01

ගැටලු විසඳීම සඳහා පරිගණක භාවිතය

මෙම පාඩම හැදෑරීමෙන් ඔබට,

- පරිගණක භාෂාවන්හි පරිණාමය
- පරිගණක අනුසාරයෙන් ගැටලු විසඳීම
- ඇල්ගොරිතම නිරූපණය
- විෂුවල් බේසික් භාවිතයෙන් වැඩසටහන් සම්පාදනය කිරීම

පිළිබඳ ව මනා අවබෝධයක් ලබා ගැනීමට හැකි වනු ඇත.

පරිගණකය අපට නැති ව ම බැරි මෙවලමක් වෙමින් පවතී. පරිගණක අනුසාරයෙන් කළ හැකි කාර්යයන් රාශියකි. මෙම පරිච්ඡේදයේ දී අපගේ අවධානය යොමුවන්නේ ගැටලු විසඳීම සඳහා පරිගණකය මෙවලමක් ලෙස භාවිත කළ හැකි ආකාරයයි.

පරිගණකය අනුසාරයෙන් ගැටලුවක් විසඳීමේ දී ගැටලුව විසඳිය යුතු ආකාරය පරිගණකයට තේරුම් ගත හැකි භාෂාවකින්, විධාන සමූහයක එකතුවක් ලෙස පරිගණකගත කළ යුතු වේ. මෙවැනි පරිගණකයකට සෘජු ව ම තේරුම් ගත හැකි භාෂා 'යන්ත්‍ර භාෂා' (Machine Languages) ලෙස හැඳින්වෙයි. සාමාන්‍යයෙන් යන්ත්‍ර භාෂාවල අඩංගු විධාන හා එම විධානවල ස්වරූපය පරිගණකයේ සකසනයේ (Processor) වර්ගය මත රඳා පවතී. එනම් විවිධ වර්ගවල සකසනයන්ට තේරුම් ගත හැකි භාෂාවන්හි ස්වරූපය එකිනෙකට වෙනස් වෙයි.

උදාහරණයක් ලෙස 'Intel' පවුලේ සකසනයන් සඳහා නිර්මාණය කර ඇති යන්ත්‍ර භාෂා 'SPARC' පවුලේ සකසනයන්ගේ යන්ත්‍ර භාෂාවලට වඩා වෙනස් වෙයි. එනමුත් බොහෝ දුරට සමාන වූ යන්ත්‍ර භාෂා ක්‍රියාකරවීම සඳහා නිර්මාණය වූ විවිධ සකසන වර්ග ද අද වෙළෙඳපොළේ දක්නට ලැබේ. උදහරණයක් ලෙස 'Pentium' සහ 'AMD' සකසන වර්ග දැක්විය හැකි ය.

විවිධ සකසන වර්ග සඳහා විවිධ යන්ත්‍ර භාෂා හඳුන්වා දී ඇතත් ඒ සියලු ම යන්ත්‍ර භාෂාවලට පොදු වූ එක් සුවිශේෂ ලක්ෂණයක් ඇත. එනම් සියලු ම යන්ත්‍ර භාෂාවල සියලු විධාන 0 හා 1 යන සංකේතයන්ගෙන් පමණක් සමන්විත වීම යි. පරිගණක වැනි ඉලෙක්ට්‍රොනික යන්ත්‍රවලට සංකේත දෙකකට පමණක් සීමා වූ ද්විමය භාෂාවන් (Binary Language) තේරුම් ගැනීමට හැකි වීම මෙයට මූලික හේතුව ලෙස දැක්විය හැකි ය. මෙවැනි ද්විමය කේතයන් මත පදනම් වූ පරිගණක භාෂා පහළ මට්ටමේ භාෂා (Low Level Languages) ලෙස ද හැඳින්වෙයි.

මෙවැනි පහළ මට්ටමේ භාෂා අපට එදිනෙදා පරිගණක ක්‍රියා කරවීම සඳහා භාවිතයට ඉතා අසීරු නමුත් පරිගණකයට සෘජු ව ම තේරුම් ගත හැකි කාර්යක්ෂම භාෂා විශේෂයක් ලෙස හැඳින්විය හැකි ය. මෙම හේතුව නිසා යන්ත්‍ර භාෂා නැතහොත් පහළ මට්ටමේ භාෂා, පරිගණකයට ඉතා සමීප, මිනිසාට දුරස් පරිගණක භාෂා විශේෂයක් ලෙස සලකනු ලැබේ.

යන්ත්‍ර භාෂා භාවිත කර විධාන දීම ඉතා අසීරු බැවින්, මේ සඳහා අපට සමීප, පහසුවෙන් හැසිරවිය හැකි පරිගණක භාෂා විශාල සංඛ්‍යාවක් නිර්මාණය කර ඇත. මෙම භාෂා ඉහළ මට්ටමේ පරිගණක භාෂා (High Level Languages) ලෙස හැඳින්වෙයි. C, 'VB', හා 'PHP' මේ සඳහා නිදසුන් කිහිපයකි.

කිසි යම් කාර්යයක් පරිගණකයෙන් කර ගැනීම සඳහා පරිගණක භාෂාවක් අනුසාරයෙන් ගොනු කළ උපදෙස් මාලාවක් පරිගණක ක්‍රම ලේඛයක් (Computer Programme) ලෙස හැඳින්වෙයි. මෙම ප්‍රකාශයන්ගෙන් සමහරක් ක්‍රියාකරවිය හැකි විධාන වන අතර අනෙක් ප්‍රකාශ විධානයන් නො වේ. මෙවැනි විධානයන් නොවන ප්‍රකාශන මගින් ක්‍රමලේඛ පහසුවෙන් තේරුම් ගැනීමට අවශ්‍ය තොරතුරු (උදාහරණ : විවරණ Comments) මෙන් ම ක්‍රමලේඛයක් යන්ත්‍ර භාෂාවකට පරිවර්තනය කිරීමට අවශ්‍ය තොරතුරු සපයා දෙනු ලබයි. මෙවැනි විධාන නොවන ප්‍රකාශනයන් පිළිබඳ මූලික හැඳින්වීමක් පසු ව දක්වා ඇත.

C, VB, PHP වැනි පරිගණක භාෂා අපට පහසුවෙන් තේරුම් ගත හැකි වුවත් මෙම භාෂා සෘජු ව ම පරිගණකයන්ට තේරුම් ගත නො හැකි ය. එබැවින් මෙම ඉහළ මට්ටමේ භාෂා මගින් තැනූ ක්‍රම ලේඛයන් පරිගණකය ක්‍රියාකරවීමට පෙර පරිගණකයට තේරුම් ගත හැකි පහළ මට්ටමේ භාෂාවකට පරිවර්තනය කළ යුතු වෙයි. මෙම ක්‍රියාවලිය භාෂා පරිවර්තනය (Language translation) හෝ සම්පාදනය (Compilation) ලෙස හැඳින්වෙන අතර මේ සඳහා ගත හැකි විවිධ ක්‍රමවේදයන් පසු ව සලකා බැලේ.

1.1 පරිගණක භාෂාවන්ගේ පරිණාමය

ආරම්භයේ සිට අද දක්වා බිහිවී ඇති සියලු ම පරිගණක භාෂාවන් ඒවායේ විධානයන්ගේ ස්වරූපය අනුව ප්‍රධාන කාණ්ඩ හතරකට වර්ග කළ හැකි ය.

1. පළමු වන පරම්පරාවේ පරිගණක භාෂා හෝ
   යන්ත්‍ර භාෂා (Machine Languages)

2. දෙවන පරම්පරාවේ පරිගණක භාෂා හෝ
   එසෙම්බ්ලි භාෂා (Assembly Languages)

3. තුන් වන පරම්පරාවේ පරිගණක භාෂා හෝ
   ඉහළ මට්ටමේ භාෂා (High Level Languages)

4. හතර වන පරම්පරාවේ පරිගණක භාෂා හෝ
   නිර්මාපිත භාෂා (Artificial Languages)

පරිගණක භාෂාවන්ගේ විකාශනයේ දී දැකිය හැකි මූලික ලක්ෂණයක් වන්නේ එම භාෂා ක්‍රමයෙන් අපට පහසුවෙන් තේරුම් ගත හැකි සහ භාවිතයට පහසු භාෂා ලෙස විකාශනය වීමයි.

පළමු පරම්පරාවේ පරිගණක භාෂා

සියලු ම යන්ත්‍ර භාෂා මෙම වර්ගයට අයත් වෙයි.

මෙම භාෂාවන්ට පොදු ලක්ෂණ

- සියලු විධාන 0 හා 1 යන ද්විමය කේත භාවිතයෙන් ප්‍රකාශ කළ යුතුයි.
- වැඩසටහන් සම්පාදනය හා විධාන පරිගණකයට ඇතුළත් කිරීම ඉතා ම සංකීර්ණ කාර්යයකි.
- යන්ත්‍රයේ වර්ගය මත යැපෙන භාෂාවකි. එනම් එක් වර්ගයක සකසන සඳහා සම්පාදනය කළ ක්‍රමලේඛයන් වෙනත් සකසන වර්ගයක් මත ක්‍රියාත්මක කළ නොහැකි ය.
- පරිගණකයේ විවිධ දෘඪාංග පිළිබඳ හොඳ දැනීමක් තිබිය යුතු ය.

දෙවන පරම්පරාවේ පරිගණක භාෂා

මෙම වර්ගයේ භාෂා 0 හා 1 යන ද්විමය සංකේත පමණක් භාවිතයෙන් ක්‍රමලේඛ ගොඩනැගීමට ඇති අපහසුතාව මගහැරීම සඳහා බිහිවූ භාෂා විශේෂයකි.

මෙම වර්ගයේ භාෂාවන්ට පොදු ලක්ෂණ ඃ

- ❖ විධාන 'ADD', 'SUB' වැනි සරල සංකේත නාමයන් මගින් දැක්වීම.
- ❖ සරල සංකේත නාම භාවිතයෙන් පරිගණක මතකයේ දත්ත ගබඩා කිරීමට හා නැවත ලබාගැනීමට හැකි වීම.
- ❖ ක්‍රමලේඛයන් නිදෙස් කිරීම (Debugging) යන්ත්‍ර භාෂා ක්‍රමලේඛ නිදෙස් කිරීමට වඩා පහසු වීම.
- ❖ යන්ත්‍ර මත යැපෙන භාෂාවන් වීම.
- ❖ ක්‍රමලේඛ ලිවීම සඳහා දෘඩාංග පිළිබඳ හොඳ දැනුමක් අවශ්‍ය වීම

මෙම වර්ගයේ භාෂාවන්ගෙන් සම්පාදිත ක්‍රමලේඛ පරිගණකයනට ඍජු ව ම ක්‍රියාකරවීමට නො හැකි ය. ඒ සඳහා 'Assembler' නම් වූ භාෂා පරිවර්තකය මගින් එම ක්‍රමලේඛ හෝ විධාන අදාළ යන්ත්‍ර භාෂාවට පරිවර්තනය කළ යුතු වෙයි.

තුන් වන පරම්පරාවේ පරිගණක භාෂා

වර්තමානයේ භාවිත වන C, VB වැනි පරිගණක භාෂා මේ ගණයට අයත් වෙයි. මෙම භාෂාවන්ගේ මූලික ලක්ෂණ ඃ

- ❖ ඉංග්‍රීසි වැනි අපට සමීප භාෂාවන් මත පදනම් වී තිබීම.
- ❖ යන්ත්‍ර මත යැපෙන භාෂාවන් නොවීම.
- ❖ ක්‍රමලේඛ ලිවීමට හා ඒවා නිදෙස් කිරීමට පහසු වීම.

මෙම භාෂා ද එසෙම්බ්ලි භාෂා මෙන් ම පරිගණකයනට ඍජු ව ම තේරුම් ගැනීමට නො හැකි ය. එබැවින් භාෂා පරිවර්තක මෘදුකාංගයක් මගින් මෙම භාෂාවන්ගෙන් ලියැවුණු ක්‍රමලේඛ අදාළ පරිගණකයේ යන්ත්‍ර භාෂාවලට පරිවර්තනය කළ යුතු ය.

හතර වන පරම්පරාවේ පරිගණක භාෂා

බොහෝ අවස්ථා වලදී හතර වන පරම්පරාවේ පරිගණක භාෂා දෘශ්‍ය පරිස්ථිතියක් (Visual Environment) ලබා දෙමින් ඉතා පහසුවෙන් පරිගණක වැඩ සටහන් සකස් කිරීමට අවස්ථාව ලබා දේ.

මෙම වර්ගයේ භාෂාවන්ගේ මූලික ලක්ෂණ ඃ

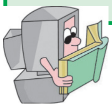
- ❖ ඉංග්‍රීසි වැනි මානව භාෂාවන්ට ඉතා සමීප වීම.
- ❖ කෙටි කාලයකින් ඉගෙන ගැනීමට හැකි වීම.

❖ අවශ්‍ය කරන විධානයන් තුන්වැනි පරම්පරාවේ පරිගණක භාෂා වලට වඩා අඩු විධාන සංඛ්‍යාවකින් කර ගැනීමට හැකි වීම.

ක්‍රියාකාරකම 1.1

(1) යන්ත්‍ර භාෂා හා එසෙම්බ්ලි භාෂා අතර ඇති මූලික වෙනස්කම් ලැයිස්තුගත කරන්න.

(2) එසෙම්බ්ලි භාෂා හා ඉහළ මට්ටමේ භාෂා අතර ඇති මූලික වෙනස්කම් ලැයිස්තුගත කරන්න.



පරිගණක භාෂා පරිවර්තනය (Programming Language Translation)

පරිගණකයට ඍජු ව ම ක්‍රියාකරවීමට හැකිවන්නේ 0 හා 1 යන ද්විමය කේතයන් මත පදනම් වූ යාන්ත්‍රික භාෂාවක් පමණක් බව අපි හඳුනා ගතිමු. එබැවින් යාන්ත්‍රික නොවන එසෙම්බ්ලි වැනි දෙවන හා තුන් වන පරම්පරාවේ භාෂාවන්ගෙන් ලියවුණු ක්‍රමලේඛ හෝ විධාන පරිගණකයක් තුළ ක්‍රියාත්මක කිරීමට පෙර පරිගණකයට තේරුම් ගත හැකි යාන්ත්‍රික භාෂාව බවට පරිවර්තනය කළ යුතු ය. මෙසේ එක් පරිගණක භාෂාවක් තවත් භාෂාවකට පරිවර්තනය කිරීමේ ක්‍රියාව පරිගණක භාෂා පරිවර්තනය (Translation) ලෙස හැඳින්වෙයි. මෙම කාර්යය සඳහා භාවිත කළ හැකි ප්‍රධාන ක්‍රමවේද තුනකි.

(1) එසෙම්බ්ලර (Assemblers) භාවිතය

(2) සම්පාදක (Compilers) භාවිතය

(3) අර්ථ විනයාසක (Interpriters)

එසෙම්බ්ලර (Assembler)

Assembly භාෂාවෙන් සකස් කරන ලද පරිගණක වැඩසටහනක් පරිගණකයට තේරුම්ගත හැකි යාන්ත්‍රික භාෂාවේ (Machine Language) කේත බවට පරිවර්තනය කරන වැඩසටහනකි.

සම්පාදක (Compilers)

සම්පාදක වර්ගයේ මෘදුකාංග මගින් ක්‍රමලේඛය පරිගණකයට පහසුවෙන් තේරුම් ගෙන ක්‍රියාත්මක කළ හැකි වස්තු කේත ගොනුවක් (Object Code) බවට පත් කිරීම සිදු කරයි. මෙසේ ක්‍රමලේඛය යාන්ත්‍රික භාෂාවට හැරවූ පසු එම යාන්ත්‍රික භාෂාවෙන් වූ ක්‍රමලේඛය පරිගණකය තුළ ක්‍රියාත්මක වේ. C, FORTRAN පරිගණක භාෂා සම්පාදක මේ සඳහා නිදසුන් කිහිපයකි.

සම්පාදක මගින් භාෂා පරිවර්තනයේ දී පරිවර්තනය සඳහා යොදගන්නා මුල් ක්‍රමලේඛය ප්‍රභව කේත ක්‍රමලේඛය (Source Code Program) ලෙස ද පරිවර්තනයෙන් පසු ලබාගන්නා ක්‍රමලේඛය වස්තු කේත ක්‍රමලේඛය (Object Code Program) ලෙස ද හැඳින්වෙයි. භාෂා සම්පාදකයක් මගින් සිදුවන ප්‍රධාන ක්‍රියාවලිය වන්නේ ප්‍රභව ක්‍රමලේඛය විෂය ක්‍රමලේඛයට පරිවර්තනය කිරීමයි. භාෂා පරිවර්තන ක්‍රියාවලියේ දී පරිගණක පරිවර්තකයා (Language Translator) සිදුකරන තවත් ප්‍රධාන කාර්යයක් නම් ප්‍රභව ක්‍රමලේඛයෙහි ඇති වැරදි හඳුනා ගැනීමයි. වැරැදි සහිත ක්‍රමලේඛ පරිවර්තකයන් මගින් විෂය ක්‍රමලේඛනයනට පරිවර්තනය කළ නො හැකි ය. එබැවින් ක්‍රමලේඛනයන් ගේ ඇති සියලු වැරදි හඳුනා ගැනීම හා නිදෙස් කිරීම මෙම ක්‍රියාවලියේ දී සිදු කළ යුතු ය. මෙසේ ක්‍රමලේඛයන් ගේ ඇති වැරැදි හඳුනා ගැනීම සහ එම වැරැදි නිදෙස් කිරීමේ ක්‍රියාවලිය 'Debugging' ලෙස ද ක්‍රමලේඛයන් ගේ ඇති වැරැදි 'Bugs' ලෙස ද හැඳින්වෙයි.

අර්ථ වින්‍යාසක (Interpreters)

අර්ථ වින්‍යාසක මෘදුකාංග මගින් මුළු ක්‍රමලේඛය ම එක වර යාන්ත්‍රික භාෂාවකට පරිවර්තනය කිරීම සිදු නො වේ. මෙහි දී සිදුවන්නේ ක්‍රමලේඛයේ අන්තර්ගත විධාන එකිනෙක ක්‍රියාත්මක කිරීමේ අවස්ථාවේ දී ම අදළ යාන්ත්‍රික භාෂා විධාන සමූහයකට පරිවර්තනය කිරීමයි. මෙහි ප්‍රතිඵලය වන්නේ මෙම වර්ගයේ ක්‍රමලේඛයක් ක්‍රියා කරවන සෑම අවස්ථාවක දී ම මෙම පරිවර්තන ක්‍රියාවලිය නැවත නැවත සිදු කිරීමට අවශ්‍ය වීමයි. එබැවින් සාමාන්‍යයෙන් මෙම වර්ගයේ ක්‍රමලේඛයක් ක්‍රියා කිරීමට ගන්නා කාලය සම්පාදනය කළ ක්‍රමලේඛයට ගන්නා කාලයට වඩා වැඩි වෙයි. VB, Python මෙම වර්ගයට අයත් පරිගණක භාෂා වෙයි.

1.2 පරිගණක අනුසාරයෙන් ගැටලු විසඳීමේ අදියර

ගැටලු විසඳීම සඳහා පරිගණක භාවිත කිරීමේ දී අනුගමනය කළ යුතු අදියර කිහිපයකි.

1) ගැටලුව විශ්ලේෂණය කිරීම.
2) ගැටලුව විසඳීම සඳහා සුදුසු ඇල්ගොරිතමයක් ගොඩනැගීම.
3) ගොඩනැගු ඇල්ගොරිතමය පරිගණක ක්‍රමලේඛයකට පරිවර්තනය කිරීම.
4) ක්‍රමලේඛයේ ඇති වැරැදි හඳුනා ගැනීම සහ නිදෙස් කිරීම.
5) ක්‍රමලේඛ අදළ දත්ත මත ක්‍රියාත්මක කිරීම.

1) ගැටලු විශ්ලේෂණය

ගැටලු විශ්ලේෂණයේ මූලික පරමාර්ථය වන්නේ

1. ආදාන (Inputs)
2. ප්‍රතිදාන (Outputs)
3. ආදාන ප්‍රතිදාන බවට පරිවර්තනය කිරීමේ ක්‍රියාවලිය (Processing) හඳුනා ගැනීමයි.

උදාහරණය ඃ සෘජුකෝණාස්‍රයක වර්ගඵලය සෙවීමට අවශ්‍ය යැයි සිතන්න. මෙම කාර්යය සඳහා අදාළ ආදාන, ප්‍රතිදාන හා ක්‍රියාවලීන් පහත දැක්වෙයි.

ආදාන ඃ සෘජුකෝණාස්‍රයේ දිග හා පළල

ක්‍රියාවලිය ඃ දිග x පළල

ප්‍රතිදාන ඃ ඉහත ක්‍රියාවලියෙන් ලබාගත් ප්‍රතිඵලය

2) ගැටලු විසඳීම සඳහා ඇල්ගොරිතම ගොඩනැඟීම

යම් ගැටලුවක් විසඳීම සඳහා අනුගමනය කළ යුතු පියවර සියල්ල අනුපිළිවෙළින් දැක්වූ ක්‍රමවේදයක් ඇල්ගොරිතමයක් ලෙස අර්ථ දැක්වෙයි.

ඉහතින් දැක්වූ උදාහරණය සඳහා පහත ඇල්ගොරිතමය ගොඩනැඟිය හැකි ය.

- සෘජුකෝණාස්‍රයේ දිග ලබා ගැනීම.
- සෘජුකෝණාස්‍රයේ පළල ලබා ගැනීම.
- වර්ගඵලය = දිග x පළල.
- වර්ගඵලය ප්‍රකාශ කිරීම.

3) ඇල්ගොරිතම ක්‍රමලේඛයන්ට පරිවර්තනය කිරීම

පරිගණක අනුසාරයෙන් ගැටලු විසඳීමේ දී ගැටලුව විසඳීම සඳහා ගොඩනැගූ ඇල්ගොරිතම ඉතා නිවැරැදි ව, පරිගණකයට තේරුම් ගත හැකි භාෂාවකින් පරිගණකගත කළ යුතු වෙයි. වර්තමානයේ මේ සඳහා නිර්මාණය වූ පරිගණක භාෂා ඉතා විශාල

සංඛ්‍යාවක් ඇත. මේ සියලු ම භාෂා පරිගණක අනුසාරයෙන් ගැටලු විසඳීම සඳහා සකස් වූ කෘත්‍රිම භාෂා වශයෙන් හැඳින්වෙයි. C, Phython, Php, VB, FORTRAN මේ සඳහා උදහරණ කිහිපයකි. ඇල්ගොරිතමයක් පරිගණක භාෂාවක් මගින් ලියැවුණු ක්‍රමලේඛයකට පරිවර්තනය කිරීමේ ක්‍රියාව කේතනය (Coding) ලෙස හැඳින්වෙයි. මෙම කාර්යය කරන්නා ක්‍රමලේඛකයෙකු (Programmer / Coder) ලෙස හැඳින්වෙයි.

4) ක්‍රමලේඛ නිදෙස් කිරීම

මෙහි දී ක්‍රමලේඛයන් ගේ ඇති වැරදි හඳුනාගැනීම සහ එම වැරදි නිදෙස් කිරීම සිදු වෙයි. මෙම ක්‍රියාවලිය නිදෙස් කිරීමේ ක්‍රියාවලිය (Debugging) ලෙස හැඳින්වෙයි. ඉන් අනතුරුව දත්ත සාම්පල ආදානය කර අදාළ ප්‍රතිදාන ලැබේ දැයි පරීක්ෂා කර බලනු ලැබේ.

5) ක්‍රමලේඛ ක්‍රියාත්මක කිරීම

මෙහි දී නිදෙස් කළ ක්‍රමලේඛ සත්‍ය ආදාන මත ක්‍රියාකර ප්‍රතිදාන ලබාගනු ලැබේ.

ක්‍රියාකාරකම 1.2

වෘත්තයක වර්ගඵලය සෙවීම සඳහා ක්‍රමලේඛයක් ගොඩනැගීමට අවශ්‍ය වන ආදාන ක්‍රියාවලිය හා ප්‍රතිදාන හඳුනාගන්න.

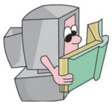
ක්‍රියාකාරකම 1.3

ඔබ අවසාන වාර විභාගයේ දී සියලු විෂයයන් සඳහා ලබා ගත් ලකුණුවල මුළු එකතුව සහ සාමාන්‍ය අගය ලබාගැනීම සඳහා ක්‍රමලේඛයක් ගොඩනැගීමේ දී අවශ්‍ය වන ආදාන ක්‍රියාවලිය සහ ප්‍රතිදානයන් හඳුනාගන්න.

1.3 ඇල්ගොරිතම නිරූපණය

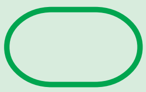
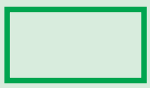
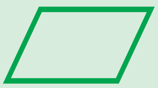
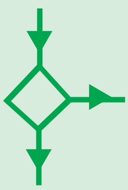
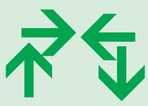
ගැටලු විසඳීම සඳහා ඇල්ගොරිතම ගොඩනැගීමේ දී අදාළ විධාන හා එම විධාන ක්‍රියා කළ යුතු අනුපිළිවෙළ පැහැදිලි ව දැක්වීම සඳහා කිසියම් ශිල්පීය ක්‍රමයක්

අවශ්‍ය වෙයි. මෙම කාර්යය සඳහා රූපමය (Graphical) හෝ ලිඛිත (Textual) නිරූපණ උපයෝගී කරගත හැකි ය. ගැලීම් සටහන් (Flowcharts) මේ සඳහා බහුල ව භාවිත වන රූපමය නිරූපණයක් වන අතර ව්‍යාජ කේත (Pseudo code) මගින් ඇල්ගොරිතම ලිඛිත ව නිරූපණය කළ හැකි ය.

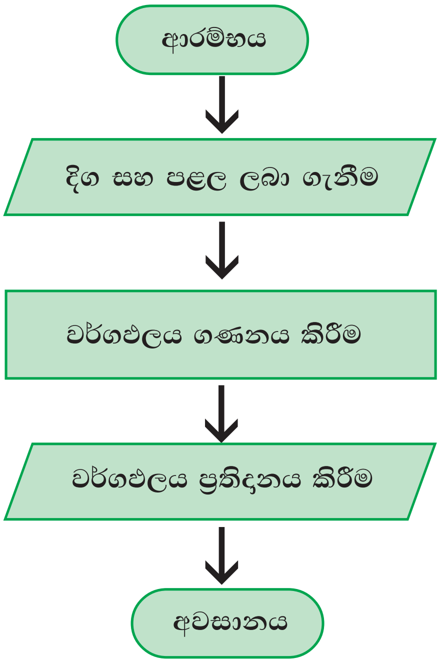


ගැලීම් සටහන් (Flowcharts)

ඇල්ගොරිතමයක විධානයන් ක්‍රියා කළ යුතු ආකාරය අනුපිළිවෙළින්, රූපමය සංකේත මගින් දක්වූ සටහනක් ගැලීම් සටහනක් ලෙස හැඳින්වෙයි. ගැලීම් සටහන් නිර්මාණයේ දී භාවිත කළ යුතු සම්මත සංකේත හඳුන්වා දී ඇත. මෙම සංකේතයන් ගෙන් බහුල ව භාවිත වන සංකේත කිහිපයක් හා එම සංකේතයන් ගේ භාවිතය පහත වගුවෙහි දක්වා ඇත.

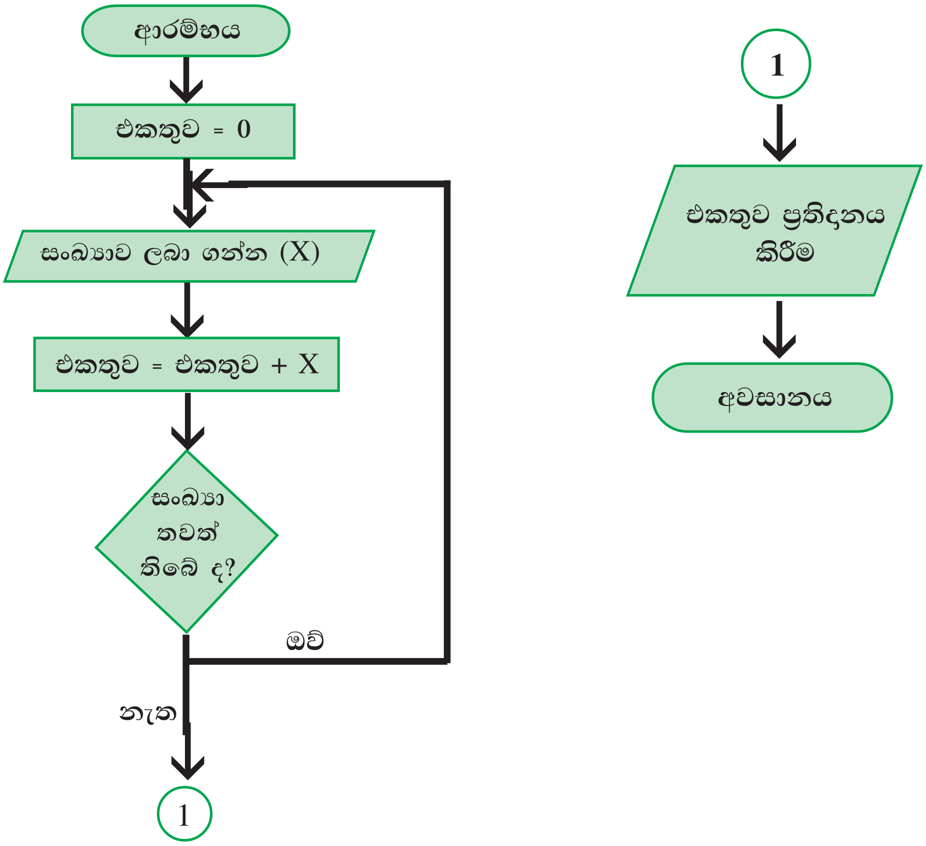
සංකේතය	නාමය	භාවිතය
	Terminator	ආරම්භය / අවසානය (Start / Stop)
	Process	විධානයක් හෝ අනුපිළිවෙළින් ක්‍රියා කළ යුතු විධාන එකතුවක්
	Data Input / Output	දත්ත ආදාන / ප්‍රතිදාන
	Decision	තීරණ
	Connector	සම්බන්ධක
	Flow Lines	දත්ත ගැලීමේ දිශාවන්

උදහරණ ඃ සෘජුකෝණාස්‍රයක වර්ගඵලය ලබාගැනීම සඳහා ගොඩනැගූ ගැලීම් සටහනක් පහත දැක්වේ.



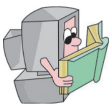
ගැලීම් සටහන් පිටු කිහිපයක් හරහා විහිදී යන අවස්ථාවන්හි දී හා ගැලීමේ දිශාවන් කෙටි රේඛා ඛණ්ඩයන් මගින් පැහැදිලි ව දැක්වීම සඳහා 'Connector' සංකේතය යොදා ගැනේ.

උදාහරණ ඃ දී ඇති සංඛ්‍යා කිහිපයක එකතුව සෙවීම සඳහා අනුගමනය කළ යුතු ක්‍රියාදාමය ගැලීම් සටහනක් මගින් දක්වන්න.



ක්‍රියාකාරකම 1.4

ඉහත උදාහරණයෙන් දක්වූ ගැලීම් සටහන සංඛ්‍යා කිහිපයක එකතුව මෙන් ම සාමාන්‍ය අගය ද ගණනය කිරීම සඳහා දීර්ඝ කරන්න.



ව්‍යාජ කේත (Pseudo codes)

ඇල්ගොරිතමයන් පරිගණක භාෂාවක් මත නොයැපෙන භාෂාවකින් (Pseudo code) සරල ව දැක්විය හැකි ය. මෙසේ දැක්වීම සඳහා භාවිත කළ යුතු භාෂාවන් හා එම භාෂාවන්ගේ අන්තර්ගතයන් සඳහා සම්මත සම්මුතියක් නොමැත. එබැවින් ඇල්ගොරිතමයක් ව්‍යාජ කේත මගින් නිරූපණය කිරීමේ දී, එම නිරූපණය ක්‍රමලේඛකයකු නොවන අයකුට ද වැටහෙන ලෙස සරල ව ප්‍රකාශ කිරීමට වගබලා ගත යුතු ය.

උදාහරණ ඃ සෘජුකෝණාස්‍රයක වර්ගඵලය සෙවීමේ ක්‍රියාවලිය පහත දැක්වෙන ලෙස ව්‍යාජ කේතයකින් දැක්විය හැකි ය.

ආරම්භය

සෘජුකෝණාස්‍රයේ දිග හා පළල ලබා ගන්න

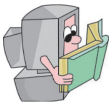
වර්ගඵලය = දිග × පළල

වර්ගඵලය ප්‍රතිදානය කරන්න

අවසානය

1.4 විෂුවල් බේසික් (Visual Basic) භාවිතයෙන් වැඩසටහන් සම්පාදනය කිරීම

මෙම කොටසේ දී ගැටලු විසඳීම සඳහා විෂුවල් බේසික් (Microsoft Visual Basic 6.0) පරිගණක භාෂාව උපයෝගි කර වැඩසටහන් සකසා ගන්නා අයුරු සලකා බැලේ.



විෂුවල් බේසික්

'Visual Basic' හෝ 'VB' යනු තුන් වන පරම්පරාවට අයත් පරිගණක භාෂාවකි. 'VB' භාෂාව ප්‍රධාන ලෙස වින්ඩෝස් මෙහෙයුම් පද්ධතිය මත ක්‍රියාකරවීම සඳහා නිර්මාණය කර ඇති පරිගණක භාෂාවකි. එබැවින් 'VB' වලින් ලියවුණු ක්‍රමලේඛයන් 'Linux' වැනි මෙහෙයුම් පද්ධති මත ක්‍රියාකරවීමට නො හැකි ය. මෙය 'VB' භාෂාවේ මූලික අඩුපාඩුවක් ලෙස දැක්විය හැකි ය.

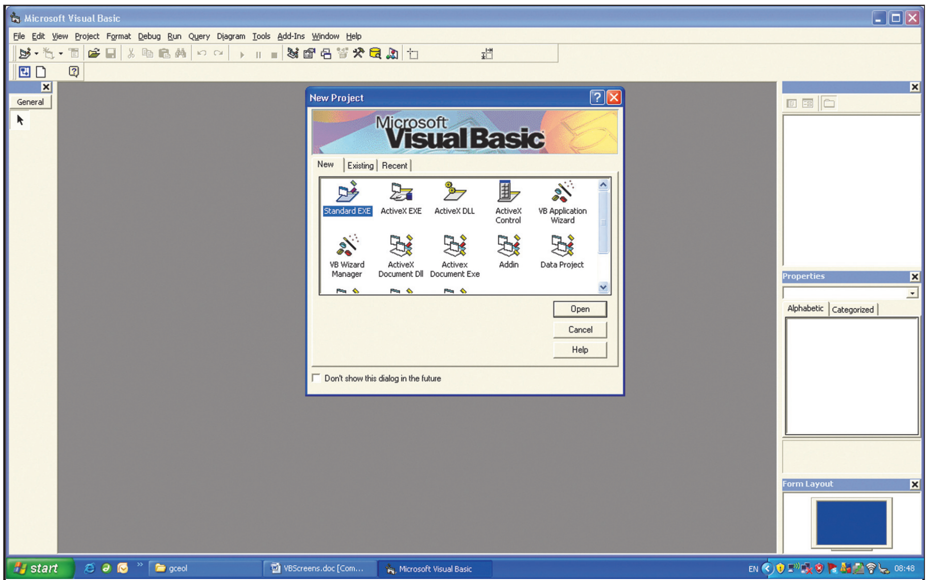
'VB' පරිගණක භාෂාවෙන් ක්‍රමලේඛ සකස්කර ගැනීම සඳහා අනුගමනය කළ යුතු ක්‍රියාවලිය සමහර පරිගණක භාෂාවන්ගෙන් ක්‍රමලේඛ සකස්කර ගැනීමේ ක්‍රියාවලියට වඩා තරමක් වෙනස් වෙයි. මෙයට මූලික හේතුව 'VB' භාෂාව වැඩසටහන් සම්පාදක මෙවලමක් හා අනුබද්ධ කර තිබීමයි. මෙවැනි වැඩසටහන් සම්පාදක මෙවලම් 'Integrated Develpoment Environments' (IDE) ලෙස හැඳින්වෙයි. මෙවැනි වැඩසටහන් සම්පාදක මෙවලම්වල ප්‍රධාන කාර්යය වන්නේ ක්‍රමලේඛ පහසුවෙන් ගොඩනැගීම සඳහා පහසු පසුබිමක් නිර්මාණය කරදීමයි. එමගින් පරිගණක භාෂාවන් පිළිබඳ මනා දැනුමක් නොමැති ව වුව ද සරල වැඩසටහන් ගොඩනගා ගැනීමට අවශ්‍ය පහසු පසුබිමක් ලබාදෙයි. එබැවින් 'VB' භාවිත කර ක්‍රමලේඛ ලිවීම සඳහා 'VB' හා අනුබද්ධිත 'IDE' ගැන මනා දැනුමක් ලබාගැනීම අත්‍යවශ්‍ය කරුණකි.

විෂුවල් බේසික් වැඩසටහන් සම්පාදක IDE මුහුණතකට පිවිසීම.

'VB' වැඩසටහනක් සම්පාදනය කිරීම සඳහා අදාළ 'IDE' මුහුණතකට පිවිසීම පහත දැක්වෙන විධානයන් ක්‍රියාත්මක කිරීමෙන් කළ හැකි ය.

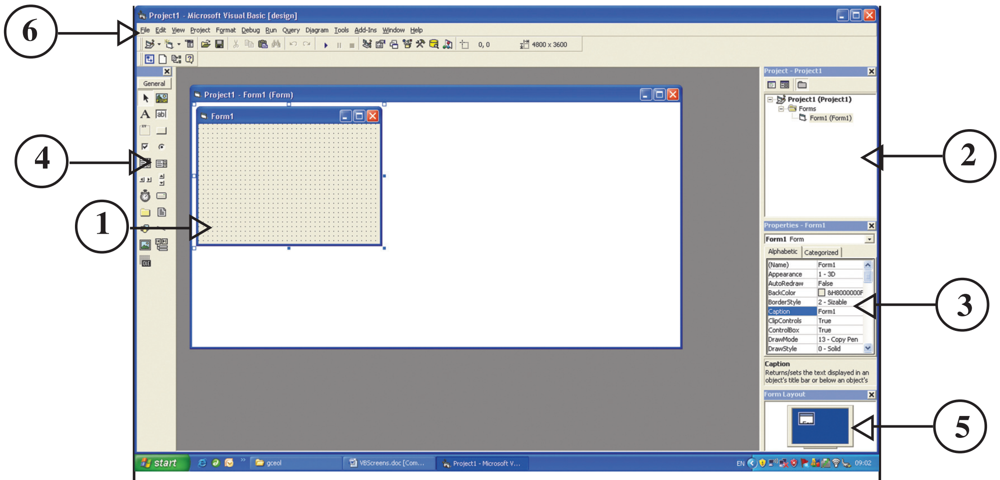
1) Start
2) All Programs
3) Visual Studio 6.0
4) Microsoft Visual Basic 6.0

ඉහත විධාන ක්‍රියාත්මක කිරීමෙන් පසුව පහත දැක්වෙන පරිදි මුහුණතක් තිරය මත දිස්වෙයි.



රූපය 1.1

විෂුවල් බේසික් 'IDE' මගින් විවිධ ආකාරයේ වැඩසටහන් සම්පාදනය කළ හැකි ය. මෙවැනි විවිධ ආකාරයේ වැඩසටහන් සම්පාදනය කිරීමේ දී ක්‍රියාකරවිය යුතු විධානයන්ට අදාළ අයිකනයන් (Icons) මෙම තිරයේ දිස්වෙයි. අපගේ අවධානය මෙහි දී යොමුවන්නේ Standard. EXE වර්ගයේ වැඩසටහන් සඳහා බැවින්, Standard. EXE අයිකනය ක්ලික් කිරීමෙන් එයට අදාළ මුහුණතකට පිවිසිය හැකි ය. මෙවැනි මුහුණතක් පහත රූපයෙන් දැක්වෙයි.



රූපය 1.2

1. පෝරම සැලසුම් කවුළුව (Form Design Window)
2. ව්‍යාපෘති ගවේෂණ කවුළුව (Project Explorer Window)
3. ගුණාංග කවුළුව (Properties Window)
4. මෙවලම් පෙට්ටිය (Tool Box)
5. පෝරම පිරිසැලසුම් කවුළුව (Form Layout Window)
6. ප්‍රධාන මෙනුව (Main Menu)

සැලකිය යුතුයි ඃ *ඔබගේ පරිගණකයේ ස්ථාපිත කර ඇති වැඩසටහන් අනුව ක්‍රියාත්මක කළ යුතු විධානයන් සහ එම විධානයන් ක්‍රියාත්මක කළ යුතු ආකාරය වෙනස් විය හැකි ය.*

'VB' වැඩසටහනක් සම්පාදනය කිරීමට අවශ්‍ය සියලු මෙවලම් ඉහත වැඩසටහන් සම්පාදක 'IDE' මුහුණතෙහි අඩංගු වෙයි.

'VB' වැඩසටහනක් විවිධ ආකාරයේ ගොනු කිහිපයකින් සමන්විත විය හැකි ය. මෙසේ කිසියම් 'VB' වැඩසටහනකට අයත් වන ගොනු සියල්ලේ ම එකතුවක් විෂුවල් බේසික් ව්‍යාපෘතියක් වශයෙන් හඳුන්වනු ලැබේ.

ඉහත 'VB IDE' මුහුණතේ 1 යනුවෙන් දක්වා ඇති පෝරම සැලසුම් කවුලුව (Form Design Window) වැඩසටහනේ උපයෝගී මුහුණත් (Application Interfaces) සකස්කර ගැනීම සඳහා භාවිත වෙයි. රූපයේ 2 යනුවෙන් දක්වා ඇති කොටසින් ඔබ වැඩකරමින් සිටින සියලු ව්‍යාපෘති (Project) වල නම් හා එම එක් එක් ව්‍යාපෘතියකට අයත් සියලු පෝරමවල නම් දක්වෙයි.

ක්‍රියාකාරකම 1.5

රූපය 1.2 හි දක්වා ඇති කවුළු සියල්ල ඔබගේ පරිගණක තිරයෙහි ඇති දැයි නිරීක්ෂණය කරන්න. යම් කිසි කවුළුවක් තිරයේ දිස් නො වේ නම් එය තිරය මතට ලබා ගන්නා ආකාරය සොයා බලන්න.

රූපය 1.2 හි (3) යනුවෙන් දක්වා ඇති කොටසින් විෂුවල් බේසික් ව්‍යාපෘතියෙහි අඩංගු සියලු උපාංගවල ගුණාංග (Properties) පෙන්නුම් කරයි. එමෙන් ම මෙම ලක්ෂණවල අගයන් වෙනස් කිරීමට අවශ්‍ය පසුබිම ද මෙම කොටසින් සලසා දෙයි.

ක්‍රියාකාරකම 1.6

ව්‍යාපෘති කවුළුවෙහි ඇති ව්‍යාපෘති නාමය මත ක්ලික් කර ගුණාංග කවුළුවෙහි ව්‍යාපෘතියේ නාමය 'First Project' ලෙස වෙනස් කරන්න.

රූපය 1.2 හි (4) යනුවෙන් දක්වා ඇති කොටස මෙවලම් පෙට්ටිය (Tool Box) ලෙස හැඳින්වෙයි. 'VB' වැඩසටහනක් සම්පාදනය සඳහා බහුල ව භාවිත වන පාලන (Controls) නම් වූ උපාංග මෙම කොටසෙහි දක්වෙයි.

රූපය 1.2 හි (5) යනුවෙන් දක්වා ඇති කොටසින් දැක්වෙන්නේ ඔබගේ වැඩසටහන ක්‍රියාත්මක කළ පසු එය පරිගණක තිරයේ දිස්වෙන ආකාරයයි.

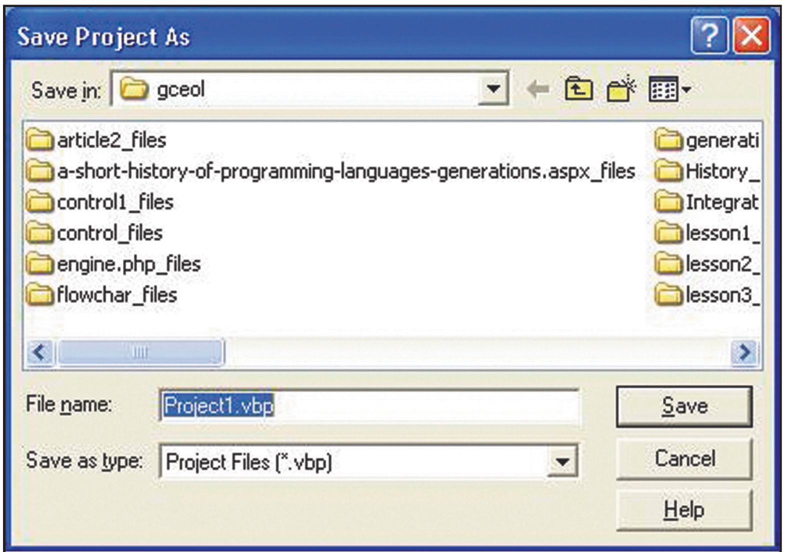
ක්‍රියාකාරකම 1.7

ෆෝරම සැලසුම් කවුළුවෙහි ඇති 'Form 1' මත ක්ලික් කර ගුණාංග කවුළුවෙහි 'Caption' යන ගුණාංගය 'First Form' ලෙස වෙනස් කරන්න. ඔබ ලබා ගත් ෆෝරමයෙහි පළල (Width) කොපමණ ද?

ක්‍රියාකාරකම 1.8

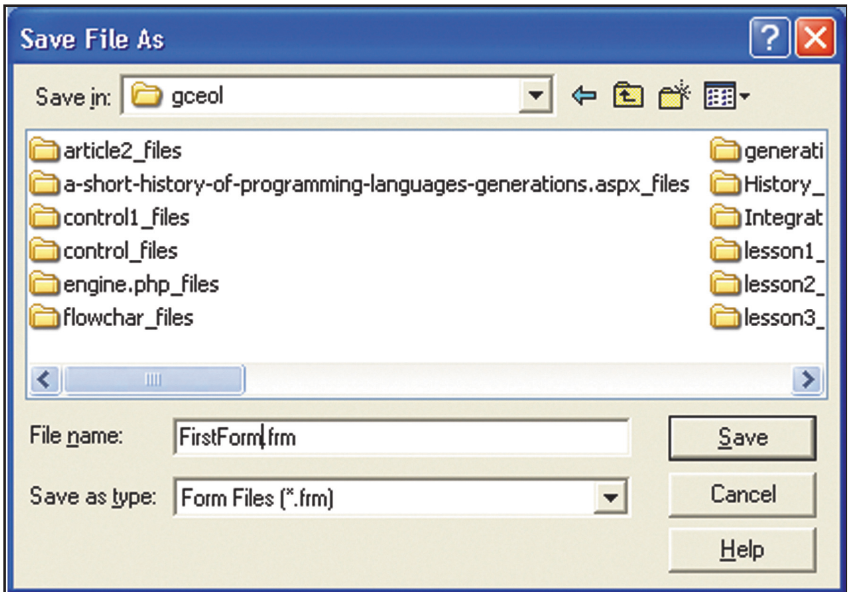
ඉහත රූපයේ 6 යනුවෙන් දක්වා ඇති ප්‍රධාන මෙනුවේ (Main Menu) Run → Start ක්‍රියාත්මක කර ඇති වන ප්‍රතිඵලය නිරීක්ෂණය කරන්න. නැවතත් Run → End විධාන ක්‍රියාත්මක කර ඇති වන ප්‍රතිඵලය නිරීක්ෂණය කරන්න.

ඔබ මෙතෙක් සකස් කළ ව්‍යාපෘතිය File → Save Project යන විධාන මගින් ඔබ කැමැති නාමයක් යටතේ තැන්පත් (Save) කරන්න. (රූපය 1.3 බලන්න)



රූපය 1.3

මෙහි දී ව්‍යාපෘති කවුළුවෙහි ඇති ඔබගේ ව්‍යාපෘතියට අදාළ සියලු ම පෝරම තැන්පත් කිරීම සඳහා අවශ්‍ය නාමයන් ඔබට සැපයීමට සිදුවෙයි. (රූපය 1.4 බලන්න)



රූපය 1.4

ප්‍රධාන මෙනුවේ ඇති File → Exit විධාන ක්‍රියාත්මක කිරීමෙන් ඔබට 'VB IDE' වැඩසටහනෙන් ඉවත් විය හැකි ය.

ක්‍රියාකාරකම 1.9

ඔබ ඉහත දී තැන්පත් කළ ව්‍යාපෘති නාමය සහිත අයිකනය මත ද්විත්ලික් කර ඇතිවන ප්‍රතිඵලය නිරීක්ෂණය කරන්න.

ක්‍රියාකාරකම 1.10

File → Open Project යන විධානයන් ක්‍රියාත්මක කිරීම මගින් කලින් අවස්ථාවක තැන්පත් කළ ව්‍යාපෘතියක් යළිත් විවෘත කර ගන්නා අන්දම සොයා බලන්න.

1.5 විෂුවල් බේසික් වැඩසටහනක් ගොඩනැගීමේ අදියර

'VB' පරිගණක භාෂාව සිද්ධි ධාවිත (Event Driven) වර්ගයේ වැඩසටහන් සම්පාදනය සඳහා නිර්මාණය වූ භාෂාවකි. මෙවැනි භාෂාවන්ගෙන් නිමවුණු වැඩසටහන්වල ක්‍රියාකාරිත්වය එම වැඩසටහනේ මුහුණත මත දිස්වෙන වස්තූන් මත පරිශීලකයා විසින් ඇතිකරන ක්‍රියාවන් මත රඳා පවතී.
උදාහරණ : 'Click', 'Change', 'GotFocus', 'KeyPress'..

'VB' වැඩසටහනක මුහුණත් මත දිස්වෙන සිද්ධියක් (Events) ජනනය කළ හැකි වස්තූන් පාලනයන් (Controls) ලෙස හැඳින්වෙයි. 'VB' වැඩසටහනක් සම්පාදනය කිරීමේ දී 'IDE' මුහුණත මත දිස්වෙන පෝරම ද එවැනි පාලනයකි.

'VB' වැඩසටහනක් සම්පාදනය සඳහා යොදගත හැකි පාලන විශාල සංඛ්‍යාවක් වෙයි. මෙයින් නිතර ම හා ප්‍රධාන ලෙස භාවිත වන පාලන කිහිපයක් 'IDE' හි ඇති මෙවලම් පෙට්ටියෙහි දැක්වෙයි.

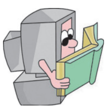
මූසික දක්වනය (Mouse Pointer) විෂුවල් බේසික් මෙවලම් පෙට්ටිය මත දිස්වෙන අයිකනයක් මතට ගෙනයාමෙන් දිස්වෙන මෙවලම් ඉඟි (Tool Tips) මගින් එම අයිකනයන් මොනවා දැයි අපට දැනගත හැකි වෙයි.

ක්‍රියාකාරකම 1.11

1. මෙවලම් පෙට්ටිය මත ඇති අයිකන මතට මූසික දක්වනය ගෙන ගොස් එම අයිකන මොනවා දැයි හඳුනා ගන්න.

2. 'IDE' මුහුණතේ ඇති පෝරමයේ ප්‍රමාණය (දිග හා පළල) වෙනස් කරන්නේ කෙසේ දැයි සොයා බලන්න. මෙසේ වෙනස් කිරීමේ දී 'Properties' කොටසේ ඇති උස (Height) සහ පළල (Width) ලක්ෂණ වෙනස් වන ආකාරය නිරීක්ෂණය කරන්න.

3. මෙවලම් පෙට්ටියෙහි ඉහළ දකුණු කෙළවරේ ඇති ☒ අයිකනය ක්ලික් කිරීමෙන් ඇතිවන ප්‍රතිඵලය නිරීක්ෂණය කරන්න. ප්‍රධාන මෙනුවෙහි ඇති View → Tool Box ක්ලික් කිරීමෙන් නැවත මෙවලම් පෙට්ටිය, 'IDE' මුහුණත මත ලබාගන්න.



පෝරමයකට පාලන (Controls) ඇතුළු කිරීම.

මෙය කළ හැකි ප්‍රධාන ක්‍රම දෙකකි.

1) අවශ්‍ය පාලන මත දෙවරක් ක්ලික් කිරීම : මෙහි දී එම පාලන පෝරමයේ මැදට ස්ථානගත වෙයි. මෙම පාලන පසු ව අවශ්‍ය ස්ථානයට මූසිකය මගින් ගෙන යා හැකි ය.

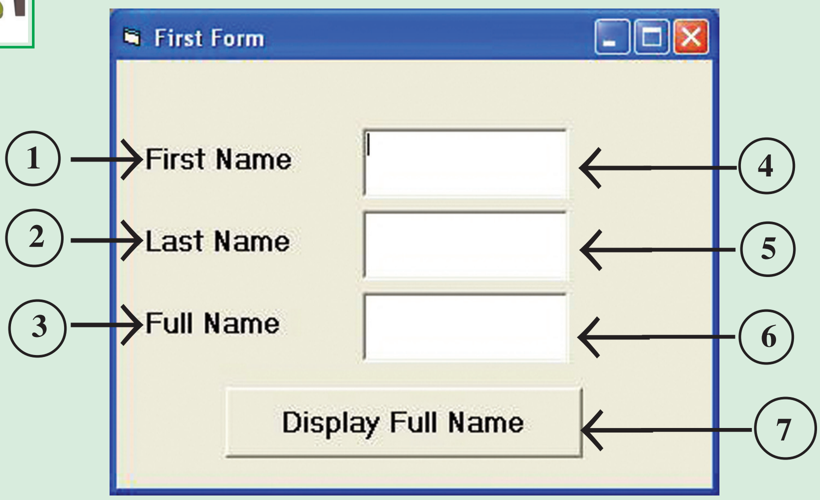
2) අවශ්‍ය පාලනය මත මූසිකය ක්ලික් කර, පෝරමයේ අවශ්‍ය තැනට ගෙන ගොස්, අවශ්‍ය ප්‍රමාණයට මූසිකය ඇදීම. (Drag)

පෝරමය මත පාලන ඇතුළත් කිරීමේ දී 'IDE' මගින් එම පාලනවල බොහෝ ගුණාංගයන්ට (Properties) සුදුසු අගයයන් ස්වයංක්‍රීය ව ලබා දෙයි. පසු ව මෙම අගයයන් අවශ්‍ය ලෙස ගුණාංග කවුළුව තුළින් වෙනස් කළ හැකි ය.

'VB' වැඩසටහන් ගොඩනැගීමේ දී අනුගමනය කළ යුතු අදියර තුනකි.

1) පෝරමයක් මත පාලන ඇතුළත් කිරීම.
2) පාලන ගුණාංග (Properties) අවශ්‍ය ලෙස වෙනස් කිරීම.
3) පාලනවල ක්‍රියාකාරිත්වය සඳහා ක්‍රමලේඛනය කිරීම.

ක්‍රියාකාරකම 1.12



ක්‍රියාකාරකම 1.12 රූපයේ දැක්වෙන ආකාරයට පාලන (Control) 7ක් ඇතුළත් කරන්න. මෙම පාලන 7හි ගුණාංගයන් පහත දැක්වෙන අගයයන්ට සකස් කරන්න.

ඔබ නිර්මාණය කළ වැඩසටහන ප්‍රධාන මෙනුවෙහි ඇති File → Save Project විධාන මගින් 'My Project.Vbp' නමින් තැන්පත් කරන්න.

මෙහි දී ප්‍රථමයෙන් ම ඔබ සකස් කරන ලද ෆෝරමය තැන්පත් කිරීමට සිදු වන අතර ඒ සඳහා FirstForm යන නම භාවිත කරමු.

a) Form Object Properties

Name	:	frmFirst
Caption	:	First Form
Width	:	10000
Height	:	6000

b) Object 1

Type	:	Label
Name	:	1b1FName
Caption	:	First Name
Width	:	1000
Left	:	450
Top	:	400
Height	:	200

c) Object 2

Type	:	Label
Name	:	1b1LName
Caption	:	Last Name
Width	:	1000
Left	:	450
Top	:	950
Height	:	200

d) Object 3

Type	:	Label
Name	:	1b1FullName
Caption	:	Full Name
Width	:	1000
Left	:	450
Top	:	1500
Height	:	200

e) Object 4

Type	:	Text Box
Name	:	txtFName
Width	:	3000
Left	:	1600
Top	:	400
Height	:	200
Text	:	

f) Object 5

Type	:	Text Box
Name	:	txtLName
Width	:	1600
Left	:	1600
Top	:	950
Height	:	200
Text	:	

g) Object 6

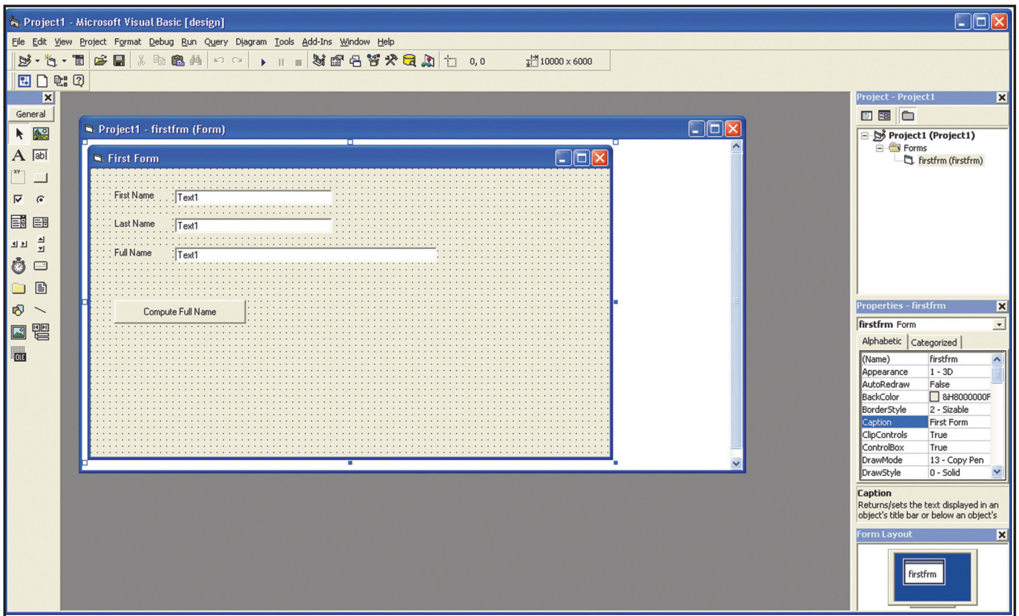
Type	:	Text Box
Name	:	txtFullName
Width	:	5000
Left	:	1600
Top	:	1500
Height	:	200
Text	:	

h) Object 7

Type	:	Command Button
Name	:	cmdFullName
Caption	:	Display Full Name
Width	:	2500
Left	:	450
Top	:	2500
Height	:	200

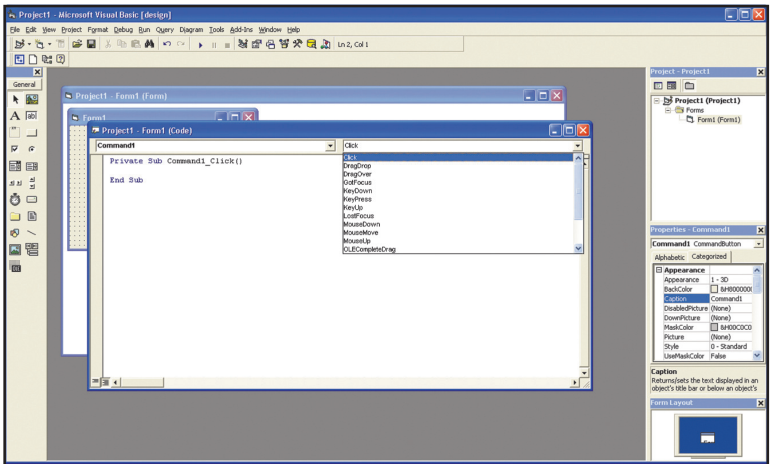
1.6 පාලනයක ක්‍රියාකාරිත්වය සඳහා කේතයන් (Code) ලිවීම.

මේ සඳහා ඉහත ක්‍රියාකාරකම තුළින් සම්පාදනය කර ගත් ව්‍යාපෘතිය නැවත විවෘත කරන්න.



රූපය 1.6

මෙම මුහුණත මත ඇති "Display Full Name" යනුවෙන් දැක්වෙන විධාන බොත්තම (Command Button) මත ද්විත්ලික් (Double Click) කිරීමෙන් පහත දැක්වෙන ආකාරයේ මුහුණතක් ලබාගැනීමට හැකි වෙයි.



රූපය 1.7

මෙම මුහුණත කේත කවුළුව (Code Window) ලෙස හැඳින්වෙයි. යම් කිසි පාලනයක් සමඟ කේතයක් සම්බන්ධ කිරීම සඳහා මෙම මුහුණත යොදා ගැනේ.

ක්‍රියාකාරකම 1.13

මුහුණතේ දිස්වෙන කේත පහත දක්වා ඇත.

Private Sub cmdFullName_Click()

End Sub

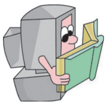
මෙම කේත අතරට පහත දැක්වෙන කේතයන් ඇතුළත් කරන්න.

```
Private Sub cmdFullName_Click()

    txtFullName.Text = txtFName.Text & "  " & txtLName.Text

End Sub
```

ඉන් පසු ව ව්‍යාපෘතිය තැන්පත් කරන්න.

මෙසේ සම්පාදනය කළ වැඩසටහන ප්‍රධාන මෙනුවේ ඇති 'Run' මගින් ක්‍රියාකරවා දිස්වෙන මුහුණත මත 'First Name' හා 'Last Name' ඉදිරියෙහි කිසියම් ශිෂ්‍යයෙකුගේ පළමු නම හා අවසාන නම යතුරු ලියනය කර 'Display Full Name' යන විධාන බොත්තම ක්ලික් කර ඇති වන ප්‍රතිඵලය නිරීක්ෂණය කරන්න.

'VB' වැඩසටහනක් 'VB' අනු ක්‍රමලේඛන (Sub Program) කිහිපයක එකතුවක් ලෙස සැලකිය හැකි ය. එබැවින් 'VB' අනු වැඩසටහක් ගොඩ නඟාගැනීමට අවශ්‍ය කරුණු සලකා බලමු.



විෂුවල් බේසික් ක්‍රමලේඛනයක සංයුතිය

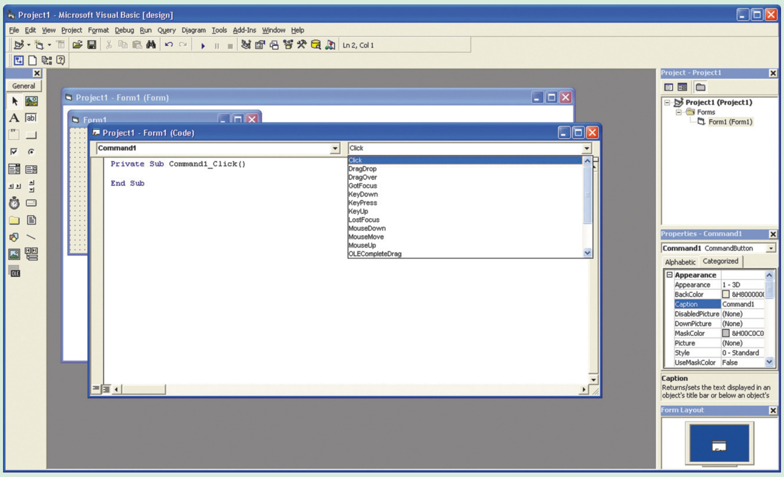
'VB' වැඩසටහනක් ප්‍රධාන ලෙස පෝරමයකින් හෝ පෝරම කිහිපයකින් සමන්විත විය හැකි ය. 'VB' හි 'IDE' ආරම්භ කිරීමේ දී පෝරමයක් ස්වයංක්‍රීය ව නිර්මාණය කරදෙනු ලබයි. වැඩසටහනට තවත් පෝරම අවශ්‍ය විට Project → Add Form විධාන ක්‍රියාත්මක කිරීමෙන් අවශ්‍ය පෝරම ලබා ගත හැකි ය. 'VB' වැඩසටහනක පෝරමයන් වැඩසටහනේ අතුරු මුහුණත් නිර්මාණය සඳහා යොද ගැනේ. වැඩසටහනේ අතුරු මුහුණත් නිර්මාණය කිරීම සඳහා අවශ්‍ය මූලික පාලනයන් මෙවලම් පෙට්ටියෙන් තෝරා ගත හැකි ය.

සෑම පාලනයක් හා අනුබද්ධ ව හඳුන්වා ඇති සිද්ධි (Events) රැසක් වෙයි. මෙම ක්‍රියාකාරකම් අදළ වැඩසටහනට අනුකූල ව සිදු විය යුතු ආකාරය කාර්ය පටිපාටියක් (Procedure) මගින් දක්විය හැකි ය. කාර්ය පටිපාටියක් විධාන එකතුවකින් සමන්විත ක්‍රමලේඛයක් (Program) ලෙස හැඳින්විය හැකි ය.

'VB' වැඩසටහනක් ක්‍රියාත්මක කළ විට එයට අදළ පෝරමයන් හා එම පෝරමයන් මත නිර්මාණය කළ පාලන තිරය මත දිස්වෙයි. පරිශීලකයාට මෙම පාලන මත අර්ථ දක්වා ඇති කාර්යයන් (Click, Move, Drag වැනි) කිරීමෙන් වැඩසටහන හැසිරවිය හැකි ය. පරිශීලකයා කිසි යම් පාලනයක් මත කාර්යයක් කිරීමේ දී එයට අදළ ව ඇති කාර්ය පටිපාටිය ක්‍රියාත්මක වෙයි.

ක්‍රියාකාරකම 1.14

'VB', පෝරමයක් මත විධාන බොත්තම් පාලනයක් (Command Button Control) නිර්මාණය කරන්න. මෙම විධාන බොත්තම් මත ද්විත්ලික් කිරීමෙන් කේත කවුළුව ලබා ගන්න. මෙම කේත කවුළුවෙන් සිද්ධි (Event) කොටසට පිවිස (පහත රූපසටහනේ දක්වා ඇත.) විධාන බොත්තමක් මගින් ජනනය කළ හැකි ක්‍රියාකාරකම් නිරීක්ෂණය කරන්න. මෙම සිද්ධි තෝරා ගත් විට කේත කවුළුවෙහි ඇති වන වෙනස්කම් නිරීක්ෂණය කරන්න.



විෂුවල් බේසික් විශේෂිත පද (Reserved Words / Key Words)

'VB' පරිගණක භාෂාවෙහි ද අනෙක් බොහෝ පරිගණක භාෂාවන්හි මෙන් විශේෂිත පද ලෙස අර්ථ දක්වූ වචන ලැයිස්තුවක් වෙයි. 'Private', 'Sub' සහ 'End', 'If', 'End If' මේ සඳහා උදාහරණ කිහිපයකි. 'VB' ක්‍රමලේඛ ගොඩනැගීමේ දී මෙම වචනවල භාවිතය ඒවායේ අර්ථයන්ට අනුකූල ව සිදු කළ යුතු ය. උදාහරණයක් ලෙස 'Sub' යන වචනය 'VB' කාර්ය පටිපාටියක ආරම්භය දැක්වීම සඳහා ද 'End Sub' යන වචන එම කාර්ය පටිපාටියෙහි අවසානය දැක්වීම සඳහා ද යොදා ගැනේ. විශේෂිත පද විෂුවල් බේසික් කේත කවුළුවෙහි නිල් පැහැයෙන් දිස්වෙයි.

විෂුවල් බේසික් කාර්ය පටිපාටි (VB Procedures)

මීට පෙර දැක්වූ පරිදි 'VB' වැඩසටහනක් කාර්ය පටිපාටි කිහිපයක එකතුවක් ලෙස සැලකිය හැකි ය. 'VB' කාර්ය පටිපාටි වර්ග තුනකි.

1) සිද්ධි කාර්ය පටිපාටි (Event Procedures)
2) පොදු කාර්ය පටිපාටි (General Procedures)
3) ක්‍රියා කාර්ය පටිපාටි (Function Procedures)

ෆෝරමයක වූ පාලන (Controls) හා බැඳුණු කාර්ය පටිපාටි, සිද්ධි කාර්ය පරිපාටි (Event Procedures) ලෙස හැඳින්වෙයි. මෙම කාර්ය පටිපාටි ක්‍රමලේඛයක 'Private Sub' යන වචන මගින් ආරම්භ වෙයි. මෙවැනි කාර්ය පටිපාටියක් අදාළ ෆෝරමයට පමණක්

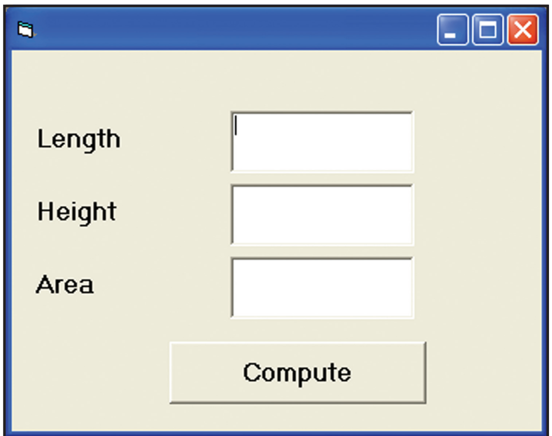
ආවේණික වන අතර 'VB' වැඩසටහනේ වෙනත් පෝරම මත වූ පාලන මගින් ක්‍රියාත්මක කළ නො හැකි ය.

පාලන හා නොබැඳුණු කාර්ය පටිපාටි, පොදු කාර්ය පටිපාටි (General Procedures) ලෙස හැඳින්වෙයි. මෙවැනි කාර්ය පටිපාටිවල ආරම්භ සඳහා 'Private Sub' හෝ 'Public Sub' යන විශේෂිත පද යොද ගත හැකි ය. 'Private Sub' යනුවෙන් ආරම්භ කළ කාර්ය පටිපාටිය අදාළ පෝරමයට පමණක් ආවේණික වන අතර 'Public Sub' යනුවෙන් ආරම්භ කළ කාර්ය පටිපාටිය 'VB' වැඩසටහනක අඩංගු ඕනෑ ම පෝරමයක් තුළින් ක්‍රියාත්මක කළ හැකි ය.

ක්‍රියාකාරකම 1.15

ක්‍රියා කාර්ය පරිපාටි (Function Procedures) අනෙක් වර්ගවල කාර්ය පටිපාටිවලින් වෙනස් වන්නේ කෙසේ දැයි සොයා බලන්න. ක්‍රියා කාර්ය පරිපාටියක් ක්‍රමලේඛයකට ඇතුළත් කිරීමේ දී භාවිත කළයුතු ආරම්භක මූල පද (Keywords) මොනවා දැයි සොයා බලන්න.

සෘජුකෝණාස්‍රයක දිග හා පළල දී ඇති විට එහි වර්ගඵලය සෙවීම සඳහා 'VB' කාර්ය පටිපාටියක් ගොඩනගා ගන්නා ආකාරය සොයා බලමු. මෙහි දී මෙම වැඩසටහන සඳහා අවශ්‍ය අතුරු මුහුණත (Interface) මුලින් ම සැලසුම් කළ යුතු වෙයි. මෙම ගැටලුව විසඳීම සඳහා පහත දැක්වෙන ආකාරයේ අතුරු මුහුණතක් යොද ගත හැකි ය.



රූපය 1.8

මෙහි 'Length', 'Height' හා 'Area' යන පාලන, ලේබල පාලන (Label Controls) වන අතර එම පාලනවලට ඉදිරියෙන් පාඨ කොටු පාලන (Text Box Controls) වෙයි. 'Compute' යන පාලනය (Command Button Control) වෙයි.

මෙම මුහුණතෙහි 'Length' හා 'Height' සඳහා අගයයන් සපයා 'Compute' යන විධාන බොත්තම ක්ලික් කළ විට වැඩසටහන මගින් සෘජුකෝණාස්‍රයේ වර්ගඵලය ගණනය කර එම අගය 'Area' යන කොටසෙහි සටහන් කළ යුතු වෙයි. මෙය සිදුකිරීම සඳහා 'Compute' යන පාලනයට විධාන එකතුවක් සම්බන්ධ කළ යුතු වේ.

මෙම ගණිතකර්මය කිරීම සඳහා විධාන බොත්තම මත ක්ලික් යන සිද්ධිය සඳහා සම්බන්ධ කළයුතු ක්‍රමලේඛයක් පහත දැක්වෙයි.

```
Private Sub cmdCompute_Click()
        L = Val(txtLength.Text)
        H = Val(txtHeight.Text)
        Area = L * H
        txtArea.Text = Area
End Sub
```

මෙම ක්‍රමලේඛයේ පහත දැක්වෙන ලක්ෂණ නිරීක්ෂණය කළ හැකි ය.

1. ක්‍රමලේඛය 'Private Sub' යන විශේෂිත පදය මගින් ආරම්භ කර ඇති නිසා සහ 'End Sub' වලින් අවසන් වී ඇති නිසා මෙය කාර්ය පටිපාටියකි. එමෙන් ම මෙහි 'Private' යන විශේෂිත පදය ඇති බැවින් මෙය අදාළ පෝරමය තුළින් පමණක් ක්‍රියාකරවිය හැකි කාර්ය පටිපාටියකි. මෙම 'Private Sub cmdCompute_Click ()' යන ප්‍රකාශය විධානයක් නොවන අතර එය ක්‍රමලේඛ පරිවර්තකයට මෙම ක්‍රමලේඛ ඛණ්ඩය අනු වැඩසටහන (Sub - Program) ආරම්භය වශයෙන් දක්වනු ලබයි.

2. මෙම කාර්ය පටිපාටිය හඳුනාගැනීම සඳහා 'cmdCompute_Click()' යන නම යොදා ඇත. මෙහි 'cmdCompute' යන නම විධාන බොත්තමෙහි නම වන අතර '_Click' යන කොටස 'VB' මගින් මෙය ක්ලික් යන සිද්ධියට අදළ බව දැක්වීම සඳහා ස්වයංක්‍රීය ව ලබා දුන් කොටසකි.

3. කාර්ය පටිපාටියෙහි අන්තර්ගතය (Body) විධාන කිහිපයක එකතුවකි. මෙම විධානයන් එක පේළියකට එක විධානයක් වන ලෙස පේළි කිහිපයකට ගොනු කර ඇත. මෙසේ එක් පේළියක දැක්වූ විධාන ප්‍රකාශයක් (Statement) ලෙස හැඳින්වෙයි.

4. කාර්ය පටිපාටියහි පළමු වන විධානය
   L = Val (txtLength.Text)
   වෙයි. මෙහි අඩංගු '=' යන සලකුණ 'Assignment Operator' ලෙස හැඳින්වෙයි. මෙම සලකුණ 'VB' මගින් කිසියම් කාර්යයක් (Operation) කළ යුතු බව දක්වන සලකුණකි. මෙවැනි කිසියම් කාර්යයක් සිදු කළ යුතු බව දක්වන සංකේත හෝ නාමයන් 'Operators' ලෙස හැඳින්වෙයි. මෙහි වම් පස ඇති 'L' යන්න විචල්‍යයකි. විචල්‍යය, මූලික ලෙස පරිගණකයේ ප්‍රධාන මතකයේ ඇති කොටසක් ඇමතීම සඳහා භාවිත කළ හැකි නාමයක් ලෙස සැලකිය හැකි ය. මෙහි '=' යන 'Operator' හි මූලික කාර්යය වන්නේ දකුණු පස දක්වා ඇති කර්මයන් සිදු කර ලැබෙන ප්‍රතිඵලය වම්පසින් දක්වා ඇති විචල්‍යයෙන් හඳුන්වන මතකයෙහි වූ කොටසෙහි තැන්පත් කරන ලෙස ය. මෙහි 'Val (txtLength.Text)' යන කොටසෙහි 'txtLength' යන්නෙන් දක්වන්නේ 'txtLength' යනුවෙන් නම්කර ඇති පාලනයේ text යන ගුණාංගයේ අගය ලබා ගන්නා ලෙස ය. මෙලෙස කිසියම් පාලනයක ලක්ෂණයට අදාළ අගය

   Control Name. Property Name

   යන ආකෘතිය භාවිතයෙන් ලබාගත හැකි ය. මෙහි 'Val' යන්න 'VB' මගින් හඳුන්වා ඇති ශ්‍රිතයකි. මෙම ශ්‍රිතයෙන් සිදු කරන්නේ 'txtLength' හි ඇති අගය ගණිතමය අගයක් බවට පරිවර්තනය කිරීමයි.

5. Area = L* H හි වූ '*' සංකේතයෙන් ගුණිතය දක්වෙයි. මෙහි දී සිදුවන්නේ 'L' හා 'H' යන විචල්‍යයන්ට අදළ අගයයන් මතකයෙන් ලබා ගෙන එම අගයයන්ගේ ගුණිතය සිදු කර ලැබෙන ප්‍රතිඵලය 'Area' යන විචල්‍යයෙහි තැන්පත් කිරීමයි.

6. අවසාන 'Statement' එක වන txtArea.Text = Area මගින් 'txtArea' යන පාලනයේ 'Text' යන ලක්ෂණයට Area යන විචල්‍යයේ අගය ආදේශ කරන මෙන් දක්වා ඇත.

ඉහත කාර්ය පටිපාටිය විශ්ලේෂණයේ දී පහත දක්වෙන ප්‍රධාන කරුණු දකගත හැකි ය.

1. කාර්ය පටිපාටියක් ප්‍රකාශ කිහිපයක එකතුවකි.

2. කාර්ය පටිපාටියක් තුළ අගයයන් තාවකාලික ව තබා ගැනීම සඳහා විචල්‍යයන් භාවිත කළ හැකි ය.

3. 'VB' මගින් සිදු කළ යුතු කාර්යයන් දැක්වීම සඳහා ප්‍රකාශ උපයෝගී කර ගැනේ. මෙවැනි ප්‍රකාශ විචල්‍යයන්ගෙන් හා විවිධ වර්ගයේ 'VB' කර්මයන් (Operators) ගෙන් නිර්මාණය වී ඇත.

4. 'VB' හි පෝරමයක වූ පාලනයක කිසියම් ලක්ෂණයක අගය 'ControlName.Property Name' යන ආකෘතිය මගින් ලබා ගැනීමට හෝ වෙනස් කිරීමට හැකි ය.

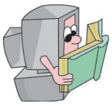
ඉහත දැක්වූ සිද්ධි කාර්ය පටිපාටිය (Event Procedure) පොදු කාර්ය පටිපාටිය (General Procedure) හා සිද්ධි කාර්ය පටිපාටියෙහි එකතුවක් ලෙස පහත දැක්වෙන පරිදි ද සම්පාදනය කළ හැකි ය.

```
Private Sub cmdCompute_Click ()
        Call ComputeArea (Val(txtLength.Text),Val(txtHeight.Text))
End Sub

Sub ComputeArea (L,H)
        Area=L*H
        txtArea.Text=Area
End Sub
```

මෙහි 'ComputeArea' යන්න පාලනයක් හා බැඳුණු කාර්ය පටිපාටි නොවන බැවින් එය පොදු කාර්ය පටිපාටියක් ලෙස දැක්විය හැකි ය. එබැවින් එය වැඩසටහනේ වූ අනෙක් ඕනෑ ම කාර්ය පටිපාටියක් මගින් ක්‍රියාත්මක කළ හැකි කාර්ය පටිපාටියකි.

ක්‍රියාකාරකම 1.16

ඉහත විස්තර කළ 'VB' වැඩසටහන සම්පාදනය කරන්න. එම වැඩසටහන ක්‍රියාත්මක කර ලැබෙන අගයයන් නිරීක්ෂණය කරන්න.



විවරණ (Comments)

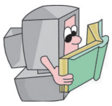
ක්‍රමලේඛයක් පිළිබඳ විස්තර ක්‍රමලේඛය තුළ ම ලේඛනගත කිරීම සඳහා විවරණ උපයෝගී කර ගැනේ. විවරණ ක්‍රමලේඛයක ඇති අනෙකුත් විධාන මෙන් ක්‍රියාත්මක කළ නො හැකි ය. ක්‍රමලේඛයක් තුළ විවරණ අන්තර්ගත කිරීමේ මූලික පරමාර්ථය වන්නේ එම ක්‍රමලේඛය පහසුවෙන් තේරුම් ගැනීමට අවශ්‍ය විස්තර සපයා දීමයි. 'VB' ක්‍රමලේඛයක විවරණ ඇතුළත් කිරීම සඳහා ' යන විශේෂ සංකේතය හෝ 'Rem' යන Reserved word එක භාවිත කරයි. මෙම සංකේතය දක්වූ ස්ථානයේ සිට එම පේළියේ අවසානය තෙක් වූ සියලු විස්තර විවරණ එකෙහි අන්තර්ගතය ලෙස සැලකෙයි. යම් විස්තරයක් යාබද පේළි කිහිපයක් තුළ විහිදී යන අවස්ථාවන්හි දී ඒ සෑම පේළියක් ම මෙම විශේෂ සලකුණෙන් (') හෝ 'Rem' වලින් ආරම්භ විය යුතු ය. 'VB IDE' හි විවරණ කොළ වර්ණයෙන් දක්වනු ලැබේ.

උදහරණ : **Program 8**

```
' ---------------------------------------------------------
' sub program to add two numbers
' ---------------------------------------------------------

Private Sub cmdAddNos_Click()

        Sum= 0

        ' Add two numbers

        Sum = Val(txtFirst.Text) + Val(txtSecond.Text)

        txtTotal.Text = Sum ' Print the Result

End Sub
```

ක්‍රියාකාරකම 1.17

1. ඉහත උදහරණයෙහි ඇති විවරණ මොනවා දැයි සොයා බලන්න.
2. sum = val(txtFirst.text)+Val(txtSecond.text) යන පේළිය 'sum = val(txtFirst.text)+Val(txtSecond.text) යන ආකාරයට වෙනස් කළ හොත් ඇති වන ප්‍රතිඵලය සාකච්ඡා කරන්න.



විෂුවල් බේසික් දත්ත වර්ග

'VB' හි භාවිත කළ හැකි දත්ත (Data), ඒවායේ ස්වභාවය, ඒවා ගබඩා කිරීමට අවශ්‍ය ඉඩ ප්‍රමාණය හා ඒවා මත ක්‍රියා කළ හැකි කර්මයන් පදනම් කර විවිධ පංතිවලට වර්ගීකරණය කෙරේ. මෙම විවිධ දත්ත වර්ගයන් 'Data Types' ලෙස හැඳින්වෙයි.

'VB' මගින් හඳුන්වා දී ඇති 'Data Types' මූලික පංති කිහිපයකට බෙදිය හැකි ය.

1. බූලියන් (Boolean) - 'True' හා 'False' යන අගයන්ට පමණක් සීමාවෙයි. 'VB' මගින් 0 අගය බූලිය 'False' අගයට සමානකර ඇති අතර අනෙකුත් සියලු අගයයන් බූලිය 'True' ලෙස සලකයි.

   මෙම වර්ගයේ අගයයන් මත 'AND', 'OR', 'NOT' යන බූලිය කර්මයන් අර්ථ දක්වා ඇත.

2. සංඛ්‍යාත්මක (Numeric) - සියලු සංඛ්‍යා මෙම වර්ගයට අයත් වෙයි.

වැඩසටහන තුළ භාවිත කිරීමට බලාපොරොත්තු වන අගයයන්ගේ විශාලත්වයන් අනුව සංඛ්‍යාත්මක දත්ත පන්තිය උප පන්ති පහකට බෙදූ ඇත.

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

මෙයින් 'Byte' 'Integer' හා 'Long' යන දත්ත පංති නිඛිල සංඛ්‍යා (පූර්ණ සංඛ්‍යා) සඳහා ද 'Single' හා 'Double' යන දත්ත පංති දශම ස්ථාන සහිත තාත්වික සංඛ්‍යා කළමනාකරණය සඳහා අර්ථ දක්වා ඇත.

3. තන්තු (String) : අක්ෂර, ඉලක්කම් හා සංකේතවලින් සැදුම්ලත් දත්ත මෙම ගණයට ගැනේ. "Sri Lanka", "Low" මේ සඳහා උදාහරණ කිහිපයකි. මෙම වර්ගයේ දත්තයන් සැම විට ම " " යන සංකේත තුළ දැක්විය යුතු ය. සංඛ්‍යා ද 'String' ලෙස දැක්විය හැකි ය. මෙහි දී මෙම සංඛ්‍යා " " යන

සලකුණු තුළ විය යුතු ය. සංඛ්‍යා 'String' ලෙස දැක්වූ විට එම සංඛ්‍යා මත 'Numeric' වර්ගයේ දත්ත මත භාවිත කළ හැකි කර්මයන් යෙදවිය නො හැකි ය. 'String' වර්ගයේ දත්තයන් මත කළ හැකි කර්මයන් අතරින් 'String Concatenation' ප්‍රධාන තැනක් ගනියි. මේ සඳහා '&' යන සංකේතය භාවිත කරන අතර මේ මගින් මෙම වර්ගයේ දත්තයන් එකිනෙකට ඈඳීම සිදු කළ හැකි ය.

උදාහරණ : "My name is" & "Saman" යන ප්‍රකාශනයේ & කර්මය සිදු කළ විට "My name is Saman" යන 'String' එක ප්‍රතිඵලය වෙයි.

Currency - මූල්‍යමය අගයන් කළමනාකරණය සඳහා අර්ථ දක්වා ඇත. මෙම වර්ගයේ අගයයන් සඳහා දශම ස්ථානයට පසු ව ඉලක්කම් හතරක් භාවිත කළ හැකි ය.

Date - දිනයන් හා වේලාවන් කළමනාකරණය සඳහා අර්ථ දක්වා ඇති දත්ත පංතියකි.

Variant - ඉහත සියලු වර්ගයන්ගේ දත්ත මෙම පංතියට ගැනේ.

1.7 විචල්‍ය (Variables)

ක්‍රම ලේඛයක් දත්ත මත ක්‍රියාකරවීමේ දී අදාළ දත්ත ප්‍රධාන මතකයේ (RAM) තබා ගැනීමට අවශ්‍ය වෙයි. මෙසේ දත්ත තාවකාලික ව මතකයේ තබා ගැනීම සඳහා විචල්‍ය (Variables) භාවිත කළ හැකි ය. පරිගණකයේ ප්‍රධාන මතකයේ නිශ්චිත කොටසකට ප්‍රවේශ වීම සඳහා උපයෝගි කර ගත හැකි සංකේත නාමයක් ලෙස විචල්‍යයක් දැක්විය හැකි ය. මෙලෙස, මතකයේ කිසි යම් කොටසකට විචල්‍යයක් අනුබද්ධ කළ විට එම විචල්‍යය උපයෝගි කර ගෙන එම අදාළ මතක කොටසේ විවිධ දත්ත ගබඩා කිරීම හා නැවත ලබා ගැනීම සිදු කළ හැකි ය.

උදාහරණ 1

A = 25 යන ප්‍රකාශනය සලකා බලමු. මෙවැනි ප්‍රකාශනයක් ක්‍රමලේඛයක අඩංගු කළ විට එය පරිගණකයට කිසියම් විධානයක් ලබා දෙයි. මෙම විධානය ක්‍රියාකරවීමේ දී සිදු වන කාර්යයන් පහත දැක්වෙයි.

01. පරිගණකයේ ප්‍රධාන මතකයේ කොටසක් A නම් වූ විචල්‍යය සඳහා ලබා ගැනීම.

02. මෙම කොටසට A යන සංකේත නාමය තාවකාලික ව අනුබද්ධ කිරීම.

03. A යනුවෙන් හඳුන්වා දුන් මතක කොටසේ 25 යන අගය ගබඩා කිරීම.

උදාහරණ 2

A = 10

B = 20

C = A + B

ඉහත ප්‍රකාශන පරිගණක ක්‍රමලේඛයක් ලෙස ක්‍රියාත්මක කිරීමේ දී සිදු වන ක්‍රියාවලිය පහත දැක්වෙයි.

01. ඉහත දැක්වූ පරිදි පරිගණකයේ ප්‍රධාන මතකයේ කොටසකට A යන සංකේත නාමය තාවකාලික ව අනුබද්ධ කිරීම හා එම කොටසේ 10 යන අගය ගබඩා කිරීම.

02. මෙලෙස ම මතකයේ වෙනත් කොටසකට B යන සංකේත නාමය අනුබද්ධ කිරීම හා එම කොටසේ 20 යන අගය ගබඩා කිරීම.

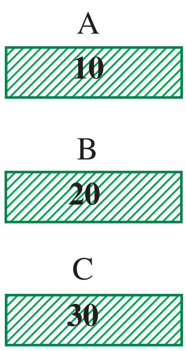
03. A කොටසේ තැන්පත් කර ඇති අගයට B කොටසේ ගබඩා කර ඇති අගය එකතු කර ප්‍රතිඵලය මතකයේ වෙනත් කොටසක ගබඩා කිරීම හා එම කොටසට C යන සංකේත නාමය අනුබද්ධ කිරීම.

ඉහත දැක්වූ විධාන ක්‍රියාත්මක කිරීමේ දී පරිගණකයේ ප්‍රධාන මතකයේ ඇති වන ප්‍රධාන වෙනස්කම් පහත රූප සටහනෙන් දැක්වෙයි.

A = 10

B = 20

C = A+B



විචල්‍ය පිළිබඳ අවබෝධ කර ගත යුතු මූලික තොරතුරු

1. පරිගණකයේ ප්‍රධාන මතකයේ දත්ත තැන්පත් කිරීම සඳහා වෙන් කර ගත් කොටසක සංකේත නාමයක් ලෙස විචල්‍යයක් දැක්විය හැකි ය.

2. විචල්‍යයක් හා අනුබද්ධිත මතකයේ විවිධ අවස්ථාවල දී විවිධ අගයන් තැන්පත් කළ හැකි ය. මෙලෙස එක ම විචල්‍යයට විවිධ අවස්ථාවන්හි විවිධ අගයන් තැන්පත් කිරීමේ දී අවසාන වශයෙන් තැන්පත් කළ අගය මගින් ඊට පෙර තැන්පත් කළ අගය ප්‍රතිස්ථාපනය වෙයි.

   උදාහරණ :

   A = 10
   A = 20

   මෙහි දී පළමු අවස්ථාවේ දී මතකයේ කොටසකට A සංකේත නාමය අනුබද්ධ කර එම කොටසේ 10 යන අගය ගබඩා කරනු ලබයි. දෙවැනි අවස්ථාවේ දී A නම් වූ මතක කොටසේ ඇති අගය 20 යන අගයෙන් ප්‍රතිස්ථාපනය කරනු ලබයි.

3. මතකයේ කොටස් විචල්‍ය සඳහා වෙන් කිරීම තාවකාලික ක්‍රියාවලියකි. ක්‍රමලේඛයක් ක්‍රියාත්මක කර අවසන් වූ විට විචල්‍ය සඳහා ලබාගත් මතකයේ සියලු කොටස් නිදහස් වේ.

විචල්‍ය නම් කිරීම (Naming Variables)

විචල්‍යයක් සඳහා සංකේත නාමයක් යෙදීමේ දී අනුගමනය කළ යුතු නීති සමූහයක් ඇත. මෙම නීති, ක්‍රමලේඛ ලිවීම සඳහා යොදා ගන්නා පරිගණක භාෂාව මත රඳා පවතී.

විෂුවල් බේසික් විචල්‍යයන්ගේ නාමයන් පහත නීතිවලට අනුව තැනිය යුතු ය.

1. ඉංග්‍රිසි හෝඩියේ අකුරකින් ආරම්භ විය යුතු ය.

2. පළමු අක්ෂරයට පසු ව එන අක්ෂර සඳහා ඉංග්‍රීසි හෝඩියේ අකුරු, ඉලක්කම් හෝ '' (underscore) සංකේතය යොදගත හැකි ය.

3. විචල්‍යයක නමට තිබිය හැකි උපරිම අක්ෂර ප්‍රමාණය 255කි.

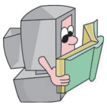
4. විෂුවල් බේසික් විධානයන් සඳහා ලබා දී ඇති නාමයන් විචල්‍ය නාමයන් ලෙස භාවිත නො කළ යුතුයි.

   'VB' විධාන සඳහා ලබා දී ඇති නාමයන් විශේෂිත පද (Reserved Words) ලෙස හැඳින්වෙයි.

ක්‍රියාකාරකම 1.18

පහත දැක්වෙන නාමයන්ගෙන් කුමන නාමයන් 'VB' විචල්‍ය නාමයන් ලෙස භාවිත කළ හැකි ද? භාවිත කළ නොහැකි නාමයන් භාවිත කළ නොහැක්කේ මන්දැයි පැහැදිලි කරන්න.

1) A5
2) MyName
3) My_Name
4) My Name
5) _ Name
6) Name?
7) abc
8) 1f
9) 5A
10) A_5



විචල්‍ය අර්ථ දැක්වීම

වැඩසටහන් සම්පාදනයේ දී අවශ්‍ය විචල්‍යයන්හි දත්ත වර්ග (Data types) වැඩසටහන ආරම්භයේ දී ප්‍රකාශ කළ හැකි ය. මෙවැනි ප්‍රකාශන විචල්‍ය අර්ථ දැක්වීමේ ප්‍රකාශන (Variable Declarations) ලෙස හැඳින්වෙයි.

'VB' විචල්‍යයන් හි දත්ත පංති ප්‍රකාශ කිරීම සඳහා 'Dim' යන රක්ෂිත පදය 'As' යන රක්ෂිත පදය සමඟ යොදා ගැනේ.

උදාහරණ :

```
Dim Value1 As Boolean
Dim  Value2 As Boolean
Dim  Name As String
```

මෙම උදාහරණයේ දී 'Value1' හා 'Value2' යන විචල්‍යයන්හි දත්ත පංතිය 'Boolean' ලෙස ද 'Name' විචල්‍යය 'String' වර්ගයේ විචල්‍යයක් ලෙස ද ප්‍රකාශ කර ඇත.

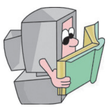
'VB' වැඩසටහනක අර්ථ නොදැක්වූ සියලු විචල්‍ය 'Variant' වර්ගයේ විචල්‍ය ලෙස සලකනු ලබයි.

ක්‍රියාකාරකම 1.19

VB' හි විචල්‍ය භාවිතයට පෙර අර්ථ දැක්වීම අත්‍යවශ්‍ය නො වේ. එනමුත් විචල්‍යයන්හි දත්ත පංති භාවිතයට පෙර ප්‍රකාශ කිරීමෙන් ඇතිවන ප්‍රයෝජන කිහිපයකි. මෙම ප්‍රයෝජන මොනවා දැයි සාකච්ඡා කරන්න.

ක්‍රියාකාරකම 1.20

'Numeric' හා 'String' යන දත්ත වර්ග මත කළ හැකි විවිධ කර්මයන් මොනවා දැයි සොයා බලන්න. මෙවැනි කර්මයන් 'VB' ක්‍රමලේඛයක යෙදීමේ දී අනුගමනය කළයුතු නීති මොනවා දැයි අධ්‍යයනය කරන්න.



ගණිත කර්ම (Mathematical Operators)

+	එකතු කිරීම	(Addition)
-	අඩු කිරීම	(Subtraction)
*	ගුණ කිරීම	(Multiplication)
/	බෙදීම	(Division)
^	බලය	(Power)
\	නිඛිල බෙදීම	(Integer division)
Mod	ශේෂය	(Remainder)

1.8 කර්ම ප්‍රමුඛතා (Operator Precedence)

පහත ගණිතමය ප්‍රකාශය සලකා බලන්න.

5 3 × 2

මෙම ප්‍රකාශය දෙයාකාරයකට සුළු කළ හැකි ය.

5(3×2) = -1 (53) ×2 = 4

මෙවැනි ප්‍රකාශන විවිධ ආකාරයට සුළු කළ විට ලැබෙන ප්‍රතිඵල බොහෝ විට එකිනෙකට වෙනස් වේ. එබැවින් ප්‍රකාශයක විවිධ වර්ගයේ ගණිතමය කර්මයන් ඇතිවිට එම ප්‍රකාශය සුළු කිරීමේ දී කර්මයන් සඳහා දියයුතු ප්‍රමුඛතා අර්ථ දක්වා ඇත. මෙම ප්‍රමුඛතා 'Operator Precedence' ලෙස හැඳින්වෙයි. ගණිතමය කර්මයන් හි ප්‍රමුඛතා පහත දැක්වෙන පරිදි වෙයි.

කර්මය	ප්‍රමුඛතා අනුපිළිවෙළ
(1)	( )
(2)	^
(3)	* , /
(4)	+, -

උදහරණ :

5+2×35∧2

කර්මයන්ගේ ප්‍රමුඛතාවන්ට අනුකූල ව ඉහත ප්‍රකාශය සුළු කල යුතු අයුරු වරහන් යොදා පහත පරිදි පෙන්වා දිය හැකි ය.

((5 +(2 * 3)–(5 ∧ 2))

③ ② ④ ①

= ((5 +(2 * 3)–25)

= ((5 + 6)–25)

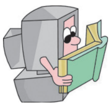
= 11–25

= –14

වරහන් යෙදීමෙන් ප්‍රකාශනයක කර්මයයන් සඳහා ලබා දී ඇති ප්‍රමුඛතා වෙනස් කළ හැකි ය.

ක්‍රියාකාරකම 1.21

වරහන් යෙදීමෙන් ප්‍රකාශයක අඩංගු ගණිත කර්මයන්හි ප්‍රමුඛතා වෙනස් වන අයුරු සාකච්ඡා කරන්න.



කොන්දේසිගත කර්ම (Conditional Operators)

මෙවැනි කර්ම ප්‍රකාශ මත ක්‍රියා කළ විට ලැබෙන ප්‍රතිඵලය බූලිය (True or False) අගයක් වෙයි.

කර්මය		තේරුම
=	සමාන	(Equal to)
>	විශාල	(Greater than)
<	කුඩා	(Less than)
>=	විශාල හෝ සමාන	(Greater than or equal to)
<=	කුඩා හෝ සමාන	(Less than or equal to)
<>	අසමාන	(Not equal to)

උදාහරණ :

ප්‍රකාශන	ප්‍රතිඵලය
1) 5 = 6	False
2) 3 + 5 = 4 + 4	True
3) 5 < 6	True
4) 6 < 5	False
5) 5 < > 6	True
6) 5 < = 6	True
7) 6 < = 5	False

මෙවැනි කොන්දේසිගත කර්ම ප්‍රකාශ මත ක්‍රියාත්මක කිරීමේ දී, එම ප්‍රකාශයේ කර්මයට දකුණුපස හා වම්පස සංකීර්ණ ප්‍රකාශ වූ විට එම ප්‍රකාශ තනි අගයකට සුළු කෙරේ. ඉන් පසු මෙම තනි අගයයන් මත අදළ කර්මය ක්‍රියාත්මක කරවා නියමිත බූලිය ප්‍රතිඵලය ලබාදෙයි.

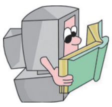
උදාහරණ ඃ

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

මෙම ප්‍රකාශය පළමු ව පහත දැක්වෙන ලෙස තනි පද සහිත ප්‍රකාශනයකට සුළු කෙරේ.

$$17 > 15$$

17 යන සංඛ්‍යාව 15ට වඩා විශාල නිසා මෙම ප්‍රකාශයේ ප්‍රතිඵලය True ලෙස ලබා දෙයි.



තාර්කික කර්ම (Logical Operators)

මෙම වර්ගයේ කර්මයන් තාර්කික (Logical) දත්ත වර්ගය මත අර්ථ දක්වා ඇති කර්මයන් වෙයි. මූලික තාර්කික කර්ම වර්ග තුනකි. මෙම කර්මයන් තුන 'AND', 'OR' සහ 'NOT' වශයෙන් හඳුන්වා දී ඇති අතර ඒවායේ අර්ථයන් පහත දැක්වෙන සත්‍යතා වගු (Truth Tables) මගින් දැක්වෙයි.

(බූලීය විචල්‍ය හා කර්මයන් පිළිබඳ පැහැදිලි කිරීමක් 10 වන ශ්‍රේණියේ 4 වන පරිච්ඡේදයේ දක්වා ඇත.)

පහත සත්‍යතා වගුවේ 'a' හා 'b' යනු බූලිය විචල්‍ය වෙයි.

AND

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

උදහරණ ඃ

(2 > 3) AND (3 > 2) යන ප්‍රකාශය සලකන්න.

මෙහි AND යන කර්මය ක්‍රියාත්මක කළ හැක්කේ බූලීය වර්ගයේ අගයයන් මත පමණක් බැවින් පළමු ව ප්‍රකාශයේ දකුණුපස සහ වම්පස සුළුකර බූලීය අගයයන් ලබාගත යුතු වෙයි. 2 > 3 යන්න 'False' වන අතර 3 > 2 යන්න 'True' වෙයි. ඉන් පසු 'False AND True' යන ප්‍රකාශයේ ප්‍රතිඵලය ඉහත සත්‍යතා වගුවට අනුව 'False' විය යුතු ය. එබැවින් ඉහත ප්‍රකාශයේ අවසාන අගය බූලීය වන අතර එම අගය 'False' ලෙස දැක්විය හැකි ය.

OR

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

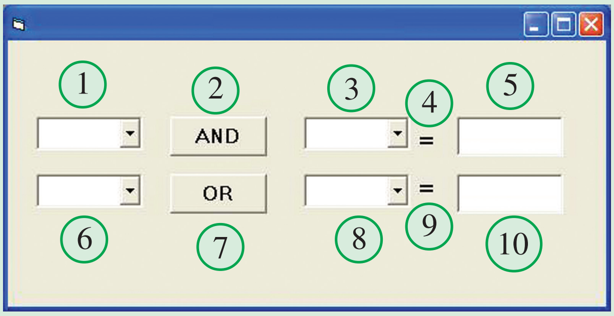
NOT

a	NOT a
True	False
False	True

ඉහත වගුවලින් දක්වා ඇති පරිදි 'AND' සහ 'OR' ද්විමය කර්මයන් වන අතර 'NOT' යන්න ඒකමය කර්මයක් වෙයි.

ක්‍රියාකාරකම 1.22

AND සහ OR යන බූලිය කර්මයන්ගේ ක්‍රියාකාරිත්වය පෙන්වීම සඳහා පහත දැක්වෙන පරිදි අතුරු මුහුණතක් ඇති වැඩසටහනක් සම්පාදනය කරන්න.



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, 5, හා 7 යන පාලනවල 'Data Format' ලක්ෂණය බූලීය ලෙස ද 'List' ලක්ෂණය සඳහා 'True' සහ 'False' අගයන් ද ලබා දෙන්න. 'Combo Box' හි 'List' ලක්ෂණය සඳහා අගයන් කිහිපයක් ලබා දීමේ දී එක් අගයක් ඇතුළත් කිරීමෙන් පසු 'Ctrl + Enter' යතුරු තද කිරීමෙන් ඊළඟ අගය ඇතුළත් කළ හැකි ය.

ඉහත cmdAnd හා cmdOr විධාන බොත්තම්වල Click Events සඳහා පහත දැක්වෙන කේත ඇතුළත් කොට වැඩ සටහන ක්‍රියාත්මක කරන්න.

Program 9

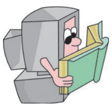
```
Private Sub cmdAnd_Click()
        Dim LeftAndValue As Boolean
        Dim RightAndValue As Boolean
        Dim ResultAnd As Boolean
        LeftAndValue = cboAndLeft.List(cboAndLeft.ListIndex)
        RightAndValue = cboAndRight.List(cboAndRight.ListIndex)
        ResultAnd = LeftAndValue And RightAndValue
        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
```

වැඩ සටහන ආරම්භ කළ විගස And හෝ Or විධාන බොත්තමක් මත ක්ලික් කර සිදු වන දේ සහ එසේ වන්නේ ඇයි දැයි සටහන් කරන්න.

'Combo Boxes' වල වූ අගය වෙනස් කරමින් And සහ Or විධාන බොත්තම් මත ක්ලික් කිරීමේ දී ලැබෙන පිළිතුරු සටහන් කරන්න.

දැන් පහත කේත කොටස එක්කර නැවත වැඩ සටහන ඉහතින් සඳහන් පරිදි ක්‍රියාත්මක කර දකින්නට ලැබෙන වෙනස්කම් සටහන් කරන්න.

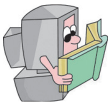
```
Private Sub Form_Load()
        cboAndLeft.ListIndex = 0
        cboAndRight.ListIndex = 0
        cboOrLeft.ListIndex = 0
        cboOrRight.ListIndex = 0
End Sub
```



ගැලීම් පාලනය (Flow Control)

ක්‍රමලේඛයක ඇති ප්‍රකාශ ක්‍රියා කෙරෙන අනුපිළිවෙළ ගැලීම් පාලනය (Flow Control) ලෙස හැඳින්වෙයි. සාමාන්‍ය වශයෙන් ක්‍රමලේඛයක ඇති ප්‍රකාශ පළමු ප්‍රකාශයෙන් ආරම්භ කර ක්‍රමලේඛයේ දක්වා ඇති අනුපිළිවෙළට අනුව ක්‍රියාත්මක කෙරේ. මෙම අනුපිළිවෙළ වෙනස් කිරීම සඳහා විවිධ පාලන ව්‍යුහයන් (Control Structures) යොදා ගත හැකි ය.

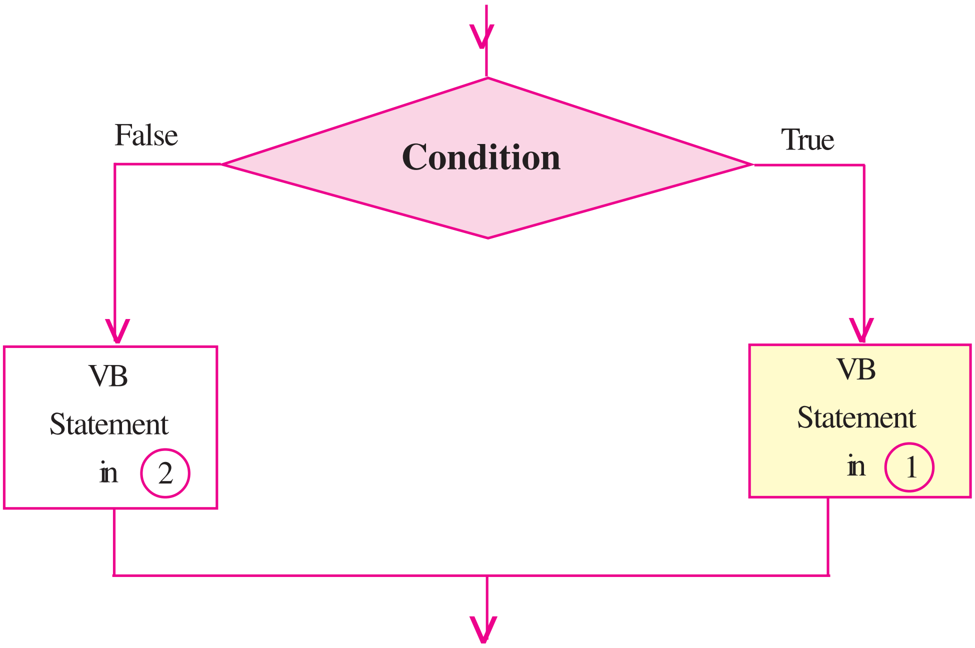
මෙම පාලන ව්‍යුහයන් යොදාගැනීමේ දී අනුගමනය කළ යුතු නීතිමය ආකෘති වෙයි. මෙම ආකෘති එම ව්‍යුහයන්ගේ කාරක රීති (Syntax) ලෙස හැඳින්වෙයි.



IF....Then.....Else ව්‍යුහය.

```
If Condition Then
        VB  Statements  ———————— (1)

Else
        VB  Statements  ———————— (2)

End IF
```

ගැලීම් සටහන



මෙම ආකෘතිය සඳහා යොදා ඇති 'If', 'Then', 'Else', 'End If' යන වචන 'VB', විශේෂිත පද වෙයි.

'Condition' යනුවෙන් දක්වා ඇති 'VB' ප්‍රකාශනයේ අවසාන ප්‍රතිඵලය බූලීය අගයක් විය යුතු ය.

'VB' Statements යනු 'VB' ප්‍රකාශයන්ගේ එකතුවකි.

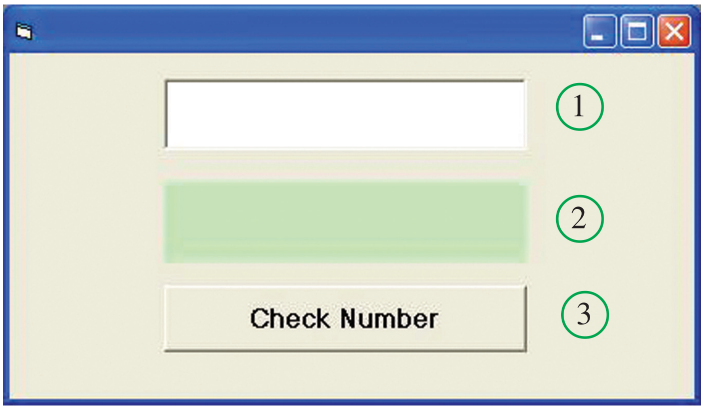
මෙම ව්‍යුහයේ ක්‍රියාකාරිත්වය පහත දැක්වෙන පරිදි දැක්විය හැකි ය.

a) 'Condition' ප්‍රකාශය සුළු කර අවසාන බූලීය අගය ලබාගන්න.

b) මෙම අගය 'True' නම් 1 කොටසෙන් දක්වා ඇති 'VB' ප්‍රකාශ ඒ දක්වා ඇති අනුපිළිවෙළට ක්‍රියාත්මක කරන්න.

c) 'Condition' ප්‍රකාශයේ අගය 'False' නම් 2 කොටසෙහි දක්වා ඇති ප්‍රකාශ ඒ දක්වා ඇති අනුපිළිවෙළට ක්‍රියාත්මක කරන්න.

Program 10 - සංඛ්‍යාවක් ඔත්තේ ද ඉරට්ටේ ද යන්න සොයා බැලීමට ක්‍රමලේඛයක් සම්පාදනය කරමු.



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
```

1.9 Select.. Case

මෙම ව්‍යුහය IF....Then.....Else ව්‍යුහයන්ගේ එකතුවක් ලෙස දැක්විය හැකි ය.

Select.. Case හි ව්‍යුහය

```
        Select Case expression ———— (a)
        Case Value 1
            VB Statements ———— (1)
        Case Value 2
            VB Statements ———— (2)
            ..............................................
            ..............................................
        Case Else
            VB Statements ———— (n)
        End Select
```

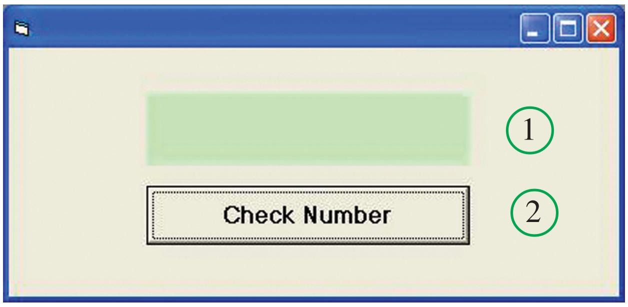
ක්‍රියාකාරිත්වය

'Select Case', 'Case', 'Case Else', 'End Select' මෙහි විශේෂිත පද වෙයි. 'Expression' හි දත්ත වර්ගය (Data type) 'Value1', 'Value2'... දත්ත වර්ගයන් හා එකඟ විය යුතු ය. මෙහි 'a' යනුවෙන් දක්වා ඇති 'expression' සුළු කර ලැබෙන අගය 'Value 1' සමඟ සසඳා බැලේ. එය සමාන වූ විට 1 දක්වා ඇති කොටස ක්‍රියා කෙරේ. මෙසේ නොවුණ හොත් එම අගය 'Value2' සමඟ ද, 'Value3'.......... අගයයන් සමඟ ඒ

අනුපිළිවෙළට සසඳා බැලේ. කිසියම් 'Value' හි අගයකට 'expression' හි අවසාන ප්‍රතිඵලය සමාන වූ විට එම 'Value' කොටසට පහළින් වූ 'VB Statements' ක්‍රියා කරවනු ලබයි.

'a' හි අගය කිසි ම 'Value1', 'Value2', ... අගයකට සමාන නොවුණ හොත් අවසාන 'Case Else' කොටසට පහළින් ඇති 'n' වශයෙන් දක්වා ඇති කොටසෙහි වූ 'VB' ප්‍රකාශ ක්‍රියාත්මක වෙයි.

Program 11 - සංඛ්‍යාවක් ධන, සෘණ හෝ බිංදුව ලෙස සොයා ගැනීම සඳහා ක්‍රමලේඛයක් සම්පාදනය කරමු.



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
```

ක්‍රියාකාරකම 1.23

Select .... Case හි ක්‍රියාකාරිත්වය දැක්වීම සඳහා ගැලීම් සටහනක් නිර්මාණය කරන්න.

1.10 පුනර්කරණය (Looping)

ක්‍රමලේඛයක ඇති ප්‍රකාශ එකතුවක් නැවත නැවතත් ක්‍රියාත්මක කිරීම සඳහා යොදගත හැකි ව්‍යුහයන් මෙම ගණයට (Looping Structures) ගැනේ.

For ... Next පුනර්කරණය - ආකෘතිය

For විචල්‍යය = ආරම්භක අගය To අවසාන අගය [Step අඩු හෝ වැඩි කරන අගය]

අදාළ කේත කොටස්

Next විචල්‍යය

මෙහි ඇති [Step අඩු හෝ වැඩි කරන අගය] යන කොටස අවශ්‍ය අවස්ථාවල දී පමණක් භාවිත කළ හැකි විකල්පයකි.

For .... Next පුනර්කරණය - ක්‍රියාත්මක වන ආකාරය

For ..... Next පුනර්කරණයේ දී මුලින් ම එහි සඳහන් කර ඇති විචල්‍යයට අදාළ අගය ආදේශ කර ගැනීම සිදු කරයි. මෙහි දී අගය ආදේශ කර ගැනීම ආරම්භක අගයේ සිට අවසාන අගය දක්වා පිළිවෙළින් එකින්එක වැඩි වන ආකාරයට සිදු කරයි. Step සමඟ යම් අගයක් ලබා දී තිබුනේ නම් අගය දෙකක් අතර පරතරය සකස් කිරීම සඳහා එම අගය භාවිත කරයි. (එනම් සාමාන්‍යයෙන් For ..... Next පුනර්කරණයේ දී අගය වැඩි වීම ආරෝහණ පිළිවෙළට එකින්එක වැඩි වන අතර වෙනස් අගයවලින් අගය වැඩි කිරීම හෝ අඩු කිරීම සඳහා Step විකල්පය භාවිත කළ හැකි ය.)

අවරෝහණ පිළිවෙළට ගමන් කරන පුනර්කරණයක් සකස් කිරීමේ දී අනිවාර්යයෙන් ම මෙම Step අගය භාවිත කළ යුතු ය.

විචල්‍යය සඳහා ආදේශ කළ අගයත් සමඟින් ඉදිරියට ඇදෙන පුනර්කරණය එහි වූ කේත කොටස් ක්‍රියාත්මක කරන අතර Next පදය/පේළිය හමු වූ විගස නැවත වරක් For පේළිය වෙත ගමන් කරයි. එහි දී අදාළ විචල්‍යයේ අගය අවසන් අගය දැයි පරීක්ෂා කරන අතර එය අවසන් අගය නම් පුනර්කරණය අවසන් කරමින් Next පදයෙන්/පේළියෙන් පසු ව ඇති පේළිය වෙත වැඩ සටහන ගමන් කරවයි. එසේ නොවන විට දී විචල්‍යයේ අගය අදාළ පරිදි වෙනස් කොට පුනර්කරණය තුළට ගමන් කරයි. එනම් නැවතත් එහි වූ විධාන ක්‍රියාත්මක කරමින් ඉදිරියට ගමන් කරයි. Next පදය/පේළිය හමු වූ විගස නැවත වරක් For පේළිය වෙත ගමන් කරමින් විචල්‍යයේ අගය පරීක්ෂා කිරීම සිදු කරයි. මෙසේ මෙම ක්‍රියාව විචල්‍යයේ අගය අවසන් අගය වන තෙක් සිදු වනු ඇත.

කිසියම් නිශ්චිත වාර ගණනක් යම් ක්‍රියාවලියක් ක්‍රියාත්මක කිරීම සඳහා ඉතා ම පහසුවෙන් For . ... Next පුනර්කරණය යොදා ගත හැකි ය.

උදාහරණ : පහත දැක්වෙන ක්‍රමලේඛ කොටස් සලකා බලන්න.

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

මෙහි දී 0 සිට 10 දක්වා අගය එකින් එක වැඩිවෙමින් (පිළිවෙළින්) පෝරමයේ පහළට මුද්‍රණය වන අතර අවසානයේ එම අගයයන්වල එකතුව 55 යන අගය ද මුද්‍රණය වනු දැකිය හැකි ය.

b)
```
Sum= 0
For i= 0 to 10 Step 2
        Sum= Sum + i
        Print i
Next
Print Sum
```

මෙහි දී 0 සිට 10 දක්වා අගය දෙකෙන් දෙක වැඩිවෙමින් (එනම් 0, 2, 4, 6, 8, සහ 10) පෝරමයේ පහළට මුද්‍රණය වන අතර අවසානයේ එම අගයයන්වල එකතුව 30 යන අගය ද මුද්‍රණය වනු දැකිය හැකි ය.

c) Sum= 0

```
For i= 10 to 0 Step -2
        Print i
Next
```

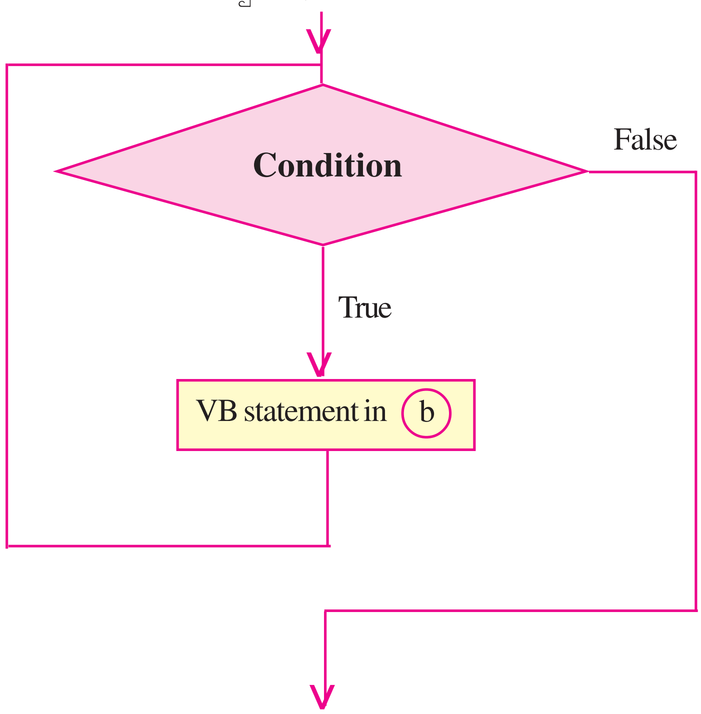
මෙහි දී 10 සිට 0 දක්වා අගය දෙකෙන් දෙක අඩුවෙමින් (එනම් 10, 8, 6, 4, 2 සහ 0) පෝරමයේ පහළට මුද්‍රණය වනු දැකිය හැකි ය.

Do While Loop

ව්‍යුහය

```
Do While <condition>          —— a
        VB Statements         —— b
Loop
```

Do While හි ලබා දී ඇති කොන්දේසිය (<condition>) සත්‍ය වන අවස්ථාවල දි පුනර්කරණය තුළට ගමන් කර එහි වූ ප්‍රකාශන (b) ක්‍රියාත්මක කරයි. අදළ කොන්දේසි අසත්‍ය වූ විට පුනර්කරණයෙන් ඉවතට ගමන් කරයි. එනම් Loop ප්‍රකාශනයෙන් පසු ව වූ ප්‍රකාශන වෙත ගමන් කිරීම සිදු වේ.



මෙහි 'Do While, Loop' යන වචන විශේෂිත පද වෙයි. 'Condition' යන ප්‍රකාශය බූලීය අගයක් ලබා දිය යුතු ප්‍රකාශනයකි.

ක්‍රියාකාරකම 1.24

නව ව්‍යාපෘතියක් ආරම්භ කර එහි Form එකකට Command button එකක් එක්කර එහි Click event එක සඳහා පහත කේත ඇතුළත් කොට එය ක්‍රියාත්මක කළ විට ලැබෙන ප්‍රතිඵලය නිරීක්ෂණය කරන්න.

උදහරණ

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

අදාළ Command button එක මත Click කළ විට දී එහි ප්‍රතිඵලය දක්වනුයේ Immediate Window එක මත ය. පහත ක්‍රියාකාරකම්වල දී ඇති වන වෙනස් කම් නිරීක්ෂණය කරන්න.

a. debug.Print counter යන්න Print counter යනුවෙන් වෙනස් කළ විට දී.

b. debug.Print counter
counter = counter + 1 යන්න counter = counter + 1
debug.Print counter
යනුවෙන් වෙනස් කළ විට දී.

1.11 අරාවන් (Arrays)

'VB' පරිගණක භාෂාවේ ද අනික් බොහෝ පරිගණක භාෂාවල මෙන් ම දත්තයන් ක්‍රමලේඛයන් තුළ තාවකාලික ව ගොනු කළ හැකි ක්‍රමවේදයන් හඳුන්වා දී ඇත. මෙසේ තාවකාලික ව ක්‍රමලේඛයක් තුළ දත්ත පෙළ ගැස්වීමෙන් සාදගන්නා දත්ත ව්‍යුහයන් 'Data Structures' ලෙස හැඳින්වෙයි. 'Arrays, Stacks', 'Queues' මේ සඳහා උදහරණ කිහිපයකි.

'VB' හි අරාවන් (Array) යනු එක ම වර්ගයේ දත්ත සමූහයක් පෙළගැස්වීමෙන් සාදගන්නා දත්ත ව්‍යුහයකි. මෙවැනි දත්ත ව්‍යුහයක් පහත දක්වෙන පරිදි දෘෂ්‍යකරණය (Visualize) කළ හැකි ය.

20	30	40	55	60	**Marks**
1	**2**	**3**	**4**	**5**	

ඉහත දත්ත ව්‍යුහය නිඛිල සංඛ්‍යා පහක් තාවකාලික ව පෙළගැස්වීම සඳහා නිර්මාණය කළ අරාවක් ලෙස දක්විය හැකි ය.

මෙවැනි දත්ත ව්‍යුහයක් ක්‍රමලේඛයක් තුළ නිර්මාණය කිරීමේ දී මුළු දත්ත ව්‍යුහය ම දක්වීම සඳහා කිසියම් නාමයක් ලබා දිය යුතු ය. ඉහත උදහරණයේ දී අරාවන් සඳහා ලබා දුන් නාමය 'Marks' යයි සලකමු. ඉන් පසු එම දත්ත ව්‍යුහයේ එක්එක් අගයන් ලබා ගැනීම හෝ ගබඩාකිරීම සඳහා 'Marks (Position)' යන ව්‍යුහය භාවිත කළ හැකි ය.

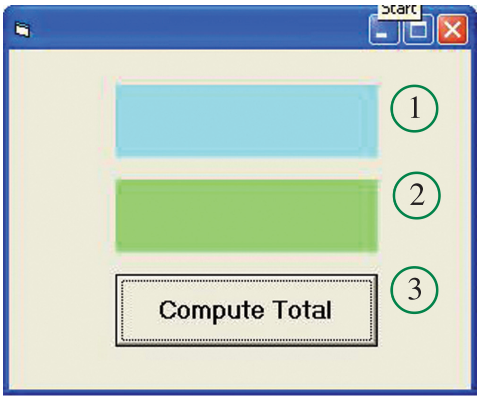
මෙම ව්‍යුහයේ පළමු කොටසින් (උදා ඃ Marks) නම දක්වෙයි. දෙවනු ව, වරහන් තුළ වූ කොටසින් (උදා ඃ Position) අරාවන්හි පළමු අගයට සාපේක්ෂ ව ලබා ගත යුතු අගයේ ස්ථානය දක්වෙයි.

උදාහරණ ඃ

Dim Marks (0 to 9) As Integer

ඉහත ප්‍රකාශය මගින් 'Marks' නම් වූ අරාවක් නිර්මාණය කර ඇත. මෙම අරාවෙන්හි පළමු අගය 'Marks (0)' ලෙස ද අවසාන අගය 'Marks (9)' ලෙස ද හැඳින්වීමට අවශ්‍ය බව දක්වා ඇත. එබැවින් මෙම 'Marks' නම් වූ අරාව තුළ අගයයන් 10ක් ගබඩා කළ හැකි ය.

උදහරණක් ලෙස ඕනෑ ම සංඛ්‍යා 10ක එකතුව සෙවීම සඳහා ආරාවක් මත පදනම් වූ වැඩසටහනක් පහත දක්වෙන පරිදි නිර්මාණය කළ හැකි ය. මෙම වැඩසටහනේ අතුරු මුහුණත පහත දක්වෙන ලෙස සකස් කර ඇත.



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 As Integer
        Dim total As Integer
        Dim numbers As String
        ‘No Numbers entered yet
        numbers = “”
        ‘ Read the 10 numbers to number array
        For i = 1 To 10
        myNumbers(i) = Val(InputBox(“Enterthe number “ & i))
            lblNumbers.Caption = lblNumbers.Caption & myNumbers(i) & “  ,  ”
        Next i
        ‘Compute the total of numbers in the array
        total = 0
        For i = 1 To 10
        total = total + myNumbers(i)
        Next i
        ‘Display the total of numbers in the array
                lblTotal.Caption = “Total of the numbers is “ & total
End Sub
```