கு (Stacks), மற்றும் (Queues) சரரை ஆகியன இதற்கான உதாரணங்களாகும்.

'VB' யின் அணிகள் (Array) எனப்படுகின்ற தரவுக்கட்டமைப்பானது ஒரே வகையான தரவுகளை சேமிக்க பயன்படுத்தப்படலாம். இவ்வாறான தரவுக் கட்டமைப்புகள் பின்வருமாறு காட்சிப்படுத்தலாம்.

20	30	40	55	60	Marks
1	2	3	4	5	

மேற்கூறப்பட்ட தரவுக் கட்டமைப்பு ஒருதொடை எண்களை தற்காலிகமாக சேமிப்பதற்கேற்ற ஒரு அணியாகும்.

இவ்வாறான தரவுக்கட்டமைப்புகளை செய்நிரலில் அமைக்கும்போது முழு தரவுக் கட்டமைப்பையும் குறிப்பிடுவதற்காக குறிப்பிட்ட பெயரொன்று வழங்கப்பட வேண்டும். உதாரணத்தில் அணிக்கென தரப்பட்டுள்ள பெயர் 'Marks' என்ற மாறியாகும். பின் அத்தரவுக் கட்டமைப்பிலுள்ள ஒவ்வொரு பெறுமானத்தையும் பெறுவதற்கு அல்லது களஞ்சியப்படுத்த 'Marks (Position)' எனும் கட்டமைப்பு பயன்படுத்தப்படும்.

இந்த கட்டமைப்பின் முதலாவது பகுதியில் அணிமாறியின் பெயர் (Marks) குறிப்பிடப்படும் இரண்டாவது பகுதியில் உள்ள வாதத்தின் மூலம் (Position) அணியின் முதலாவது பெறுமானத்துக்கு சார்பாக மற்றைய பெறுமானங்களின் இடம் குறிப்பிடப்படும்.

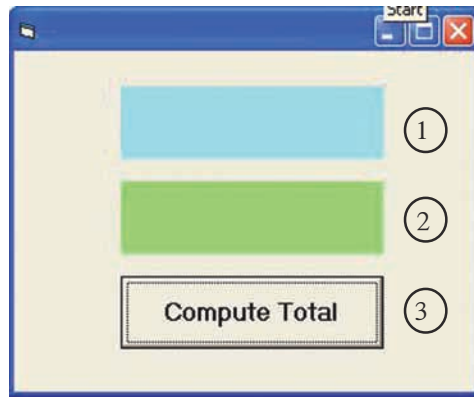
### உதாரணம் :

Dim Marks (0 to 9) As Integer

மேற்கூறப்பட்ட உதாரணத்தின் மூலம் 'Marks' எனப்படும் அணி பிரகடனப்படுத்தப்பட்டிருக்கும். இவ்வணியின் முதற் பெறுமானம் 'Marks (0)' என்பதாலும் முடிவுப் பெறுமானம் 'Marks (9)' என்பதாலும் குறிப்பிடப்படும் ஆதலால் இவ் அணியில் 10 பெறுமானங்களை  $[(9-0) + 1 = 10]$  சேமிக்கலாம்.

உதாரணமாக எந்த 10 எண்களினதும் கூட்டுத்தொகையை அறிவதற்கான செயற்றிட்டம் ஒன்றினை தயாரித்துக் கொள்ளலாம். செயற்றிட்டத்தின் இடைமுகம் பின்வருமாறு தயாரிக்கப்பட்டுள்ளது.

இவ் இடைமுகத்தில் 1, 2 ஆகிய எண்கள் முகப்பு அடையாளங்களினையும் (label) எண் 3, கட்டளைப் பொத்தான் (command button) ஒன்றினையும் குறித்துக் காட்டுகின்றன. இங்கு முதலாம் கட்டத்தில் உள்ளீடு பெறுமானம் எடுத்துக் காட்டப்படும். இரண்டாவது கட்டத்தினுள் பெறுமானங்களின் கூட்டுத்தொகை எடுத்துக் காட்டப்படும் இலக்கம் 3 இனால் குறிப்பிடப்பட்டுள்ளது செயன்முறையை ஆரம்பிப்பதற்கான பொத்தானாகும். பின்வரும் செய்நிரல் அதனுடன் இணைக்கப்பட்டுள்ளது. இந்த 1, 2, 3 ஆகிய கட்டுப்படுத்திகளின் வகை மற்றும் பெயர் என்பன அடுத்துள்ள அட்டவணியில் தரப்பட்டுள்ளது.



Control Number	Type	Name
1.	Label	lblNumbers
2.	Label	lblTotal
3.	Command Button	cmdTotal

```

Private Sub cmdTotal_click()
    Dim myNumbers (1 to 10) As Integer
    Dim i, total As Integer
    Dim numbers As String

    'No numbers entered yet
    numbers=""

    'Read the 10 numbers to no Array
    For i = 1 to 10
        myNumbers (i) = val (InputBox ("Enter the number" & i))
        lblNumbers.Caption = lblNumbers. Caption & myNumbers (i) & ","
    Next i

    'Compute the total of Numbers in the Array total = 0
    For i = 2 to 10
        total = total +mynumbers (i)
    Next i

    lblTotal caption = "Total of the numbers is" & total
End Sub

```