

# **Java**

# **Myanmar**

# **Learning**

**For Friends**

**Created by Thureanoo**

**Academy 2**

## JAVA Developer Guide

### 1.3 Statements Expression and Operators

#### Statements and Expression

Statements ဆိတာက JAVA Programming Language မှုသာမက Programming Language တိုင်းမှာလိုလို ပါဝင်ပါတယ်။ အခြေခံသဘာတရားကတော့ အတူတူပါပဲ။ ဒီအခန်းကဏ္ဍမှာဆိုရင် လက်နဲ့ချွောက်ရမယ့်အပိုင်းတွေ runလုပ်ဖြီး လေ့လာရမယ့် sample code တွေ၊ Listing တွေ ပါဝင်မှန်ညီးဖြီးထပ်ခါတလဲလဲ ကျတ်မှတ်ဖတ်ရှုရမယ့် theory သဘာတရားတွေပဲ ပါဝင်တာလို့ စနစ်တကျ ဖတ်မှတ်ပြီး လေ့လာမှသာလျှင် JAVA Language ကို တကယ်တတ်မြောက်မှာဖြစ်ပါတယ်။ ကဲ Statements ဆိုတာ ဘာလဲဆိုတာကို အောက်မှာ ရှင်းပြထားပါတယ်။ လေ့လာလိုက်ပါဉိုး။

JAVA Language မှာ Statements တွေကို အောက်ပါအတိုင်း အမျိုးအစားခွဲလိုက်လို့ ရပါတယ်။

#### Control Statements

##### Evaluates Statements (expression)

##### Null Statements

ကဲ ဒီခေါင်းစဉ်တွေကို တစ်ခုချင်းစီ လေ့လာကြတာပေါ့။ အောက်မှာ Control Statements အကြောင်းကစပြီး လေ့လာလိုက်ပါဉိုး။

#### Control Statements

Control Statements ဆိတာကတော့ ကွန်တော် ရှိမှာ သင်ကြားပေးခဲ့ပြီးပါ၌။ if, else-if, case, for, while, while-do, break, continue စိတ် control structure တွေကို အသုံးပြုပြီး ရေးသားရတဲ့ statements တွေကို ခေါ်တာပါ။ ဒီ statements တွေဟာ program flow ကို ထိန်းချုပ်ရမှာ အမိကအသုံးပြုပါတယ်။ ဥပမာအနေနဲ့ ပြောရရင်တော့

if(ဖြစ်ခဲ့မှသာလျှင်) —————> ဒီအလုပ်ကိုလုပ်။

while(စဉ်မှာ ) —————> { ဒီအလုပ်ကို လုပ်။ }

စသဖြင့်ပေါ့။ Program flow ကို ထိန်းချုပ်တယ်ဆိုတာကတော့ if(ဖြစ်ခဲ့မှသာလျှင်)တို့၊ while(စဉ်မှာ)ဆိုတဲ့ control structure တွေကြောင့်မဖြစ်ခဲ့လျှင် escape ဖြစ်ပြီး နောက်တစ်ကြောင်းကို ဆင်းသွားတာတို့ block အပြင်ကို ထွက်သွားအောင်တို့ ရည်ရွယ်ပြီး ရေးသားအသုံးပြုရတဲ့အတွက် control statement လို့

ခေါ်တာပါ။ Listing 1.3 က စွဲးတော့ ရေးပြထားပါတယ်။ နောက် chapter နဲ့ Listing တွေမှာလည်း ထည့်သွင်းရေးသားရှင်းပြေားမှာပါ။ Exam; တွေမှာ objective type အနေနဲ့ မေးလာခဲ့ရင် ဖြေဆိုနိုင်အောင်လို့ control statements ဆိုတာ ဘာလဆိုတာကို သေသေချာချာ ရှင်းပြရတာပါ။ လိုအပ်ရင် Listing 1.3 က စွဲးတော့ ဖြန့်လေ့လာကြည့်ပါ။ ကဲ အောက်မှာ evaluates (expression) statements ကို ရှင်းပြထားပါတယ်။ လေ့လာလိုက်ပါဉိုး။

#### Evaluates (expression) Statements

Expression ဆိတာကတော့ ရှိုက Listing တွေကို လေ့လာခဲ့ပြီးရင် အခေါ်အစောင့် မစိမ်းလောက်ပါဘူး။ အဂွယ်ဖြောရရင်တော့ expression ဆိုတာ program ကို အလုပ် ဆက်လုပ်သင့်/မလုပ်သင့်ကို စိစစ်တဲ့ control statement ရဲ့ တစ်စိတ်တစ်ခေါ်လို့ ပြောရင်လည်း မမှားပါဘူး။ ကဲ နားလည့်သွားအောင် အောက်မှာ ရေးပြထားတာကို လေ့လာကြည့်ပါ။

#### sample

```
if(marks = >40)           System.out.println
[expression]                ("PASS!");
                           [statements]
control statements
```

ဒါကို အသုံးပြုပြီးရေးထားတဲ့ ဒီ control statements ထဲမှာ marks ဟာ 40နဲ့ ညီးညီးလို့လျှင်ဆိုတဲ့ expression ပါဝင်ပါတယ်။ ဒီ expression လေး မှုန်မှသာလျှင် နောက်က statements ကို အလုပ်ဆက်လုပ်လို့ရမှာမဟုတ်လား။ ဒါကြောင့် expression statements ကို Evaluates Statement လို့လည်း ခေါ်ခေါ်တာပေါ့။

**Note:** Expression တွေကို ရေးတဲ့အခါမှာ program စတင်ရေးစဉ်မှာ declareလုပ်ခဲ့တဲ့ constant, variable, array စတာတွေကို operator တွေနဲ့ ချိတ်ဆက်ရေးသားခြင်းပဲဖြစ်ပါတယ်။ ဥပမာ အပေါ်မှာရေးပြခဲ့တဲ့ sample code မှာဆိုရင် marks ကို int date type ကို အသုံးပြုပြီးတော့ ကြည်းရမှာပါ။ ဒီ marks ကို Relational operator ဖြစ်တဲ့ Greater Than or Equals Sign ကို အသုံးပြုပြီးတော့ မှန်မမှန် စစ်ဆေးတာမဟုတ်လား။ ဒါကို test expression လုပ်တယ်လို့ ခေါ်ပါတယ်။ ကဲ ဒါဆိုရင်

## ကွန်ပျိုတာသင်ခန်းစာ

တော့ expression statements အကြောင်းကို နားလည် သဘောပါက်သွားလောက်ပါပြီ။

### Null Statement (the do nothing operator)

Null Statements ဆိတာကတော့ program flow ကို ထိန်းချုပ်ထားတဲ့ control statements တွေကို ရုပ်နားသွားဖော့တဲ့ ရေးသားနည်းပါပဲ။ အမှန်ကတော့ semicolon(;) ကို စနစ်တကျ အသုံးပြခြင်းပါပဲ။ semicolon (;) ဆိတာ JAVA Language မှာ program code ကို တစ်ကြောင်းရေးသားပြီးတိုင်း end sign အဖြစ် ရေးသားပေးရပေမယ့် Null Statements အဖြစ် ရေးသားလိုလည်း ရပါတယ်။ ကဲ အောက်မှာလေ့လာလိုက်ပါပြီး။

#### sample

```
if (SomeValue<10 ); // null statement
SomeValue = 10;

ဒီ code ရဲ့ အမိဘယ်ကတော့ SomeValue ဟာ 10 ထက် ငယ်ခဲ့ရင်ဆိတဲ့ test expression ကို if control structure ကို အသုံးပြုပြီးတော့ ရေးသားထားပါတယ်။ SomeValue ဟာ 10 ထက် ငယ်တယ်ဆိတာနဲ့ တစ်ခြိုင်နှင်း အောက်တစ်ကြောင်းကို ဆင်းသွားပြီး SomeValue ဟာ 10 နဲ့ပဲ အလုပ်ဆက်လုပ်ပါတယ်။ အောက်မှာပါဝင်တဲ့ code-Line တွေကို အလုပ်ဆက်လုပ်နေသရွာ တော့ SomeValue ကို 10 အနေနဲ့ပဲ Setလုပ်ပြီး တွက်ချက်သွားတော့မှာပါ။ ဘာကြောင့်လေဆိတော့ if Statement ဟာ expression နောက်မှာ ရေးထားတဲ့ Semicolon (;) ကိုတွေ့တာနဲ့ အလုပ်လုပ်နိုင်စွမ်းမရှိတော့ဘဲ terminateဖြစ်သွားပါပြီ။ အလုပ်ဆက်လုပ်နိုင်စွမ်းမရှိတော့ဘဲနဲ့ အောက်တစ်ကြောင်းတို့ ဆင်းလာရပါပြီ။ ဒီလိုဖြစ်အောင်ရေးတာကို Null Statements နည်းနဲ့ရေးတယ်လို့ ခေါ်သလို do-nothing operator နဲ့ ရေးတယ်လိုလည်း ခေါ်ပါတယ်။ အပေါ် sample ကို နောက်တစ်မျိုးရေးပြုပါပြီးမယ်။ လေ့လာလိုက်ပါပြီး။
```

#### sample

```
if (SomeValue<10) // test
;
// do-nothing
SomeValue = 10; // assign

if statement ဟာ အပေါ်ဆုံးစာကြောင်းမှာတော့ control လုပ်ပြီး test expression လုပ်နိုင်စွမ်းမရှိပေမယ့် နောက်တစ်ကြောင်းကို ရောက်တဲ့အခါမှာတော့ Semicolon(;) ကို တွေ့တဲ့ အခါမှာ do-nothing operator အနေနဲ့ ရေးထားတာဖြစ်တဲ့ အတွက် အလုပ်လုပ်နိုင်စွမ်းမရှိတော့ဘဲနဲ့ အောက်မှာရေးထားတဲ့ code ကို ဆက်ဆင်းလာရပါတယ်။ ဒီအခါမှာ SomeValue ရဲ့ တန်ဖိုးဟာ အပေါ်မှာ ဘယ်လောက်ပဲဖြစ်ခဲ့ဖြစ်ခဲ့ SomeValue = 10 လို့ assign လုပ် ကြညာလိုက်ပါတယ်။ နောက်ထပ် ဆက်တွက်ချက်နေသရွာ ဒီတန်ဖိုးကိုသာလျှင် အခြေခြားပြီး တွက်ချက်
```

သွားတော့မှာပါ။ အပေါ် sample နဲ့ အောက်က sample code ဟာ အတူတူပါပဲ။ ကဲ ဒါဆိုရင်တော့ Null Statement အကြောင်းကို နားလည်းသောပါက်သွားလောက်ပါပြီ။ Statements နဲ့ Expression အကြောင်းကို နားလည်းသောပါက်သွားပြီ ဆိုတော့ operators အကြောင်းကို ဆက်လက် လေ့လာကြပါပြီးနှင့်။

**Note:** ဒီနေရာမှာ ပြောစရာရှိတာကတော့ အချို့စာအုပ်မှာ learn လုပ်တဲ့အခါ Expression နဲ့ Statements ကို ခွဲမရေား Statements Expression လို့ ရေးကြသလို Expression Statements လို့ ရေးကြပါသေးတယ်။ ကျဉ်းတော့အနေနဲ့ကတော့ ပိုပြီးရှင်းလုပ်းအောင် Expressions နဲ့ Statements ကို သီးခြားစီရေးလို့ ရှင်းပြရတာပါ။ JAVA Developer Guide Book မှာ ရေးထားတာကတော့ All expression are statements လို့ အဆိုပြုထားပါတယ်။ Complexy ဖြစ်မှာစိုးလို့ သီးသန့်ခြားစီရေးသားတင်ပြရတာပါ။ ကဲ operators အကြောင်းကို ဆက်လက် လေ့လာလိုက်ပါပြီး။

### Operators

Operators ဆိတာကတော့ programming language တိုင်းမှာ အမြဲလိုလိုပါဝင်ပြီး formula တွေ၊ logical ဆိုင်ရာ condition တွေရေးရာမှာ ကူညီပေးတဲ့ Helping Objects လေးတွေလို့ ပြောရင်လည်း မမှားဘူးပေါ့။ Operators တွေကို အပ်စွဲပြရင်တော့ အပ်စုံ နှစ်ပြီး အဲဒါတွေကတော့ အောက်ပါ အချို့အစားတွေပါပဲ။ Logical operator, Arithmetic (Maths) operator, Assignment operator, Relational operator ဆိုပြီးတော့ တွေ့ရမှာပါ။ Operator တစ်ခုစီအလိုက် သီးသန့်ရှင်းပြပါမယ်။

**Table 1.3 The Arithmetic Operator**

Operator	Evaluates
+	addition
-	subtraction
*	multiplication
/	division
%	remainder after division (modulus)

ဒီ Table 1.3 မှာပါဝင်တဲ့ operator တွေရဲ့ စွမ်းရည်ကတော့ maths operator ပါပဲ။ ပေါင်း၊ နှုတ်၊ မြောက်၊ စား စတာတွေကို လုပ်ပေးနိုင်ပါတယ်။ စားတဲ့သက်တာမှာ နှစ်ပိုးကွဲတဲ့အတွက် ကြောင့် (/) division ကို စားလုပ်ပြလို့မှတ်ပြီး (%) remainder after division ကို Mod လို့ ခေါ်ပါသေးတယ်။ ဥပမာ BASIC Language Arithmetic Operator အကြောင်းကို အရင် Listing တွေ ရှင်းပြရမှာပါဝင်ပြီးသားဆိုပေမယ့် ပိုပြီးနားလည်းအောင် Listing 1.9 နဲ့ Demonstrate လုပ်ပြပါမယ်။ လေ့လာကြည့်ပါ။

## ကွန်ပျိုတာသင်ခန်းစာ

**Listing 1.9** A demonstration of Arithmetic operator (OR) Maths operator

```
1. class ArithClass {  
2.     public static void main (String [ ]  
3.         args) {  
4.             int a, b, x;  
5.             a = 100, b = 50;  
6.             x = a-b;  
7.             System.out.print("Difference = "+ x);  
8.         }  
Output  
Difference = 50  
Analysis
```

၁။ လိုင်းနံပါတ် ၁ မှာ ArithClass ဆိုတဲ့ Java primary class တစ်ခုကို ဖန်တီးပြီးတော့ အဖွင့် brace { ကို ဖွင့်ထားပါတယ်။ လိုင်းနံပါတ် J မှာတော့ main () function ကို ရေးသားထားတာကိုတွေ့ရမှာပါ။ အဖွင့် brace { ကိုလည်း သတိပြုမိမှာပါ။

၂။ လိုင်းနံပါတ် ၃ မှာတော့ a, b, x ဆိုတဲ့ ကိုယ်စားလှယ်ကိန်တွေ ကို int data type သုံးပြီးတော့ Declase လုပ်ထားပါတယ်။ နောက်တစ်ကြောင်းမှာမူ a ရဲ့ တန်ဖိုးဟာ 100, b ရဲ့ တန်ဖိုးဟာ 50 လို့ သီးသန့်ကြည်ထားပါတယ်။ ဒီတော့ program တွေကျက်နေသရွှေ့တော့ ဒီတန်ဖိုးဟာ ကိန်းသေပဲပေါ့။

၃။ လိုင်းနံပါတ် ၅ မှာတော့ x ဟာ a ထက် b ကိုနှစ်တယ်ဆိုတဲ့ formula ကို ရေးသားထားပါတယ်။ ကဲ ဒီတော့ a က 100 ပါ။ b က 50 ပါ။ x = 100-50 လို့ ရေးထားတာနဲ့ အတူတူ ပါပဲ။ ဒီတော့ x = 50 ဆိုတဲ့ အဖြောက် ရှုပြီးတော့ Java Byte code အနေနဲ့ memory မှာ Load လုပ်ထားပါတယ်။

၄။ လိုင်းနံပါတ် ၆ မှာတော့ System.out.print(" Difference = "+x); ဆိုပြီး အဖြေထုတ်လိုက်တဲ့ အခါမှာ Out put အနေနဲ့ Difference = 50 ဆိုတဲ့ အဖြေကို လာပြုပါတယ်။

၅။ လိုင်းနံပါတ် ၇ မှာ main () function ရဲ့ close brace } ကို လာပိတ်ထားပါတယ်။ လိုင်းနံပါတ် ၈ မှာတော့ primary class ရဲ့ close brace } ကို လာပိတ်ထားတာကို တွေ့ရမှာပါ။ ဒီတော့ program ဟာ ပြီးသွားပါပြီ။ ဒီ Listing ဟာ Arithmetic operator အကြောင်းကို နားလည်သွားအောင် ရှင်းပြတဲ့ sample လေးပါပဲ။ သိပ်ဆက်ခဲပါဘူး။ ဒါပေမယ့် ဒါကို လေ့လာသောပေါက်မှုသာလျှင် နောက်ထပ်ပို့ချမယ့် Java programming step တွေကို သောောပေါက်မှာမို့ သေသေချာချာ လေ့လာထားစေလိုပါတယ်။

### Assignment Operator

Assignment Operator ရဲ့ အဓိပ္ပာယ်ကတော့ A ရဲ့ တန်ဖိုးဟာ 10 ဖြစ်တယ်။ B ကတော့ သူဟာ A ရဲ့ 2 ဆရှုတယ်လို့ ပြော

ရင် 20 ပေါ့။ ဘယ်လိုသိတာလဲလို့ ပြောရင် ပထမအချက် A ကို အမြစ်မှ တွက်လို့ရတာဖြစ်လို့ပါ။ ဒီလိုအချက်တစ်ချက်ကို အမြဲ ခံပြီး တွက်လို့ရတာကို programming လုပ်တဲ့ အခါမှာ assign လုပ်တယ်လို့ ခေါ်ပါတယ်။ Listing တွေနဲ့ နောက်ပိုင်းမှာ သေသေချာချာရင်းပြောပါမယ်။ အပေါ်မှာရေးပြုခဲ့တာကို sample ရေးပြုပါမယ်။

### sample

```
int a=10; //declare a and his amount is 10  
int b = a + a; //b = 20
```

ဒဲ ဒါဟာ assignment operator ကို အမြစ်ရေးပြတာပါပဲ။ အမှန်တကယ် Assignment operator အကြောင်းကို ရေးရတာကတော့ Increment operator ( ++ ), Decrement operator ( -- ) အပါအဝင် အခြား Logical ဆန်ဆန် ရေးသားရတာပါပဲ။ ဒါတွေတို့ အောက်မှာ ဥပမာအနေနဲ့ ရှင်းပြုပြီးတော့ ကျွန်း တာတွေကိုတော့ Listing တွေနဲ့ တွက်ပြပြီး Line By Line Method နဲ့ ရှင်းပြုပါမယ်။

```
a = 10; a = b; //b is 10
```

a ဟာ 10 ပါ။ a နဲ့ b ဟာ ညီတယ်။ ဒီတော့ b ရဲ့ တန်ဖိုးကလည်း 10 ပေါ့။

```
a = 6; b = a++; //b is 7
```

a ဟာ 6 ပါ။ b ဟာ a ထက် (a ကို 1 ပေါင်းထားတယ်ဆိုတော့) b ဟာ 7 ပေါ့။ ဒီနောက်မှာ a++ ; ဆိုတာ increment operator ကို သုံးထားတာပေါ့။ ရှေ့မှာ သေသေချာချာရင်းပြုခဲ့ပြီးပါပြီ။ အထူးသတိပြုရမယ့် အချက်ကတော့ assignment operator မှာ Lvalue နဲ့ Rvalue ဆိုပြီး နားလည်ရမယ့်အချက်ရှိပါတယ်။ အဓိပ္ပာယ်ကတော့ ပြောင်းလဲနိုင်တဲ့ တန်ဖိုးတွေဟာ (=) Equal sign ရဲ့ right side ဘက်မှာ ရှိရပါမယ်။ ဒါကို rvalue လို့ သိရပါမယ်။ ဒဲ အထွေထွေပြုပါမယ်။

```
x = 35; //OK
```

ဒီလိုရေးတော့ x ရဲ့ တန်ဖိုးဟာ 35 ပေါ့။ မှန်ပါတယ်။ တကယ်လို့ 10 ထည့်ပြီး ပေါင်းမယ်။ လေ့လာကြည့်ပါ။

```
x + 35 + 10; //OK! right
```

ဒါလည်း မှန်ပါသေးတယ်။ ညာဘက်ကပြောင်းလဲနိုင်တဲ့ ကိန်းတန်ဖိုးပါဝင်နိုင်လို့ rvalue လို့ ခေါ်တာပေါ့။ ကဲ ဆက်ကြည့်ပါး။

```
35 = x; //error, not an Lvalue
```

ဒီလိုရေးရင်တော့ မှားသွားပါပြီ။ 10 + 35 = x; //ဒီလိုလည်း ရေးလိုမရနိုင်ပါဘူး။ ပြောင်းလဲနိုင်တဲ့ ကိန်းတန်ဖိုးဟာ Equal Sign (=) ရဲ့ ဘယ်ဘက်ကိုရောက်လာတဲ့ အတွက် Lvalue ဖြစ်နေပါတယ်။ Lvalue ဟာ (+), (-), (\*), (/), (%) လုပ်နိုင်စွားမရှိခို့တာကို သိထားရင် နောက်ပိုင်းမှာ အမှားကို သတိပြုရောင်ရှားနိုင်မှာပါ။ ဒဲ ဒါဆိုရင် Lvalue ကို အထွေထွေမှတ်ကြရအောင်။

## ကွန်ပျော်ဘတ်ခန်းဘ

$\text{ცြောင်းလဲလို့မရတဲ့$	$(+)(-)(+)(/)(\%)$ လုပ်ရမယ့်ဘန်ဖိုး
ကိုယ်စားလှယ်ကိန်း	(OR) ကိုယ်စားလှယ်အကွဲရာ
အကွဲရာ (char) သည်	(char) သည်
Lvalue	= Rvalue
က ဒီလိုမှတ်ထားမယ်ဆိုရင်တော့ Assignment Operator	
အကြောင်းကို နားလည်သောပေါက်ပြီးတော့ ရေးတဲ့အခါမှာ	JAVA Interpreter အနေဖြင့် ကတော့ myAge ရဲ့ မူလတန်ဖိုး 5
မမှားနိုင်တော့ဘူးပေါ်နော်။ က ဆက်လက်ပြီးတော့ Arithmetic	ထဲကို 2 ပေါင်းချုပ်သာ myAge ပဲလို့ သဘောပေါက်ပြီး နောက်
(Maths) operators နဲ့ Assignment operators တို့ကို ပေါင်း	တစ်ကြောင်းမှာ myAge ရဲ့ တန်ဖိုးဟာ 7 လို့ လာပြုမှာပါ။
စပ် အသုံးပြုနည်းကို ရေးသားရင်းပြထားပါတယ်။ လေ့လာကြည့်ပါ။	က ဒီဆိုရင် ဒီထက်လွယ်တဲ့ နည်းလမ်းတို့ ရေးပြပါဉိုးမယ်။

### Combining the Assignment and Arithmetic operators

Assignment နဲ့ Arithmetic operators တွေကို ရှင်းပြီးတဲ့ အခါမှာ ဒီ operator တွေကို ပေါင်းစပ်အသုံးပြုလို့ရတဲ့ Method တွေကို ရေးပြပါ။ Operator ကို နိုင်နင်းအောင် အသုံးပြုနိုင်မှသာလျှင် programmer ကောင်းတစ်ယောက် Developer ကောင်းတစ်ယောက် ဖြစ်လာမှာပါ။ က အောက်မှာရေးထားတဲ့ sample လေးကစြိုး လေ့လာကြတာပေါ့။

#### sample

```
int myAge = 5;
int temp;
temp = myAge+2; //add 5 + 2 put it in temp
myAge = temp; 1//put it back myAge
System.out.println (myAge); //output
ဒီ sample လေးကို အကြည့်ရင် myAge ရဲ့ တန်ဖိုးက 5 ပါ။
temp = myAge + 2; လို့ ရေးတဲ့အတွက် temp = 5+2; temp
= 7 ရလာပြီပေါ့။ ဒီနေရာမှာ သိထားရမှာက temp ဆိတာ ကိုယ်
စားလှယ်ကိန်းပါ။ ကိုယ်စားလှယ်လို့ ဘာလို့ပြောသလိုဆိုတော့
temp ဟာ တွက်ချက်လို့ရထားတဲ့အဖြေကို သူ့အနေနဲ့ ယာယို
သိလောင်ပြီး သိမ်းမှတ်ပေးထားတာပါ။ နောက်တစ်ကြောင်းမှာ
တော့ myAge = temp; လို့ရေးပြီး temp ရဲ့ တန်ဖိုးကို myAge
ထဲကို ပြန်စွာပေးလိုက်ပါပြီ။ အဖြေထဲတဲ့ အခါမှာတော့ myAge
နဲ့ပဲ ထဲတဲ့ အပေါ်မှာရေးခဲ့သလို ရေးတဲ့ အခါမှာ
အနည်းငယ် complex ဖြစ်သလို သိပ်မကျမ်းကျင်ရင် အမှားတွေ
ရေးမြို့ဌ်း error တွေ တက်လာနိုင်ပါတယ်။ ဒီတော့ assignment operator ကို အသုံးပြုပြီးတော့ ဒီလိုရေးရင် ငိုဂျယ်မှာပေါ့။
```

#### sample

```
int myAge = 5;
myAge      = myAge + 2; //add 5 + 2
က ဒီလိုရေးလိုက်တော့ ပိုမလွယ်ဘူးလား။ Algebraic expression အနေဖြင့်မယ်ဆိုရင်တော့ အမိုးယူယ်မရှိပေါ်မယ်
```

JAVA Interpreter အနေဖြင့် ကတော့ myAge ရဲ့ မူလတန်ဖိုး 5 ထဲကို 2 ပေါင်းချုပ်သာ myAge ပဲလို့ သဘောပေါက်ပြီး နောက်

တစ်ကြောင်းမှာ myAge ရဲ့ တန်ဖိုးဟာ 7 လို့ လာပြုမှာပါ။

က ဒီဆိုရင် ဒီထက်လွယ်တဲ့ နည်းလမ်းတို့ ရေးပြပါဉိုးမယ်။

#### sample

```
int myAge = 5;
myAge += 2; // self-assign addition
ဒီလိုရေးပြီး အဖြေထဲတဲ့ လိုက်ခြင်းအားဖြင့် 7 ဆိတဲ့ အဖြေကို
ရုံးမှာပါပါ။ ဒီလိုရေးတာကို Self-assign လုပ်တယ်လို့ ပေါ်တယ်။
Self-assign လုပ်ရာမှာ (+=) addition အပြင် Subtraction
(-=), division (/=), multiplication (*=) နဲ့ modulus (%) =) တွေကိုလည်း ရေးလို့ရုံးပါတယ်။ Listing 1.7 အပြီး Increment နဲ့ Decrement operator အခန်းမှာ ရှင်းပြီးပေါ်မယ့်
သက်ဆိုင်ရာ operator အပိုင်းအနေနဲ့ ထပ်မံ ရှင်းပြတာပါ။ အသုံး
အနှစ်နဲ့၊ အခေါ်အခေါ်လေးတွေကအစ ဂရို့ကိုပြီး လေ့လာမယ်
ဆိုရင် အလွန်အကျိုးရှိမှာပါ။
```

### Relational Operator

Relational Operators တွေကို Table 1.4 မှာ ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။

**Table 1.4 The Relational Operator**

Name	Operator	Sample	Evaluates
Equals	$==$	100 == 50;	false
		50 == 50;	true
Not Equals	$!=$	100 != 50;	false
		50 != 50;	true
Greater Than	$>$	100 > 50;	true
		50 > 50;	false
Greater Than	$\geq$	100 >= 50;	true
or Equals		50 >= 50;	true
Less Than	$<$	100 < 50;	false
		50 < 50;	false
Less Than	$\leq$	100 <= 50;	false
or Equals		50 <= 50;	true

ဒီ operators တွေကတော့ အသေးစိတ်ရှင်းပြစ်ရာမလိုပါဘူး။ အလွယ်တက္ကသိုလ်ပြီးသားတွေပဲလို့ Listing တွေမှာပဲ သေသေ ချာချာ ရှင်းပြီးတော့မယ်။ မှတ်မိအောင်တော့ လေ့လာထားပါ။ Relational operators ရဲ့ Statement evaluates ဟာ ဟုတ်ခဲ့ရင် 1 (TRUE)၊ မဟုတ်ခဲ့ရင် 0 (FALSE) ဆိုပြီး ရွေးချယ်စရာ choice နှစ်လမ်းတဲ့မှာပဲ အလုပ်လုပ်တာပါ။ ဒီတော့ အလွယ်မှတ်ရင် 1 (TRUE) OR ) (FALSE) လို့ မှတ်ထားပါ။ ဒီတော့ Data type သေားမှာပါဝင်တဲ့ Boolean type နဲ့ ယုံ့တဲ့ပြီးလည်း အသုံးပြုနိုင်ပါတယ်။ အခုအခါမှာတော့ Boolean type ကို အသုံး

## ကွန်ပျိုတာသင်ခန်းဘ

မပြုသေးဘဲ sample အနေနဲ့ပဲ ရေးပြပါမယ်။ Relational Operators အကြောင်းကို နားလည်သောပေါက်သွားအောင် Listing 1.10နဲ့ရေးပြီး ရင်းပြထားပါတယ်။ လေ့လာကြည့်လိုက်ပါၤီး။

**Listing 1.10** A demonstration of braching based on relational operators

```
1. class RelaClass {  
2.     public static void main ( String args  
3.         [ ] ) {  
4.             int RedSox, Yankee;  
5.             RedSox = 9, Uamlee = 18'  
6.             if (RedSox > Yankee) System.out.println  
("RedSox Score!");  
7.             if (RedSox < Yankee) System.out.println  
("Yankee Score!");  
8.             if (RedSox == Yankee) System.out.println  
("Equal Score!");  
9.         }  
output  
Yankee Score!
```

Analysis

၁။ ဒီ program ကတော့ relational operators ကို ဖြင့်တတ်အောင် ရေးပြတဲ့ sample လေးပါပဲ။ ယယ်, ညီ, ကြီးဆိုတဲ့ relational operator တွေကို if statement နဲ့ တွဲဖြီးတော့ အသုံးပြထားပါတယ်။ လိုင်းနံပါတ် ၁ မှာ RelaClass ဆိုတဲ့ primary class ကို ဖန်တီးပြီးတော့ open brace { ကို စတင်ထားတာကို တွေ့ရမှာပါ။

၂။ လိုင်းနံပါတ် ၂ မှာတော့ main () function ကို ရေးသားထားပါတယ်။ main () function ရဲ့ open brace { ကိုလည်း သတိပြုမှာပါ။ လိုင်းနံပါတ် ၃ မှာတော့ int data type ကို အသုံးပြုပြီးတော့ RedSox နဲ့ Yankee ဆိုတာကို decalare လုပ်ပါတယ်။ လိုင်းနံပါတ် ၄ မှာတော့ RedSox ရဲ့ တန်ဖိုးသည် ၉ဖြစ်ပြီး Yankee ရဲ့ တန်ဖိုးသည် ၁၈ပါလို့ ကြေညာပါတယ်။ ဒီ program ကို Run လုပ်နေသရွှေ့တော့ ဒီတန်ဖိုးတွေဟာ ကိန်းသေပါပဲ။ int data type ကို အသုံးပြပြီးတော့ Declare လုပ်စဉ်ထဲက တန်ဖိုးတွေကို ထည့်သွင်းကြညာလို့ ရပေမယ့် အခုမှ အခြေခံကစာစြီး လေ့လာသူတွေ မျက်စီရင်းအောင်လို့ လိုင်းခွဲပြီးတော့ ရေးပြရတာပါ။

၃။ လိုင်းနံပါတ် ၅ မှာတော့ if statement ကို သုံးပြီး Relational operator နဲ့ တွဲဖက်ပြီးတော့ RedSox ဟာ Yankee ထက် ကြီးသလား? လို့ မေးပါတယ်။ RedSox ရဲ့ တန်ဖိုးက ၉ဖြစ်ပြီး Yankee ရဲ့ တန်ဖိုးက ၁၈ပါ။ ဒီတော့ မကြီးဘူးဆိုတော့ if (expression) နောက်ကပါတဲ့ state-

ment ကို အလုပ်ဆက်လုပ်စရာမလိုပါဘူး။ Escapeဖြစ်ပြီး တော့ နောက်တစ်ကြောင်းကို ဆင်းသွားမှာပါ။

၄။ လိုင်းနံပါတ် ၆ မှာတော့ RedSox ဟာ Yankee ထက် ကျယ်သလားလို့ လာပြီး testလုပ်ပါတယ်။ ဒီအခါမှာ RedSox ဟာ Yankee ထက် ကျယ်နေပြီးဆိုတော့ မှန်သွားပါပြီ။ အဲဒီတော့ expression နောက်က statement ကို အလုပ်ဆက်လုပ်တဲ့ အခါမှာ Yankee Score! ဆိုတဲ့အဖြေကို Output အနေနဲ့ လာပြပါတယ်။

၅။ လိုင်းနံပါတ် ၇ မှာတော့ RedSox နဲ့ Yankee ဟာ တူညီသလားဆိုတဲ့ expression ကို လာပြီး test လုပ်ပါတယ်။ အဖြေ မှန်ကိုရပြီးပေမယ့် break; Control Statement ကို အသုံးပြထားခြင်းမရှိတဲ့အတွက် ဆက်လက်ပြီးတော့ test လုပ်တာပါ။ ဒါလေမယ့် မညြိတ်အတွက် သူရဲ့နောက်က Statement ကို အလုပ်ဆက်လုပ်ဘဲ escapeဖြစ်ပြီးတော့ လိုင်းနံပါတ် ၈ မှာ ရေးထားတဲ့ min () function ရဲ့ close brace } ဆိုတဲ့ ဆင်းသွားပါပြီ။

၆။ လိုင်းနံပါတ် ၉ မှာတော့ primary class ရဲ့ close brace } ကို လာပိတ်ထားပြီးတော့ program ဟာ ပြီးသွားပါပြီ။ က ဒါဆိုရင် output ထွက်တဲ့အထိကို သေသေချာချာ လေ့လာချွေကြပြီး လေ့လာပါ။ Run လုပ်ပြီးတော့လည်း output ကို test လုပ်ကြည့်ပါ။

အမှန်တကယ်ကတော့ ဒီ Relational operators အကြောင်းကို Listing 1.4မှာ ထည့်သွင်းရေးသားပြီးပါပြီ။ ဒါလေမယ့် သက်ဆိုင်ရာအခန်းကဏ္ဍအလိုက် ဖြစ်စေချင်တဲ့အတွက် ထင်စုံပြီး sample အနေနဲ့ ရေးသားရင်းပြရတာပါ။ Sample တွေဟာ အလွယ်တွေပဲဖြစ်နေတဲ့အတွက် လေ့လာသင်ယူရာမှာ မပေါ့ဆို စေချင်ပါဘူး။ လေ့လာရာမှာ ဂိုတယ်ဆိုတာ မရှိပါဘူး။ နောက်ပိုင်း chapter 2 ကလို့တော့ အဆင့်ဖြင့်ပုစ္စတွေချည်းပဲ တွက်ချက်ရတော့မှာပါ။ တရာ့နေရာတွေမှာဆိုရင် နိုင်ငံတကာအဆင့် Exam; တွေက JAVA Question အဟောင်းတွေကို အသုံးပြပြီးတော့ သင်ကြားပို့ချေပေးသွားမှာပါ။ ဒီ chapter 1 မှာ JAVA Language အကြောင်းကို စတင် လေ့လာသင်ယူတာဖြစ်လို့ အလွယ်ဆုံး အရှင်းဆုံးဖြစ်အောင် sample တွေ Listing အလွယ်လေးတွေနဲ့ပဲ ရင်းပြနေတာပါ။ အလွယ်ကူဆုံးပြောရရင်တော့

က၊ ခ မတတ်ဘဲနဲ့ မြန်မာစာရေးလိုအပ်တဲ့ မရနိုင်တော့ပါဘူး။  
A, B, C, Dမတတ်ဘဲနဲ့ အင်ပိုင်စာ ရေးလိုအပ်တဲ့ မရနိုင်ပါဘူး။  
+, -, \*, / ကို နားမလည်ရင်လည်း သချုပ်တွက်လို့ မရနိုင်ပါဘူး။

အခြေခံ JAVA Language အကြောင်းကို ပိုင်နိုင်မှသာလျှင် Theory သဘောတရားကို နားလည်သောပေါက်မှသာလျှင် programmer အဆင့် Developer အဆင့် Software Engi-

## ကွန်ပူတာသင်ခန်းစာ

neerအဆင့်တွေကို တက်လှမ်းနိုင်မှာဖြစ်ပါတယ်။ Sommerville ရေးသားတဲ့ Software Engineering စာအုပ်ဟာ Lancaster University က ထုတ်ဝေတာပါ။ အဲဒီစာအုပ်မှာဆိုရင် C++နဲ့ Ada programming language ကို အခြေခံပြီးမှ ရှင်းလင်းသွားတာကို တွေ့ရရင် ကျွန်ုင်တော် ဘာတို့ဆိုလိုသလဲဆိုတာကို သတိပြနားလည်နိုင်မှာပါ။ ဒါကြောင့် အခြေခံပိုင်နိုင်အောင် နည်းမျိုးစုံနဲ့ ရှင်းပြန်တာပါ။ Software Engineering သင်ရှိုးကို ယခုလက်ရှိ Computer Universiy & College တွေမှာ ပြဋ္ဌာန်းထားပါတယ်။

### Logical Operator

Logical operators တွေကတော့ အောက်မှာ Table 1.5 အနေနဲ့ ရေးပြထားပါတယ်။ Logical operators ဟာ သာမန် operators (အဓိုး Arithmetic, Assign, Relational စုံ operators) တွေထက် ပိုနက်နဲ့ပြီးတော့ တွေးပေါ်မြှုပ်မြင်သုံးသပ် မှတွေ ပါဝင်ပါတယ်။ Operators တစ်ခုချင်းစိတိ သီးသန် ရှင်းပြပါမယ်။

**Table 1.5 The Logical Operators**

Operators	Symbol	Example
AND	&&	expression 1 && expression 2
OR		expression 1    expression 2
NOT	!	expression

ကဲ ဒီယေားမှာပါဝင်တဲ့ Logical operators ရဲ့ symbol လေးတော်ကို မှတ်မိအောင်ကြည့်ပြီးရင်တော့ အောက်မှာ Logical operators တွေကို တစ်ခုချင်းစိုင်းပြတာကို လေ့လာကြည့်ပါ။ အလွန်အရေးကြီးတဲ့အတွက် သေသေချာချာလေ့လာပါ။

### Logical AND

Logical AND operator ရဲ့ စွမ်းဆောင်မှုကတော့ အဆိုပြုချက်နှစ်ခုလုံး မှန်ကန်မှာသာလျှင် အဲဒီအဆိုပြုချက်ကို လက်ခံမှာပါ။ အကယ်၍ တစ်ခုချက်ချက်ကသာ မှားနေရင်တော့ အဲဒီအဆိုပြုချက် တွေဟာ လက်ခံခြင်း မရှိနိုင်တော့ပါဘူး။ ဥပမာအနေနဲ့ ရှင်းပြရင်တော့

expression 1 — expression 2 — statement  
မင်းထမင်းတကယ် AND(ပြီးတော့) ငွေရေး ဒါမှာသာ ထမင်းဆာန်သလား? ပါရဲ့လား? စားလို့ရမှာပေါ့။  
— operator —

ဒီအဆိုတိ လေ့လာရင် expression 1 နဲ့ expression 2 ဆိုတဲ့ အချက်နှစ်ချက်ကို AND operator ခံပြီး ရေးထားပါတယ်။ Expression 1 က ထမင်းတကယ်ဆာန်သလား? ဆိုတော့ မဆာ ရင်လည်း စားလို့ရပါမဲ့လား? မရပါဘူး။ ဆာမှ စားချင်မှာပေါ့နော်။

က ဒီတော့ အဆိုပြုချက် expression 1 ကတော့ မှန်ပါပြီ။ ဆာနေတယ်ပေါ့။ Expression 2 ကတော့ မင်းငွေရေးပါရဲ့လား? ဆိုပြီး အဆိုပြုထားပါတယ်။ ထမင်းဆာပေမယ့် ငွေသာမပါရင်တော့ စားလို့ရပါမဲ့လား? မရနိုင်ပါဘူး။ ဒီတော့ ငွေပါဝါတယ်ဆိုတဲ့ အဆိုပြုချက်ကို လက်ခံလိုက်ပါပြီ။ က ဒါဆိုရင်တော့

ထမင်းလည်းဆာတယ်။ AND (ပြီးတော့) ငွေလည်းပါတယ်။ ဒီတော့ ထမင်းစားလို့ရပြီပေါ့နော်။ ဒီအတွေးအခေါ်နည်း Method လေးကို လေ့လာကြည့်မယ်ဆိုရင် AND (&&) operator အကြောင်းကို အခြေခံတော့ နားလည်သွားမှာပါ။ Expression 1 နဲ့ expression 2 ဆိုတဲ့ အဆိုပြုချက်နှစ်ချက်စလုံး မှန်မှာသာလျှင် နောက်က အလုပ်ကို ဆက်လုပ်မှာပါ။ Program code အနေနဲ့ရေးရင်တော့ AND လို့ ရေးလို့မရပါဘူး။ (&&) ဆိုတဲ့ Symbol လေးကိုပဲ အဆိုပြုရမှာပါ။ အောက်မှာ sample အနေနဲ့ ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။ Logical poerator AND ကို if statement နဲ့တွဲပြီး သုံးပြထားပါတယ်။

**sample**

```
if (x==5) && (y==5) System.out.print ("Equal!");
```

အမို့ပွာယ်ကတော့ x ကလည်း 5 နဲ့ညီပြီး y ဟာလည်း 5 နဲ့ညီမှာသာလျှင် Equal! ဆိုတဲ့အဖြေကို ထုတ်ပေးနိုင်မှာပါ။ အကယ်၍ x သည် 5 နဲ့မည်းရှုင်ဖြစ်ဖြစ်စေ သို့တည်းမဟုတ် y သည် 5 နဲ့မည်းရှုင်ဖြစ်ဖြစ်စေ တစ်ခုချက်သာ 5 နဲ့မည်းရှုင်တော့ Equal ဆိုတဲ့ output ကို ထုတ်ပေးနိုင်ပါမဲ့လား? မထုတ်ပေးနိုင်ပါဘူး။ ဒါဟာ AND (&&) operator ကို useလုပ်တဲ့နည်းပါပဲ။ သေသေချာချာသာ လေ့လာကြည့်မယ်ဆိုရင် အလွန်အသုံးဝင်တဲ့ operator ဖြစ်ကြောင်းကို တွေ့ရမှာပါ။ နောက် chapter တွေရဲ့ Listing တွေမှာ ထည့်သွင်းအသုံးပြုပြီး ရှင်းပြပါမယ်။ ဆက်လက်ပြီး Logical Or (!) operator ကို လေ့လာကြည့်ကြပါဦးစီးနှင့်။

### Logical OR

Logical OR operator ရဲ့ စွမ်းဆောင်နှင့်မှုကတော့ တစ်ခု မဟုတ် တစ်ခုဆိုတဲ့ စွမ်းဆောင်ရည်ပြဖြစ်ပါတယ်။ အဆိုပြုချက် expression နှစ်ခုထဲက တစ်ခုမှန်ရှင်ဖြစ်ဖြစ်စေ နှစ်ခုလုံးမှန်သည်ဖြစ်စေ သူရဲ့နောက်က အလုပ်ကို ဆက်လုပ်ခွဲပြုမှာပါ။ ဥပမာ အနေနဲ့ရေးပြပါမယ်။

expression 1 — operator — expression 2 — statement —  
မင်းမှာ OR အကြွေးဝယ်ကတ် ဒါမှာသာလျှင် ငွေပါရဲ့လား? (ဒါမှာမဟုတ်) ပါရဲ့လား? လိုချင်တာကို ဝယ်လို့ရမှာပေါ့။

က ဒီဥပမာလေးကို လေ့လာမယ်ဆိုရင်တော့ ငွေပါရဲ့လား? ဆိုတဲ့ expression 1 မှာ ငွေပါလာခဲ့ရင် လိုချင်တာကိုဝယ်လို့ရပြီပေါ့နော်။ OR (ဒါမှာမဟုတ်) ငွေမပါဘဲ အကြွေးဝယ်ခွင့်ကတ်ပြား

## ကွန်ပူးတာသင်နှင့်စာ

ပါခဲ့ရင်? ဆိုတဲ့ expression 2 မှန်ခဲ့ရင်လည်း လိုချင်တာကို ဝယ်ခွင့်ရမှာပေါ့။ အဲဒီ expression နှစ်ခုထက် expression တစ်ခုခဲ့ဖြစ်စေ၊ expression နှစ်ခုလုံးဖြစ်စေ မှန်ခဲ့ရင် နောက်ကအလုပ်ဖြစ်တဲ့ လိုချင်တာကို ဝယ်လို့ရမှာပေါ့ဆိုတဲ့ statement ကို အလုပ်ဆက်လုပ်မှုပါ။ OR operator ကို အလွယ်တက္ကမှတ်ရင်တော့ တစ်ခုမဟုတ် တစ်ခုလို့ အလွယ်မှတ်ပါ။ နှစ်ခုစလုံးမှန်ခဲ့ရင်လည်း statement က အလုပ်ဆက်လုပ်မှုပါ။ အလွယ်မှတ်ရအောင်ရေးပြုပါမယ်။

တစ်ခုမဟုတ်တစ်ခုဖြစ်ခဲ့လျင်ဖြစ်စေ

Statement ကို  
အလုပ်လာလုပ်

နှစ်ခုစလုံးဖြစ်ခဲ့လျင်ဖြစ်စေ

Programming လုပ်ရမှာတော့ OR လို့ ရေးလို့မရပါဘူး။ (!!) symbol လေးနဲ့ ရေးရပါတယ်။ Sample အနေနဲ့ code Line လေး ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။

### sample

```
if (x==5) 11 (y==5) System.out.print ("OK");  
ဒါ ဒီ program code ရေးထားပုံကတော့ x သည် 5 နဲ့ညီတယ်။  
OR (ဒါမှမဟုတ်) y ဟာ 5 နဲ့ညီမယ်။ ဒါတော့ နှစ်ခုစလုံးဖြစ်စေ၊  
တစ်ခုခဲ့ဖြစ်စေ 5 နဲ့ညီခဲ့ရင် OK: ဆိုတဲ့အဖြောက် အဖြောက်  
ထုတ်ပေးမှာပါ။ က ဒါ ဒီ sample လေးကိုကြည့်ပြီး သေသေချာချာ  
လေ့လာမယ်ဆိုရင် Logical OR (။) အကြောင်းကို နားလည်  
သဘောပေါက်သွားမှုပါ။ Logical AND (&&) နဲ့ Logical  
OR (။) အကြောင်းကို နားလည်သွားပြီဆိုရင်တော့ Logical NOT  
(!) အကြောင်းကို ဆက်လက်လေ့လာပါ၏။
```

### Logical NOT

Logical NOT ကို အသုံးပြုတဲ့ method ကတော့ မဖြစ်ခဲ့လျင် (မဖြစ်ခဲ့မှသာလျင်) သူရဲ့နောက် statement ကို အလုပ်ဆက်လုပ်မှုပါ။ က ဒါ ဒီ operator အကြောင်းကိုတော့ ဥပမာအနေနဲ့ မရေးပြတော့ဘဲ sample coede အနေနဲ့ပဲ ရေးပြီး ရှင်းပြု တော့မယ်။ အမို့ပုံယ်နက်နတဲ့အတွက် သေသေချာချာလေ့လာပါ။

### sample

```
if (! (x==5)) System.out.print ("Hello JAVA");  
ဒါ ဒီ code ရဲ့ အမို့ပုံယ်ကတော့ x ဟာ 5 နဲ့မညီမှသာလျင် Hello JAVA! ဆိုတဲ့ output statement ကို ထုတ်ပေးမှာပါ။ မညီမှ ထုတ်ပေးမှာနော်။ တကယ်လို့ x ရဲ့တန်ဖိုးဟာ 5 နဲ့ညီခဲ့ရင်တော့ Hello JAVA! ဆိုတဲ့ output statement ကို အလုပ်လာပြီး လုပ်ပါ၌းမလား? မလုပ်တော့ဘဲ escape ဖြစ်ပြီးနောက် code Line တစ်ကြောင်းကို ဆင်းသွားမှုပါ။ ဒါဟာ NOT (!) operator ကို အသုံးပြုတဲ့ method ပါပဲ။ က ဒါကို အလွယ်မှတ်ကြရအောင်။
```

မဟုတ်ခဲ့ရင် —————> statement ကို အလုပ်လာလုပ်

ဂိုဏ်းနားလည်သွားအောင်လို့ အပေါ်မှာရေးထားတဲ့ sample code ကိုလည်း နောက်တစ်နည်း ရေးပြုပါမယ်။ ဂိုဏ်း compact ဖြစ်ပြီးတော့ အများဆုံး useလုပ်တဲ့နည်းလမ်းပါပဲ။ အကြောက်တဲ့ရေးနည်းကို အသုံးပြုနိုင်ပါတယ်။

### sample

```
if (x!=5) system.out.print ("Hello JAVA");  
ဒါ ဒီဒို့ရေးရင်လည်း အပေါ်က sample code ထက် ဂိုဏ်းတော့ လွယ်ကြပြီးတော့ compact ဖြစ်တာပေါ့။ ဒီ Logical operator တွေဟာ အခြား Assignment operator, Arithmetic operator (Maths), Relational operators တွေထက် ပို့ပြီးနက်နပါတယ်။ Programming လုပ်ရာမှာ program flow ကို ထိန်းချုပ်ရာမှာဖြစ်စေ Limited data system ကို အသုံးပြုတဲ့ အပေါ်ဖြစ်စေ အသုံးပြုရတဲ့ operators တွေပဲဖြစ်လို့ example ပျို့စုံပေးပြီး နည်းလမ်းအမျိုးမျိုးနဲ့ ရှင်းပြနေတာပါ။ ထပ်ခါ တလဲလဲ ဖော်ရှုပြီးတော့ တွေးခေါ်ကြည့်ပါ။ ဒါမှသာလျင် ထူးချွဲနိုင်တော့ အဆင့်မြင်တဲ့ JAVAပညာရှင်တစ်ယောက်ဖြစ်လာမှာပါ။
```

ယနေ့ ဖြန်မာနိုင်ငံမှာသာမက နိုင်ငံတော်တော်များမှာ စိတ်ဝင်တစား ဖြေဆိုနေကြတဲ့ ဂုပ်နိုင်မှစစ်ဆေးတဲ့ JITEC Fundamental Engineering Examination အပါအဝင် နိုင်ငံတကာအဆင့်မြင်တဲ့ Examination တွေမှာပါ ဒါ operators တွေ အကြောင်းကို objective type (All questions are compulsory) ပုံစံနဲ့ မေးသလို select one of the four questions ဆိုပြီး JAVA Language, C Language, Fortran Language, Assembler Language စောတွေနဲ့ program decription တွေပေးပြီးမှ sub-questions တွေကိုဖြစ်စေ၊ operators တွေနဲ့ expression တွေ၊ statements တွေ၊ control structure တွေကို ပေါင်းစပ်ပြီး ရေးဖွဲ့တဲ့နေရာတွေမှာ ပါ Blank ချုပ်ပြီး ဖြည့်စွက်ရမယ့် Answers တွေကိုတော့ အောက်မှာ choice လုပ်စေပါတယ်။ ဒါဆို သေသေချာချာ လေ့လာကြည့်မယ်ဆိုရင် Exam Type တွေ ဖြေဆိုအောင်မြင်နိုင်ဖို့အတွက် operators အပိုင်းကို နားမလည်ရင် ဖြေဆိုနိုင်မှာမဟုတ်ဘူးဆိုတာ ကို သိရှိနားလည်ထားစေချင်ပါတယ်။ ဥပမာအနေနဲ့ JITEC Examination က JAVAမေးခွန်းဟောင်းတစ်ပုဒ်ကို ရေးပြထားပါတယ်။ လေ့လာပြီးရင် operators တွေကို အသုံးပေါင်းများ ကျွန်ုင်တော်များတွေကို နားလည်သဘောပေါက်သွားမှုပါ။ JAVA Question program type ကိုပဲ ထည့်ပေးထားတာပါ။ Sub-questions တွေကိုတော့ ထည့်မပေးထားပါဘူး။ Sample အနေနဲ့ ပဲဖြစ်လို့ပါ။

## ကွန်ပျိုတာသင်ခန်းစာ

**Q13**

Consider the following JAVA program then answer

**Sub-questions 1-4**

(Program)

line no.

```
1. public class Swapper {  
2. byte myData [] [] ;  
3. public static void main (String args  
[]) {  
4.     Swapper obj = new Swapper ();  
5.     obj.init ();  
6.     obj.sorting ();  
7.     obj.printData ();  
8. }  
9. public void init () {  
10.    myData = new byte [5] [] ;  
11.    for (int i = 0; i < myData.length;  
           i++) {  
12.        myData [i] = new byte [5-i] ;  
13.    }  
14.    for (int i = 0; i<myData.length;  
           i++)  
15.        for (int j=0; j<myData [i].  
                  length; j++) {  
            A  
16.        }  
17.    }  
18. public void sorting () {  
19.    int length = 0;  
20.    for (int i = 0; i<myData.length;  
           i++) {  
21.        length +=myData [i].length;  
22.    }  
23.    boolean isSwap;"  
24.    do {  
25.        isSwap = false;      B  
26.        for (int i=0; i<myData.  
                  length; i++) {  
27.            int j;  
28.            for (j = 0; j<myData  
                  [i].length -  
                  1;j++) {  
29.                if (myData [i]  
                    [j] < myData [i]  
                    [j+1]) {  
30.                    isSwap = true;  
31.                    Swap (i,j,i,j+1);  
32.                }  
33.            }  
34.            if (myData [i] [j] < myData  
                  [i+j] [0]) {  
35.                isSwap = true;  
36.                Swap (i,j,i+1,0);  
        }  
    }  
}
```

```
37.            }  
38.        }  
39.    } while (isSwap);  
40. }  
41. public void swap (int i1, int i1, int  
                     i2, int j2) {  
42.     byte temp = myData [i1] [j2];  
43.     myData [i1] [j2]=myData [i2] [j2];  
44.     myData [i2] [j2]=temp;  
45. }  
46. public void printData () {  
47.     for (int i=0;i<myData [i].length;  
           j++) {  
48.         for (int j=0; j<myData [i].  
                   length; j++) {  
49.             System.out.print (myData  
                   [i] [j]+ " ");  
50.         }  
51.         System.out.println ();  
52.     }  
53. }  
54. }
```

**Sub-question 1**

Select the expression that can be inserert into block A to generate an integer of type byte which has value between 11 to 65 and the program can be compiled correctly?

- (a) myData [i] [j] = (byte) 11+(byte) (Math. random()\*55);
- (b) myData [i] [j] = 11+(byte) (Math.random () \*55);
- (c) myData [i] [j] = (byte) 11+(byte) Math. random() \*55
- (d) myData [i] [j] = (byte) (11+(Math. random()\*55);

**Sub-question 2**

While of the following answers is the correct result of executing the program?

a)	b)	c)
28 28 29 34 38	60 52 48 44 43	30 34 33 38 62
44 47 51 52	42 39 37 30	18 28 42 43
53 56 60	29 23 23	34 39 62
62 63	22 17	33 45
65	12	46

- d) The program yields Runtime Error because of ArrayIndexOutOfBoundsException

ဒီ sample question ကို လေ့လာကြည့်ရင် operators တွေ၏ အသုံးဝင်ပုံနဲ့ program flow ကို ထိန်းချုပ်ပုံကို နားလည် သဘောပါက်သွားမှာပါ။ Operators တွေ၏ အသုံးဝင်ပုံဟာ ဒီလောက်ပဲလားဆိုရင် မှားသွားပါလိမ့်မယ်။ ယနေ့ DataBase Technologies အပိုင်းမှာ အသုံးပြုနေတဲ့ DataBase Lan-

ကွန်ပျူတာသင်ခန်းတ

guage (SQL) မှာဆိုရင် ဒီ operators တွေကို အသုံးပြု အခြေခံ  
ပြီးမှ DataBase Command Control System (DBCCS)  
အနေဖြင့် အသုံးပြုရတာပါ၊ ဒီအချက်တွေကို သိရှိရားလိုအပ်ခြိုက်ရင်တော့  
operators ဆိုတာဟာ programming language တွေမှာ  
ဘယ်လောက်အရေးပါတယ်ဆိုတာကို သိရှိနေးလည်နိုင်မှာပါ။

အရေးအကြီးဆုံးအချက်ကတော့ ဘယ် programming language ကိုပဲ လေ့လာလေ့လာ Program Flow ကို သိအောင်လို့ အခေါက်ခေါက်အခါအခါ လက်နဲ့ချွောက်ပြီး Analysis တွေကို သေသေချာချာဖတ်ကြည့်ခြင်းဟာ code တွေကို interpreter မှာ run လုပ်ပြီးမှ လေ့လာတာထက် ပိုပြီးတတ်မြောက်လွယ်တယ် ဆိုတာတို့ အဲနဲ့အတွေ့အတွေ့ အကြောင်းအမြန်မျှ တင်ပြပိုပါတယ်။ ဒါကြောင့် Listing တစ်ပါတ်ချင်းကို လက်နဲ့ချွောက်ပါ။ ပြီးမှ run လုပ်ပြီးတော့ လေ့လာပါ။ က ဆက်လက်ပြီးတော့ Relational operators နဲ့ Logical Operators နှစ်ခုကို ယူဉ်တဲ့အသုံးပြုတဲ့ Relational operators အပိုင်းကို လေ့လာကြည်ပါဘူးနှိုး။

## Relational Precedence

Relational Precedence ඩීතාගලටු Relational operators & Logical operators අනුමත යුතු තේරිස් වෙත පි

တာပဲဖြစ်ပါတယ်။ Complexy ဖြစ်တယ်လို့ ထင်ရပေမယ့် အလွန် အသုံးဝင်တဲ့ နည်းပညာရပ်ဖြစ်တာကြောင့် သေသေချာချာ ရှင်းပြတားပါတယ်။ Sample အနေနဲ့ ရှေ့ပြထားပါတယ်။ လေးလာ ကြည်ပါ။

### sample

ဒါ code လေးကိုလေ့လာပြီး အဖြေထွက်လိုက်ရင် အောက်မှာ  
[exp1] [exp2] [exp3] [statement]  
if(x>5 && y>5) || z>5) System.out.print("OK!");

ရေးပြထားတဲ့အဖြူတွေကို ရမှာပါ။

x යන් 5 යන්**ලිංගය** (AND ලිංගය) y ගලයු: 5 යන්**ලිංගය** (OR සියලුමයි) z යන් 5 යන්**ලිංගය** සියලුම OK! සියලුම Output ගිය ලාපුවාපි || තම්බන්ස්: ප්‍රාග්‍රැන්දෙනු Logical පෙනීමක් x න් y යන් 5 යන්**ලිංගය** අවශ්‍ය න් OR z යන් 5 යන්**ලිංගය** අවශ්‍ය න් යන්**වැනිවිට්ත්** න් අනුමත න් Statement ගිය අලුත්තුවා දුරකථනයේ output ගිය තුළ පෙන්වාපි ||

နောက်တစ်နည်းကတော့ x က 3ဖြစ်နေပြီး y ကတော့ 10 ဖြစ်နေပါတယ်။ z ကတော့ 10 ပါ။ ဒါလည်း Statement ကို အလုပ်ဆင်လုပ်ပြီး output ကို ထုတ်ပေးပါတယ်။ ဘာကြောင့် လဲဆိုတော့ x ကလည်း 5 ထက်ကြီးမှရမှာပါ။ y က 10 ဖြစ်တိုင်း မရပါဘူး။ Expression 1 နဲ့ Expression 2 ကို AND နဲ့ ခံရေး ထားလိုပါ။ ဒါပေမယ့် ဒီဘက်က OR (!) ခံရေးထားတာက z သည် 5 ထက်ကြီးလျင်ဆိုတော့ x နဲ့ y ဟာ တစ်တွဲပေါ့။ သူတို့ မှားသားပေမယ့် z ကတော့ 10 ဆိုတော့ နောက်က statement ထိ အလုပ်ဆင်လုပ်လို သေားတာပါ။ ပံ့ပိုပြီး ရေးပါပါမယ့်။

ဒီအခါမှာ ရှုံးက expression 1 က မှားသွားတော့ ex-  
 x က 3 ပြီးတော့ y က 10ဖြစ်ရမယ်၊ ဒါမှမဟုတ် z က 10  
 if ( $x > 5$        $\&\&$        $y > 5$       ||       $z > 5$ )  
 \exp 1 \exp 2 \exp 3 \exp  
 x နဲ့ y နှစ်စုစလုံးထက် OR z က 5ထက်ကြီးရင်ဖြစ်ဖြစ်  
 ကြီးရင်ကြီးရမယ်

expression 2 ကလည်း အကျိုးမဝင်တော့ပါဘူး။ ဒါပေမယ့် expression 3 က မှန်နေတော့ နောက်က statement ကို အလုပ် ဆက်လည်ပါတယ်။ က ဒါဆိုရင်တော့ နောက်ပုံစံတစ်မျိုး ရေးပြပါ မယ်။ ဒါပေမယ့် ဒီလိုရေးတာဟာ အမှားကြီးဆိုတာကိုတော့ သတိပြပါ။ Sample အနေနဲ့ ရေးပြတာပါ။

**sample:** if((x>5)&&(y>5) || z>5))System.out.  
              print ("OK!");//error!test

အရင်ရေးပြခဲ့တဲ့ sample code ရေးပြတာ နဲ့ တူသလိုရှိပေ  
မယ့် အမှန်တကယ်တော့ မတူပါဘူး။ သူ့ရဲ့အမိဘယ်ကို အောက်မှာ  
ရေးပြထားပါတယ်။ သေချာလေ့လာကြည့်ပါ။

ကွန်ပျူတာသင်ခန်းစာ

x ဟာ 5 ထက်ကြီးရမယ်။ AND(ပြီးတော့) y OR(သို့မဟုတ်) z သည် 5 ထက်ကြီးမသာလျှင် အနောက်က statement ကို အလုပ်လုပ်၍ outputဖြစ်တဲ့ OK! ဆိုတဲ့အဖြေကို လာပြေးမှာပါ။

အရင်ရေးပြုတဲ့ sample code နဲ့ တူသလိုရှိပေမယ့ y နဲ့ z ကို OR (!! ) သက်တဲ့ အခါမှာ အဖွဲ့အပိတ် bracket ( ) ထွေ ခံရေးလိုက်တဲ့ အတွက် အခုလို ထွေးခေါ်ဖြေရှင်းမှုပြောင်း သွားတာပါ။

ဒါမေတ္တာင့် program တွေကို Line By Line Method နဲ့ ရှင်းပြတဲ့အခါမှာ အဖွင့်/ အပိတ် စတဲ့ brace တွေကအစ အတိ အကျ ရှင်းပြတာကို ထံတိပြုမိမှာပါ။ ကဲ ဆက်လက်ပြီးတော့ Logical test တွေအနေနဲ့ operator တွေကို use လုပ်တာကို လေ့လာကြည့်ပါဦး။

## More About test the Logical Statements

အပေါ်မှာ ရေးပြခဲ့တဲ့ sample code တွေကို နားလည် သငောပါက်ပြီဆိုရင်တော့ ပိုပြီး လွယ်ကူတဲ့ clear cut style ကို ရေးပြပါမယ်။ JAVA Language မှာ zero ဆုံးရင် False လို သိထားရပါမယ်။ က ဒါကိုမှတ်ထားပြီးတော့ အောက်မှာရေးထားတဲ့ sample code ကို လေ့လာကြည့်ပါ။

### *sample*

```

if (x)      // if x is true (nonzero)
    x = 0;    // set it to 0 (zero)

```

အမိပ္ပါယ်ကတေသာ `if(x)` ဆိုတာက `x` သည် 0 (zero) မဟုတ်ဘာ  
ကိန်းတစ်ခုဖြစ်ခဲ့လျှင် အောက်တစ်ကြောင်းကလုပ်စီးတေသာ `pro-`  
`gram` ကို `calculs` လုပ်နေသရ၍ 0 (zero) အနေနဲ့ `set` လုပ်စီး  
တေသာ တွက်ချက်ရမှာပါ။ ဒါ ကောင်းမူပါ။

**sample** :if (x!= 0) // if is nonzero

`x = 0;` // set it to 0(zero)  
 ଅପ୍ରୋଗରମେ ପ୍ରକାଶ କରିବାକୁ ପାଇଁ ଏହା କାମ କରିବାକୁ ଆବଶ୍ୟକ ନାହିଁ ॥  
`x == 0` (zero) ମହାତମ୍ଯରେ ପାଇଁ ଏହା କାମ କରିବାକୁ ଆବଶ୍ୟକ ନାହିଁ ॥  
 NOT operator କିମ୍ବା ଅବ୍ୟାସିତ ପରିମାଣ କରିବାକୁ ଆବଶ୍ୟକ ନାହିଁ ॥  
 କିମ୍ବା `x != 0` (zero) ମହାତମ୍ଯରେ ପାଇଁ ଏହା କାମ କରିବାକୁ ଆବଶ୍ୟକ ନାହିଁ ॥

တော့မှာပါ။ ကဲ အဲဒီအပေါ်မှာ ရေးပြခဲ့တာတွေနဲ့ same mean

ဖြစ်တဲ့ sample code ကို ရေးပြပါမယ်။ လေ့လာကြည့်ပါဘီး။  
**sample** : if (!x) // if x is false(zero)  
                if (x = 0) // if x is zero  
ဒီလိုရေးလိုက်ခြင်းအားဖြင့် အပေါက ရေးပြခဲ့တဲ့ sample တွေနဲ့  
အမြဲဗုံယူတော်မယ့် အခြေနောက်ဆုံးရေးပြတဲ့ sample က ပိုမြဲ  
ကျေ easier to understand ဖြစ်တယ်။ ဒါ ဒါမိုင်ကျေ

operators පෙනෙයාදී දියුණුව වෙත තොරතුරු කිරීමේදී  
JAVA Language තුළ ඇත්තු ඇත්තු නොවා ඇත්තු මෙය දියුණුව  
පිතයි වායු වෙත පිළිගියා ඇත්තුවේදී ප්‍රාග්ධනය දීම් ප්‍රාග්ධනය  
වෙත පිළිගියා ඇත්තුවේදී ප්‍රාග්ධනය දීම් ප්‍රාග්ධනය

## Case Sensitivity

ဒီနေရာမှ JAVA Language အကြောင်းကို အတော်အသင့် သဘောပါက်သွားပြီဆိတ်တော့ အထူးသတိပြုရမယ့်အချက်ကတော့ JAVA ဟာ C & C++ ထိလိုပဲ Case Sensitivity ဖြစ်တယ်ဆိတဲ့ အချက်ပါပဲ။ ဥပမာအနေနဲ့ ရှင်းပြရရင်တော့ age လို့ ကြညာထားတဲ့အမည်ကို Age လို့ ခေါ်သုံးလို့မရသလို AGE လို့ ခေါ်ယူ အသုံးပြုလိုလည်း မရဂါဘူး။ Error တက်လာမှာပါ။ age လို့ Declare လုပ်ခဲ့ရင် ခေါ်ယူအသုံးပြုတဲ့အခါမှာလည်း age လို့ပဲ ခေါ်ယူအသုံးပြုရမှာပါ။ Age လို့ Declare လုပ်ခဲ့ရင်လည်း Age အနေနဲ့ပဲ ခေါ်ယူအသုံးပြုရမှာပါ။ အဲဒါလိဖြစ်တာကို case sensitivity ဖြစ်တယ်လို့ ခေါ်ပါတယ်။ ဒါပေမယ့် တရာ့၏ Language တွေမှာတော့ မဖြစ်ပါဘူး။ age လို့ Declare လုပ်ခဲ့ပေမယ့် Age ပဲဖြစ်ဖြစ်၊ AGE ပဲဖြစ်ဖြစ် ခေါ်ယူအသုံးပြုလို့ရပါတယ်။ ဥပမာ အနေနဲ့ပြောရရင်တော့ ဒီနေ့ မြန်မာနိုင်ငံအပါအဝင် နိုင်ငံတော်တော်များများမှာပါ family ဖြစ်နေတဲ့ Visual Basic ရဲ့ မိခင် BASIC Language, Pascal စဲတဲ့ High-Level-Language တွေမှာ case sensitivity မဖြစ်ပါဘူး။ ဒါကြောင့် JAVA Language မှာတော့ program code တွေရေးရာမှာ Declare လုပ်တဲ့အမည်နဲ့ ခေါ်ယူအသုံးပြုတဲ့အမည်ကို သတိပြုမသာဘူး။ Case Sensitivity မဖြစ်မှာပါ။

ကျော်လေး (ကသာ)

# JAVA Developer Guide

## 1.4 Appendix of control statements in JAVA

JAVA Language အကြောင်းကို တင်ပြတဲ့နေရာမှာ အခြေခံ ကောင်းဖြစ်အောင် ပြီးတော့ အခြား Language တွေနဲ့ ထိပြုး အမြဲးမြဲတဲ့ သူတွေအတွက်ပါ ရည်ရွယ်ခဲ့တဲ့ အတွက် သင် ကြားပို့ချုတဲ့ နေရာမှာ စနစ်ကျအောင်၊ လွယ်ကူအောင် သက်ဆိုင်တဲ့ Listing တွေနဲ့ ယူဝါတွဲပြီး ပို့ချေလေးရှုပါတယ်။ ဒီအခါမှာ 1.3 Statements Expression and Operators အခန်းမှာ ပို့ချုံခဲ့တဲ့ တာတွေဟာ လိုအပ်လာဖြစ်ဖို့ ကြားဖြစ် ပို့ချေခြင်းပဲ ဖြစ်ပါတယ်။ ဒါကြောင့်မိ အခုံ့ချုတဲ့ အခန်းကဏ္ဍဟာ 1.2 Decision, control Structure and Looping Statements of JAVA ရဲ့ အဆက်လိုပဲ မှတ်ယူထားပါ။ ပို့ပြုးပြည့်စုံသွားအောင်လို့ နောက် ဆက်တွေအနဲ့ ဖြည့်စွက်တယ်ဆိုရင်လည်း မမှားပါဘူး။ ကဲ ဆက်လက်ပြီး လေ့လာလိုက်ပါၤီး။

### The While Statement

JAVA Language မှာ for statement လိုပဲ Looping (OR) Repetition လုပ်ပေးတဲ့ control statement တစ်ခုရှိပါတယ်။ အဲဒါကတော့ while statement ပါပဲ။ While Statement ရဲ့ Syntax\_form ကို အောက်မှာ ရေးပြထားပါတယ်။

```
while (boolean expression) {
    statement 1;
    statement 2;
    . . . .
}
```

While Statement မှာ ပါဝင်တဲ့ Syntax\_form ဟာ အပေါ်မှာ ရေးပြထားတဲ့ အတိုင်းပါပဲ။ Program statement ဟာ while ကို ရောက်တဲ့ အချင့် while ရဲ့ နောက်က expression ကို မှန်မမှန် စီစဉ်ပါတယ်။ Boolean expression လို့ ရေးထားရတဲ့ အကြောင်းရင်းက data type သေားကို လေ့လာစိုက် Boolean Type မှာ TRUE (OR) FALSE ဆိုတဲ့ conditions နှစ်ခုပဲ ရှိတယ်ဆိုတာကို သိထားမှာပါ။ အဓိပ္ပာယ်ကတော့ အဆိုပြချက်မှန် နေသရွာ။ while ရဲ့ နောက်က brace { ထဲကိုဝင်ပြီး statement ကို အလုပ်လုပ်နေမှာပါ။ အကယ်၍ while ရဲ့ နောက်က expression ကို စီစဉ်လို့ FALSE ဆိုတဲ့ condition ကို ရော်ချိရင် တော့ While ရဲ့ နောက်က brace { ထဲကို မဝင်တော့ဘဲ Looping ဟာ ရှင်သွားပြီတော့ while Loop ရဲ့ close brace }

အောက်က program statement တွေကိုပဲ အလုပ်ဆက်လုပ်သွား တော့မှာပါ။ ကဲ ဒါဆိုရင်တော့ while loop ရဲ့ flow ကို နားလည်သွားပြီးထင်ပါတယ်။ ပို့ပြုးနားလည်သောာပေါက်သွားအောင်လို့ Listing 1.11 နဲ့ ရေးပြီး ရင်းပြထားပါတယ်။ လေ့လာကြည့်ပါ။ သတိပြုရမယ့်အချက်ကတော့ while statement ရဲ့ နောက်က test expression (boolean expression) မှာ semicolon (;) မပါရှိရပါဘူး။ ကဲ စတင် လေ့လာကြပါၤီး။

### Listing 1.11 Demonstration the while Loop

```
1. class WhileClass{
2.     public static void main (String args []
3.     ) {
4.         int i=0;
5.         while (i<5) {
6.             i = i +1;
7.             System.out.println("Value of i=: "+i);
8.         }
9.     }
10.    output
11.    Value of i=1
12.    Value of i=2
13.    Value of i=3
14.    Value of i=4
15.    Value of i=5
16.    Analysis
```

၁။ လိုင်းနံပါတ် ၁ မှာ ရေးထားတာကတော့ WhileClass ဆိုတဲ့ primary class တစ်ခုကို ရေးဖွံ့ထားပါတယ်။ လိုင်းနံပါတ် ၂ မှာတော့ public static void main (String [ ]args) ဆိုတဲ့ WhileClass ရဲ့ main ()function ကို ရေးဖွံ့ထားပါတယ်။ အဖွင့် brace { တွေကိုလည်း သတိပြုမိမှာပါ။ ၂။ လိုင်းနံပါတ် ၃ မှာတော့ i=0; ဆိုပြီး int data type ကို အသုံးပြုပြီးတော့ ကြည့်ပါတယ်။ အဓိပ္ပာယ်ကတော့ program စတင်တဲ့ အချင့်မှာ i ရဲ့ တန်ဖိုးသည် (0-Zero) သာလျှင် ဖြစ်ပါတယ်။ ၃။ လိုင်းနံပါတ် ၄ မှာတော့ while statement ကို use လုပ်ပြီး တော့ boolean expression ကို စီစဉ်ပါတယ်။ အဓိပ္ပာယ်ကတော့ i ဟာ 5 ထက်ငယ်ရဲ့လား? ဆိုပြီး စီစဉ်တဲ့ အခါမှာ i ဟာ 0 ဆိုတော့ 5 ထက်ငယ်တယ်ပေါ့။ ဒီတော့ TRUE ဆိုတဲ့

## ကွန်ပူးတာသင်ခန်းစာ

<p>condition ကို ရှုံးတော့ while ရဲ့ block brace { ထဲကို ဝင်လျှေးတော့ i=i+1;ဆိုတဲ့ statement ကို အလုပ်လာလုပ်ပါတယ်။ ဒါအခါမှာ i ရဲ့ မူလတန်ဖိုးဟာ 0 ဆိုတော့ i = 0+1;ဆိုတဲ့ အဖြေကို ရလာပြီးတော့ လိုင်းနံပါတ် ၆ မှာ ရေးထားတဲ့ print statement ကို အလုပ်သွားလုပ်ပြီးတော့ value of i=1 ဆိုတဲ့အဖြေကို လာပြတာပါ။ ပြီးတဲ့ while statement ရဲ့ close brace } ကို တွေ့တဲ့အခါမှာ လိုင်းနံပါတ် ၄ မှာရေးထားတဲ့ while loop ရဲ့အစကို ပြန်သွားပါတယ်။</p> <p>၄။ While Statement ဟာ မှန်နေသရွှေ့ အလုပ်ဆက်လုပ်တယ် ဆိုတော့ ပြန်ပြီးတော့ test expression လုပ်ကြည့်တဲ့အခါမှာ i ဟာ 1 ဖြစ်နေပါပြီ။ ၁ ဟာ ၅ ထက်ကော်သလား? ဆိုတဲ့အဆိုကို စိစစ်တဲ့အခါမှာ ငါယ်ပါတယ်။ ဒါတော့ i ဟာ ၅ ထက်ကော်စွဲမှာ ဆိုတဲ့ brace { ထဲကို ဝင်လာပြီးတော့ i ရဲ့တန်ဖိုးကို ၁ လာပြီးတော့ Add လုပ်ပါတယ်။ ပြီးတော့ Value of i=2 ဆိုတဲ့အဖြေကို လာပြီး output လာပြေားပါတယ်။</p> <p>၅။ ဒီလိုနည်းနဲ့ i ဟာ ၅ ထက် မကြီးမချင်း စိစစ်ပြီးတော့ value of i=3, value of i=4, value of i=5 ဆိုတဲ့အဖြေကို print ထုတ်ပေးမှာပါ။ ဒါပေမယ့် i ရဲ့တန်ဖိုးကို ၅ အဖြစ် print ထုတ်ပေးပြီးတဲ့အခါမှာတော့ while loop ရဲ့အစကို ပြန်သွားပြီးတော့ test expression လုပ်ပါတယ်။ ဒီအခါမှာတော့ i ရဲ့တန်ဖိုးဟာ ၅ ဖြစ်နေပါပြီ။ ၅ ဟာ ၅ ထက်ကော်သလား? ဆိုတဲ့ စိစစ်ချက်မှာ မင်္ဂလာကြော်တွေ့လို့ အဖြေကို ထွက်ပါတယ်။ ဒါတော့ boolean expression မှာ FALSE ဖြစ်သွားပါပြီ။ FALSE condition ကို ရလာပြီးဆိုတော့ while looping ကို escape ဖြစ်ပြီးတော့ လိုင်းနံပါတ် ၇ မှာရှိတဲ့ main () function ကို ရေးဖွဲ့ထားပါတယ်။ အဖွင့် brace တွေကို မရေ့ဖို့စေနဲ့။</p> <p>၃။ လိုင်းနံပါတ် ၁ မှာတော့ int i ရဲ့ program စစ်တန်တန်ဖိုးသည် ၁ ပါလို့ ကြေညာပါတယ်။ အဲဒီနောက် လိုင်းနံပါတ် ၄ မှာတော့ while statement ကို အသုံးပြုပြီးတော့ test expression လုပ်ပါတယ်။ ဒီအခါမှာ i ရဲ့တန်ဖိုးသည် ၁ ဆိုတော့ ၆ နဲ့မညီ စဉ်မှာ while Loop ရဲ့ brace ထဲကို ဝင်ရမယ်လို့ ရေးထားတဲ့အတွက် brace { ထဲကိုဝင်ပြီးတော့ လိုင်းနံပါတ် ၅ မှာ i ရဲ့တန်ဖိုးကို print လုပ်ပါတယ်။ လိုင်းနံပါတ် ၆ မှာတော့ i ရဲ့ မူလတန်ဖိုးကို ၂ လာပေါင်းပါတယ်။ ဒါတော့ i ရဲ့တန်ဖိုးသည် ယခုအခါမှာ ၃ ဖြစ်သွားပါပြီ။ ပြီးတဲ့အခါမှာ while ရဲ့ close brace } ကို တွေ့တဲ့အခါမှာ လိုင်းနံပါတ် ၇ မှာရေးထားထားတဲ့ while statement ရဲ့ expression ဆီကို ပြန်သွားပါတယ်။ အပေါ်ကို ပြန်တက်သွားတာပေါ့။</p> <p>၄။ ဒီအခါမှာ ၁ ဟာ ၃ ဖြစ်နေပါပြီ။ ၃ ဟာ ၆ နဲ့မညီစဉ်မှာ while block ထဲကို ပြန်ဝင်သွားပြီး output ကို ထုတ်ပေးပါတယ်။ ပြီးတော့ ၂ ပေါင်းပြီးတော့ လိုင်းနံပါတ် ၈ ကို ကို ပြန်ရောက်သွားပြန်ပါပြီ။ နောက်တစ်ကြိမ် test လုပ်စဉ်မှာတော့ i ရဲ့ တန်ဖိုးသည် ၅ ဖြစ်နေပါပြီ။ အဲဒီအခါမှာလည်း while ရဲ့ block { ထဲကိုဝင်ပြီးတော့ statement တွေကို အလုပ်လုပ်ပြီးတော့ အဖြစ်ထုတ်ပေးပါ့မှာပါ။ ဒီလိုနည်းနဲ့ Loop ပတ်ရင်းနဲ့ ၁ ၃</p>
---

## ကွန်ပျူးတာသင်ခန်းဘ

<p>579...အစရှိတဲ့ မကဏ္နးတွေကို outputအဖြစ် ထုတ်ပေးနေပါတယ်။ Programဟာ သတ်မှတ်ကြညာထားတဲ့ data type ရဲ့ Limit ကို မရောက်မချင်း never ending Loop ဖြစ်သွားပါတယ်။</p> <p>၅။ ဘာကြောင့်လဆိတော့ Logical NOT Equal to operator ကို အသုံးပြုပြီး ရေးထားတဲ့ <math>i \neq 6</math>လို့ ရေးထားတဲ့အခါမှာ မည့်မချင်းလို့ ပြောထားတဲ့အတွက် condition ဟာ always TRUE ဖြစ်နေပါတယ်။ Condition ဟာ FALSE ဖြစ်မှ သာလျှင် Looping က ရပ်မှာဆိတော့ <math>i \neq 6</math>ဆိတဲ့ expression ဟာ ညီစရာအကြောင်းမရှိတော့ Looping ကလည်း never ending Loop ဖြစ်သွားပါတော့တယ်။ ဒီတော့ condition ဟာ FALSE မဖြစ်တော့သလို ရပ်လည်း မရပ်တော့ပါဘူး။</p> <p>၆။ လိုင်းနံပါတ် ၇ မှာ main ( ) function ရဲ့ close brace } ကို ပိတ်ထားပြီးတော့ လိုင်းနံပါတ် ၈ မှာတော့ while test ဆိတဲ့ Java primary class ရဲ့ close brace } ကို လာပိတ်ထားပါတယ်။ ဒီနေရာမှာ program ဟာ ပြီးသွားပြီဆိုပေမယ် အမှန်တကယ်တော့ while condition ဟာ FALSE မဖြစ်တော့ while block ထဲကနေ ထွက်လို့မရတော့တဲ့အတွက် ကြောင့် main ( ) function နဲ့ primary class ရဲ့ close brace } တွေကိုတောင် လာလို့မရတော့ပါဘူး။</p> <p><b>Note:</b> ဒီ program ကို Run လုပ်ပြီး မလေ့လာသင့်ပါဘူး။ လက်နဲ့စာရွက်ပေါ်မှာ ချွောက်ချုပ်ဖြစ်စေ၊ ထပ်ခါထပ်ခါ သေသေချာချာဖော်ချုပ်ဖြစ်စေ လေ့လာသင့်ပါတယ်။ Logical Operator အပါအဝ် operators ပျို့စုံအကြောင်းကို ဒီသင်ခံးစာမတိုင်းခေါ်မှာ ဘာကြောင့် ပို့ချေရာလဆိတာကို အခုအခါမှာ သဘောပေါက်သွားမှာပါ။ ပြီးတော့ while statement ရဲ့ control flow ကို သိအောင် Listing တွေကို တွက်ချက်ပြုတဲ့အခါမှာ output တွေကို ရှင်းရှင်းလင်းလင်းဖြင့်ရအောင်လို့ System.out.println ()ဆိတဲ့ နောက်တစ်ကြောင်းကိုဆင်းပြီးမှ output ကို ထုတ်ပေးစေတဲ့ print command ကို အသုံးပြုထားပါတယ်။ ဒါကြောင့်သာ output တွေဟာ Line By Line ဖော်ပြီး လာပြတာပါ။</p> <p><b>The do-while statement</b></p> <p>JAVA Language မှာ Looping တစ်ခုကို while statement ကို အသုံးပြုပြီးတော့ ပတ်တဲ့အခါမှာ Looping block ထဲကို ဝင်စေသင့်မှု/မဝင်သင့်မှု Looping ရဲ့ အစမှာတင် test လုပ်ပါတယ်။ Test လုပ်တဲ့အခါမှာတော့ boolean expression အနေနဲ့ပေါ်။ ဒါပေမယ့်ပထမအကြိမ်မှာတော့ အကန့်အသတ်မရှိလုပ်ခွင့်ပြုပြီးတော့ နောက်တစ်ကြောင်းကိုဆင်းပြီးမှ output ကို ထုတ်ပေးစေတဲ့ print command ကို အသုံးပြုပါတယ်။ အောက်မှာ do-while</p>	<p>statement ရဲ့ Syntax_form ကို ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။</p> <pre>do {     statement 1;     statement 2;     ..... } while (boolean expression);</pre> <p>ဒီ syntax_form မှာ ရေးထားတာကို လေ့လာကြည့်မယ်ဆိုရင် program အလုပ်စလုပ်စဉ်မှာ အကန့်အသတ် တားဆီးမှု မရှိဘဲနဲ့ while block { } ထဲကို ဝင်စေပါတယ်။ ဒါပေမယ့်နောက်တစ်ကြိမ် အလုပ်ဆက်လုပ်ခွင့်ပြု/မပြုတိတော့ while block ရဲ့အဆုံးမှာ test လုပ်စေပါတယ်။ ဒီအခါမှာ while ရဲ့ နောက်က (test expression) ကို စစ်လိုက်တဲ့အခါမှာ TRUE condition ဖြစ်လာခဲ့ရင် do-while Loop ကို နောက်တစ်ကြိမ် ပတ်ခွင့်ပြုမှာပါ။ အကယ်၍ test expression ဟာ FALSE condition ဖြစ်လာခဲ့ရင်တဲ့အတွက် Looping ကို ပတ်ခွင့်မပြုတော့ဘဲ while ရဲ့အောက်မရှိဘဲ အခြားသော program statement တွေဆိုကိုပဲ သွားပြီး အလုပ်ဆက်လုပ်တော့မှာပါ။ က ဒါဆိုရင်တော့ do-while statement ရဲ့ သဘောတရားကို နားလည်သဘောပါက်သွားမှာပါ။ Listing 1.3 ကို အသုံးပြုပြီးတော့ do-while statement ကို ရှင်းပြုထားပါတယ်။ အခြားသောတရားတွေပါဝင်တဲ့အတွက် သေသေချာချာ စနစ်တာကျ လေ့လာကြည့်ပါ။</p> <p><b>Listing 1.13</b> Illustrate in do-while Looping In JAVA Language</p> <pre>1. class doWhileClass { 2.     public static void main (String args [ ]) { 3.         int i=0; 4.         do { System.out.println ("Value of i=" + i); 5.             i=i+1; 6.         } while (i&lt;5); 7.     } 8. }</pre> <p>output value of i=0 value of i=1 value of i=2 value of i=3 value of i=4</p> <p><b>Analysis</b></p> <p>၁။ ဒီ program ရဲ့ လိုင်းနံပါတ် ၁ မှာ doWhileClass ဆိတဲ့ JAVA ရဲ့ primary class ကို ဖန်တီးထားပါတယ်။ လိုင်းနံပါတ် ၂ မှာမှာ main ( ) function ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် ၃ မှာတော့ int data type ကို အသုံးပြုပါတယ်။</p>
---	---

## ကွန်ပူးတာသင်ခန်းစာ

i=0;လို့ declare လုပ်ပါတယ်။ ဒီတော့ program အစမှာ တော့ i ရဲ့ တန်ဖိုးဟာ 0 ဖြစ်နေပါပြီ။

၂။ လိုင်းနံပါတ် ငါ မှာတော့ do statement ကို ရေးသားခဲ့ပါ တယ်။ စစ်ချင်းမှာတော့ do ရဲ့ block ထဲကို အတားအဆီးမရှိ ဝင်လာခဲ့ပြီး value of i=0 ဆိတ် output ကို လာပြုပါတယ်။ လိုင်းနံပါတ် ၅ မှာ ရေးထားတဲ့ while (i<5) ဆိတ် test expression ကို လာစိစစ်တဲ့ အခါမှာတော့ i ရဲ့ တန်ဖိုးဟာ ၁ အနေနဲ့ပါ။

၃။ ဒီအခါမှာ do-while block ထဲကို ဝင်သင့်မဝင်သင့် ဆုံးဖြတ် တဲ့ အခါမှာ i ဟာ ၅ ထက် ငယ်နေသေးတဲ့ အတွက် ခြောင်း ဒုတိယအကြိမ် Looping block ထဲကို ပြန်ဝင်လာပြန်ပါ တယ်။ ဒီတော့ value of i=1 ဆိတ် အဖြောက် လာပြုပါတယ်။ ပြီးတော့ ၅ ထက် ငယ်/မငယ်ကို while နဲ့ use လုပ်ပြီးတော့ စစ်ဆေးပါတယ်။ ဒီအခါမှာ condition ဟာ TRUE ဖြစ်တော့ do-while Looping ကို အလုပ်ဆက်လုပ်ခွင့်ပြုတာပေါ့။

၄။ ဒီလိုနည်းနဲ့ စစ်ဆေးပြီးတော့ do-while statement ကို အလုပ်ဆက်လုပ်ပါတယ်။ ဒီအခါမှာတော့ Loop ကို ၅ ကြိမ် မြောက် ပတ်တဲ့ အခါမှာ (i ရဲ့ တန်ဖိုးဟာ ၅ ဖြစ်စဉ်မှာတော့) i ရဲ့ တန်ဖိုးဟာ ၅ ထက်မငယ်တော့တဲ့ အတွက် Looping ကို ခွင့်မပြုတော့ပါဘူး။ ဒါဆိုရင်တော့ Looping ဟာ ၄ ကြိမ်ပဲ ခွင့်ပြုတာပေါ့။ ဟုတ်ပါတယ်။ သေသေချာချာ လေ့လာကြည့်ရင် value of i=0 ဆိတ် လာပြစ်တော့ do-while Loop ထဲကို အတားအဆီးမရှိ ဝင်လာတာပါ။ i ရဲ့ တန်ဖိုး ၁ က စပြီးတော့မှာ သော Looping ပတ်ခွင့်ပြုတာရှိ။ Looping ကို ငါ ကြိမ်သာ ပတ်ခွင့်ပြုတယ်လို့ ပြောတာပါ။

၅။ လိုင်းနံပါတ် ၇ မှာ main ( ) function ရဲ့ close brace } ကို လာပိတ်ထားပြီးတော့ လိုင်းနံပါတ် ၈ မှာတော့ primary class ရဲ့ class brace } ကို လာပိတ်ထားတာကို တွေ့ရမှာပါ။ က ဒါဆိုရင်တော့ do-while statement အကြောင်းကို နားလည်းသောပါကြသွားလောက်ပါပြီ။ သက်ဆိုင်ရာ project တွေထဲမှာတော့ ထည့်သွင်းပြီး ရေးသားရှင်းပြေားလီးမှာပါ။ အခုတော့ အခြေခံအနေနဲ့ပဲ ရှင်းပြုတာပါ။

**Note:** while Loop ကို ရေးသားအသုံးပြုတဲ့ အခါမှာ while statement ရဲ့ နောက်မှာ semicolon (;) မပါရှိပါဘူး။ ဒါပေမယ့် do-while statement မှာတော့ while ရဲ့ နောက်မှာ (do-while block ရဲ့ အပိုဒ် while (boolean expression) ရဲ့ နောက်မှာ) Semicolon (;) ပါမှသာလျှင် အလုပ်ဆက်လုပ်တာပါ။ ဒီအချက်ကို သေသေချာချာလေ့လာထားပါ။ While Statement နဲ့ do-while statement ကွာခြားပုံကတော့ while statement သည် program စတင်စဉ်မှာထဲက အလုပ်လုပ်ခွင့်ပြုသင့်/ မပြုသင့်ကို

စီစစ်ပြီးတော့ do-while statement မှာတော့ program စလျင် စဉ်းစောင်း အတားအဆီးမရှိဘဲ အလုပ်ကို တစ်ကြိမ်ခွင့်ပြုပါတယ်။ နောက်တစ်ကြိမ်မှသာလျှင် test လုပ်ပြီးတော့ လုပ်ခွင့်ပြုသင့်/ မပြုသင့်ကို ဆုံးဖြတ်တာပါပဲ။

က ဒါဆိုရင် Looping statement တွေအကြောင်းကို နားလည်းသောပေါက်သွားလောက်ပါပြီ။ ဒီတော့ ဆက်လက်ပြီး continue statement နဲ့ break statement တို့အကြောင်းကို ရှင်းပြုထားပါတယ်။ ဆက်လက်လေ့လာကြည့်ပါပြီး။

### The Continue Statement

Continue Statement ဆိုတာကတော့ Looping တစ်ခုကိုပတ်နေဆဲမှာ statement တွေကို အကုန် execute မလုပ်ဘဲနဲ့ သတ်မှတ်ထားတဲ့ continue statement ရဲ့ နေရာကို ရောက်တဲ့ အခါမှာတော့ Looping ကို အစကနေပြီးတော့ ပြန်ပတ်ပါတယ်။ Continue statement ကို for, while, do-while စတဲ့ Looping Technique တွေနဲ့ ပေါင်းစပ်အသုံးပြနိုင်ပါတယ်။ အောက်ပါ continue statement ရဲ့ syntax\_form ကို ရေးပြထားပါတယ်။ လွယ်ကူပြီး ရှိုးရှင်းပါတယ်။

continue;

က လက်တွေ၊ အနေနဲ့ continue statement ကို Listing 1.14 မှာ ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။ Continue Statement အကြောင်းကို သောပေါက်သွားမှာပါ။

### Listing 1.14 Demonstrate the continue statement

```
1. class continueClass{  
2.     public static void main (String args  
3.     [ ]) {  
4.         int i=10;  
5.         while (i<=100){  
6.             i += 10  
7.             if (i<=50) continue; //goto while test  
expression  
8.             System.out.println ("Value of i=" +i);  
9.         }  
10.    }  
11.    Output  
12.    Value of i=60  
13.    Value of i=70  
14.    Value of i=80  
15.    Value of i=90  
16.    Value of i=100  
17.    Value of i=110
```

Analysis  
၁။ လိုင်းနံပါတ် ၁ မှာ continueClass ဆုံးတဲ့ primary class ကို ရေးဖွံ့ထားပါတယ်။ main () function ကိုတော့ လိုင်းနံပါတ် ၂ မှာ ရေးထားတာကို တွေ့ရမှာပါ။ လိုင်းနံပါတ် ၃

## ကွန်ပူးတာသင်ခန်းစာ

- မှာတော့ int data type ကို အသုံးပြုပြီးတော့  $i = 10$ ; လို့ declare လုပ်ပါတယ်။
- J) လိုင်းနံပါတ် ငဲ့ မှာတော့ while statement ကို အသုံးပြုပြီး တော့  $i$  ဟာ 100 နဲ့ညီမယ်/ ငယ်မယ်ဆိုရင် brace { (while block)ထကို ဝင်ခွင့်ပြုမယ်ဆိုပြီး ရေးသားထားပါတယ်။ 100 ထက်ငယ်ရင်ဖြစ်စေ ညီရင်ဖြစ်စေ while ရဲ့ block { ထကို ဝင်ခွင့်ရှုပါတယ်။ ကဲ ဒီအထိရပြုနော်။
- ၃) လိုင်းနံပါတ် ၅ မှာတော့  $i$  ရဲ့တန်ဖိုးထကို 10 ထည့်ပေါင်းစေစွဲ အတွက် self-assign (=) symbol ကို သုံးပြီးတော့ ရေးသားထားတာကို တွေ့ရမှာပါ။ ကဲ ဒါဆိုရင်တော့  $i$  ရဲ့တန်ဖိုးဟာ 20 ဖြစ်သွားပြီဆိုတာ သိထားရမယ်နော်။ လိုင်းနံပါတ် ၆ မှာတော့ if ( $i <= 50$ ) continue; လို့ ရေးသားတာကတော့  $i$  ဟာ 50 နဲ့ ညီစွဲ/ ငယ်စွဲ ဆက်လုပ်ပါဆိုတော့ ငယ်စွဲရင် ဆိုတဲ့ အခြာအနေကြောင့် လိုင်းနံပါတ် ၇ မှာရှိတဲ့ print command ကို အလုပ်သွားမလုပ်ဘဲနဲ့ လိုင်းနံပါတ် ငဲ့ မှာရှိတဲ့ while Loop ဆိုတဲ့ ပြန်သွားပြီးတော့  $i = 20$  ဆိုတဲ့ တန်ဖိုးနဲ့ သွားပြီး test လုပ်ပါတယ်။ ဒီဒီအခါမှာလည်း  $i$  တန်ဖိုးဟာ 100 နဲ့ညီမယ်/ ငယ်မယ်ဆိုတဲ့ အနေအထားအထိ ခွင့်ပြုထားတာကြောင့် while Loop ရဲ့ block ထကို ပြန်ဝင်လာခွင့်ပြုပါတယ်။
- ၄) ဒီအခါမှာတော့  $i$  ရဲ့တန်ဖိုးကို self-assign (=) symbol operator ကို အသုံးပြုပြီးတော့ 10 ပေါင်းပြန်ပါတယ်။ ဒီအခါမှာ  $i$  ရဲ့တန်ဖိုးဟာ 30 ဖြစ်သွားပြန်ပါပြီ။ လိုင်းနံပါတ် ၆ မှာ if statement ကို အသုံးပြုပြီးတော့  $i$  ဟာ 50 နဲ့ ညီစွဲ/ ငယ်စွဲလျှင်လို့ ထပ်စစ်ပါတယ်။  $i$  ရဲ့တန်ဖိုးက 30 ဆိုတော့ ငယ်နေပါသေးတယ်။ ဒီတော့ if (expression) ရဲ့နောက်မှာ ရေးထားတဲ့ continue statement အရ လိုင်းနံပါတ် ငဲ့ မှာရှိတဲ့ while ကို ပြန်သွားပြန်ပါပြီ။
- ၅) ဒီလိုနည်းနဲ့ အလုပ်လုပ်တဲ့အခါမှာ Looping ကို ၆ ကြိမ် တိတိ ပတ်ပါတယ်။ ဒီတော့  $i = 60$  ဆိုတဲ့ အဖြစ်ကို ရတဲ့အထိ ပေါ့နော်။ ကဲ ဒီတော့  $i = 60$  လို့ ရဟန်တဲ့အချင်မှာတော့ လိုင်းနံပါတ် ၆ မှာရေးထားတဲ့ if ( $i <= 50$ ) ဆိုတဲ့နောက် လာစီစစ်တဲ့အခါမှာ 60 ဟာ 50 ထက်မယ်သလို ညီလဲ မည့် တော့ပါဘူး။ ဒီအခါမှာ continue statement ဟာ အလုပ်ဆက်မလုပ်တော့ဘဲနဲ့ နောက်တစ်ကြောင်း လိုင်းနံပါတ် ၇ ကို ဆက်ဆင်းလာပါပြီ။
- ၆) လိုင်းနံပါတ် ၇ ကို ရောက်တဲ့အခါမှာ value of  $i = 60$  ဆိုတဲ့ output ကို ထူတ်ပေးလိုက်ပါတယ်။ ဒါပေမယ့် loop ကနေ ထွက်မသွားနိုင်သေးတာ သတိပြုပါ။  $i$  ရဲ့ တန်ဖိုး 60 နဲ့ Loop ကို ဆက်ပတ်ပါတယ်။ 100 ထက် ငယ်ဆဲစွဲ Loop ထကို ဆက်ဝင်ပြီး 10 ထပ်ပေါင်းထည့်တယ်။ if statement

ထဲမှာ 50 ထက် ငယ်(သို့)ညီခြင်းမရှိတဲ့အတွက် continue statement သို့ မရောက်တော့ဘဲ။  $i = 70$  ကို print out ထုတ်ပါတယ်။ ဒီနည်းအတိုင်း ပင်ရင်း while Loop ရဲ့ test boolean expression ရဲ့တန်ဖိုး true ဖြစ်နေသရွှေ့ Loop ဆက်ပတ်ပါတယ်။

ဒါ Listing 1.14 ကို လေ့လာကြည့်ရင်  $i$  ရဲ့တန်ဖိုး 50 ထက် ငယ်(သို့)ညီနေသေးရင် print out ထုတ်မပြတာကို တွေ့ရပါမယ်။ ဘာကြောင့်လဲဆိုတော့ အထက်ပါအခြာအနေမှန်နေသေးသရွှေ့ continue statement ကို လုပ်ဆောင်တာကြောင့် System.out.println("Value of i=" + i); ဆိုတဲ့ statement သို့ မရောက်တော့ဘဲ while Loop ရဲ့ test boolean expression ( $i <= 100$ ) သို့ ပြန်ပြန်ရောက်သွားလို့ပဲ ဖြစ်ပါတယ်။ Loop ထဲက အပြီးထွက်သွားတာတော့ မဟုတ်ပါ။ ဒါဟာ ဆက်လေ့လာမယ့် break statement နဲ့ ကွာခြားပါတယ်။

### The break statement

Break Statement ဆိုတာကတော့ Looping ပတ်နေရက နေ ခန့်ထွက်ဖို့၊ တိတိကျကျပြောရရင်တော့ ရောက်ဆဲ current Looping ကနေ exit လုပ်ဖိုးအတွက် အသုံးပြုရတဲ့ control statement တစ်ခုပါပဲ။ Switch statement နဲ့ တွဲဖက်ပြီး exit လုပ်ဖိုးလည်း အသုံးပြုနိုင်ပါတယ်။ ဒါကြောင့် break statement ကို while, do-while, for နဲ့ switch statement တွေမှာ တွဲဖက်ပြီး အသုံးပြုနိုင်ပါတယ်။ ကဲ ဒါဆိုရင်တော့ break statement ရဲ့ syntax\_form ကို ရေးပြထားပါတယ်။

break;

Break Statement အကြောင်းကို Listing 1.6 using the Switch Statement ဆိုပြီးမှ Switch Statement နဲ့ တွဲဖက်ပြီး သင်ကြားပို့ချပေးခဲ့ပြီးပါပြီ။ ဒါပေမယ့် ဒီနေရာမှာတော့ သက်ဆိုင်ရာအခန်းကဏ္ဍအလိုက်တော့ ရှင်းပြရမှာဖြစ်လို့ Listing 1.15 အနေနဲ့ ရေးပြီး ရင်းပြထားပါတယ်။ အခုရေးပြတဲ့ Listing 1.15 ကို လေ့လာပြီးရင်တော့ Listing 1.6 ကို ပြန်လေ့လာသင့်ပါတယ်။ ဒါမှာသွေ့ဗောပေါက်သွားမှာပါ။ ကဲ Listing 1.15 အနေနဲ့ break statement ကို လေ့လာလိုက်ပါပြီး။

### Listing 1.15 Using the break statement in JAVA

```
1. //exit the current Looping in current block
2. class breakClass{
3.     public static void main (String args[]){
4.         int i=10;
```

## ကွန်ပူးတာသင်ခန်းစာ

```

5. while(i<=100){
6. i+=100; //use in self-assign operator
7. if (i== 50) break;
8. System.out.println ("Value of
i=" +i);
9. }
10. }
11. }
output
value of i=20
value of i=30
value of i=40
Analysis

```

၁။ လိုင်းနံပါတ် ၁ မှာရေးထားတဲ့ (// Backslash symbol) သက်တစ်ခြီး ရေးထားတဲ့ စာကြောင်းကတော့ comments အနေနဲ့ ရေးထားတာပါ။ ထည့်ရေးရှင်လည်း JAVA Interpreter က အသိအမှတ်မပြုပါဘူး။ ထည့်ရေးပို့လည်း ရပါတယ်။ ရှေ့က Double Backslash (//) သက်တကိုတော့ မမေ့စေနဲ့ပေါ့။

၂။ လိုင်းနံပါတ် ၂ မှာတော့ breakClass ဆိုတဲ့ Java ရဲ့ primary class ကို ဖုန်တီးပြီး လိုင်းနံပါတ် ၄ မှာတော့ main () function ကို ရေ့ဖွဲ့ထားပါတယ်။ လိုင်းနံပါတ် ၄ မှာတော့ int data type ကို အသုံးပြုပြီးတော့ i = 10 ဆိုတဲ့တန်ဖိုးတို့ Declare လုပ်ပါတယ်။ လိုင်းနံပါတ် ၅ မှာတော့ while statement ကို အသုံးပြုပြီးတော့ i ရဲ့တန်ဖိုးဟာ 100ထက်ငယ်စဉ်ကြီးစဉ်မှာဆိုပြီး test လုပ်ပါတယ်။ While statement ကို အသုံးပြုထားတာဖြစ်တဲ့အတွက် program ကို စစချင်းမှာပဲ test လုပ်ပြီးတော့ စီစဉ်ပါတယ်။ do-while statement မှာလိုပြီး block ထဲကို ဒီအတိုင်း မဝင်စေချင်ပါဘူး။ ကဲ စီစဉ်ပိုက်တဲ့အခါမှာ i ရဲ့တန်ဖိုးဟာ 100ထက်ငယ်စဉ်ကြီးစဉ်မှာဆိုပြီး လော့လုပ်ပါတယ်။

၃။ လိုင်းနံပါတ် ၆ မှာတော့ i ရဲ့တန်ဖိုးကို self-assign (+=) operator ကို အသုံးပြုပြီးတော့ Addလုပ်ပါတယ်။ ဒီအခါမှာ i ရဲ့တန်ဖိုးဟာ 20 ဖြစ်လာပါပြီ။ ဒီအခါမှာ i ရဲ့တန်ဖိုးဟာ 20 ဆိုတာကို အသုံးပြုပြီးတော့ i ရဲ့တန်ဖိုးဟာ 50 နဲ့ ညီပြီလား? ဆိုပြီး if statement ကိုသုံးပြီး စီစဉ်ပါတယ်။ အကယ်၍ i ရဲ့တန်ဖိုးသည် 50 နဲ့ညီပြီလျှင်တော့ break statement ကြောင်း အဖြေ output ကို ရိုက်ထုတ်မပြတော့ဘဲ လိုင်းနံပါတ် ၉ မှာ လာဝိတ်ထားတဲ့ while block ရဲ့ close brace } ကိုရောက်ပြီး block အပြင်ကို ခန်းထွက်သွားမှာပါ။ ဒါပေမယ့် i ရဲ့တန်ဖိုးသည် လက်ရှိအချိန်မှာ 20 ပဲဖြစ်နေတော့ 50 နဲ့ ညီပြီပါဘူး။ ဒီတော့ while ရဲ့ block အပြင်ကို ခန်းထွက်လို့မရပါဘဲနဲ့ လိုင်းနံပါတ် ၈ မှာရေးထားတဲ့ print state-

ment ကြောင်း value of i=20 ဆိုတဲ့အဖြေကို output အနေနဲ့ လာပြေးပါတယ်။

၄။ လိုင်းနံပါတ် ၉ မှာ ရေးထားတဲ့ while block ရဲ့ close brace } ကို ထွေ့တဲ့အခါမှာတော့ while ရဲ့ အစဉ်း test expression ကို ပြန်သွားပြီးတော့ while looping ရဲ့ block ထဲကို ဝင်သင့်မဝင်သင့် test လုပ်ပါတယ်။ ဒီအခါမှာ သူရဲ့ Limited data ဟာ 100 နဲ့ညီရိုက်ဖြစ်စေ ငယ်ရှင်ဖြစ်စေ ဝင်ခွင့်ပြုတဲ့အတွက် Looping ရဲ့ block ထဲကို ဒုတိယအကြိမ် ပြန်ဝင်လာပြန်ပါပြီ။

၅။ Looping block ထဲကိုရောက်ရောက်ချင်း i += 10; ဆိုတဲ့ statement ကြောင်း မူလတန်ဖိုး 20 ထဲကို 10ထည့်ပေါင်းပြီး 30 ဖြစ်လာပါပြီ။ ဒီအခါမှာ if (i == 50) break; ဆိုပြီး ရေးထားတဲ့ statement အနေနဲ့ test လုပ်တဲ့အခါမှာ i ရဲ့ တန်ဖိုးဟာ 30 ဆိုတော့ if statement နဲ့ မကိုက်ညီပါဘူး။ ဒီတော့ i ရဲ့တန်ဖိုးဟာ 30 ဆိုတော့ လိုင်းနံပါတ် ၈ မှာ ရေးထားတဲ့ print statement ကို အလုပ်သွားလုပ်ပြီးတော့ value of i=30 ဆိုတဲ့ output ကို လာထုတ်ပေးပါတယ်။

၆။ ဒီလိုနည်းနဲ့ပဲ တတိယအကြိမ်မြောက် i ရဲ့တန်ဖိုးဟာ 30 အနေနဲ့ပဲ while block ထဲကို ဝင်လာပြန်မှာပါ။ ဒီအခါမှာတော့ i ရဲ့တန်ဖိုးဟာ 40 ဖြစ်သွားမှာပါ။ ဘာကြောင်းလဲဆိုတော့ block ထဲကို ရောက်ရောက်ချင်း i += 10; ဆိုတော့ 30 + = 10; ဆိုတော့ i=40 ဖြစ်သွားပြီပေါ့။ သူကလည်း 50 နဲ့ မညီသေးတဲ့ အတွက် အပြင်ကိုမထွက်နိုင်ဘဲ value of i=40 ဆိုတဲ့အဖြေကို output ထုတ်ပေးပြီးမှာပါ။

၇။ နောက်တစ်ကြိမ် while block ထဲကို ဝင်လာတဲ့အခါမှာတော့ i ရဲ့တန်ဖိုးဟာ 40 ပါ။ ဒီအခါ Looping ရဲ့ block ထဲမှာ i += 10; ဆိုတဲ့ self-assign method နဲ့ 10ကို ထည့်ပေါင်းလိုက်တဲ့ အခါမှာ i ရဲ့တန်ဖိုးဟာ 50 လို့ ဖြစ်လာပါပြီ။ ဒီတော့ လိုင်းနံပါတ် ၈ မှာရေးထားတဲ့ if (i == 50) break; ဆိုတဲ့ statement expression နဲ့ သွားရောက်စစ်ဆေးတဲ့အခါမှာ မှန်နေတဲ့အတွက် break statement ဟာ သက်ဝင်လာပြီးတော့ လိုင်းနံပါတ် ၉ မှာရေးထားတဲ့ while block ရဲ့ close brace } အပြင်ကို ခန်းထွက်သွားပါပြီ။

၈။ ဒီအခါမှာ program ဟာ while Loop ရဲ့နောက်က statement တွေကို အလုပ်ဆက်လုပ်လို့ပြီးတော့ လိုင်းနံပါတ် ၁၀ မှာရေးထားတဲ့ main () function ရဲ့ close brace } ဆိုကို ဆက်ပြီး ဆင်းလာပါတယ်။ အဲဒါနောက်မှာတော့ လိုင်းနံပါတ် ၁၁ မှာ ရေးထားတဲ့ close brace } ဆိုကိုပါ ဆက် ဆင်းလာပြီးတော့ program ဟာ ပြီးသွားပါပြီ။ ကဲ ဒါဆိုရင်တော့ break statement အကြောင်းကို သေသေချာချာ နားလည် သဘောပေါက်သွားလောက်ပါပြီ။ အလွန်တရာ အသုံးဝင်တဲ့

ကွန်ပျိုတာသင်ခန်းစာ

အပြီးထွက်သွားပါတယ်။ continue statement ကို တွေ့တဲ့အခါ Loop test boolean expression သို့ ရောက်ရှိသွားခြင်းသာဖြစ်ပါတယ်။ အဲဒီ Logical expression အရသာ Loop ကိုဆက်ပတ်သင့် မပတ်သင့် ဆုံးဖြတ်တာဖြစ်ပါတယ်။ Loop ထက်အပြီးထွက်သွားတာမဟုတ်ပါ။

က ဒါဆိုရင်တော့ အခ သင်ကြားနို့ချွဲတဲ့ control statement တွေအကြောင်းကို လေ့လာကြည့်မယ်ဆိုရင်တော့ JAVA Language အကြောင်းကို ထဲထဲဝင်ဝင် နားလည်းသွားဖြစ်ပို့ ယုံကြည်ပါတယ်။ ရှေ့မှာ ဆက်လက်ပို့ချမယ့် သင်ခန်းစာတွေက တော့ chapter 1 ရဲ့ index တွေဖြစ်တဲ့ Array အကြောင်း၊ Array တွေကို ဖန်တီးရင် အသုံးပြုရတဲ့ new keyword ရဲ့ အကြောင်း၊ new keyword ကို အခြေခြားပို့ချမယ့် Memory Leaks နဲ့ Creating objects on the free store ဆိုတဲ့ အခန်းကဏ္ဍ ပြီးတော့ JAVA Byte code ရဲ့ အကြောင်း၊ JAVA Byte code ရဲ့ အဓိပ္ပာဇာတ် programming Language ရဲ့ primary code တွေဖြစ်တဲ့ Decimal number, Binary number, Hexadecimal numbers တွေရဲ့ အကြောင်း ပြီးတော့ လိုအပ်ရင် ဆွဲနော်ယူပြီး အသုံးပြုလို့ရတဲ့ package တွေထဲက object-function တွေရဲ့ အကြောင်းတွေရုပ်ကို ပို့ချမယ်သွားမှုပါ။ ဒါ သက်ဆုံးရဲ့ index တွေပြီးရင်တော့ chapter 1 ပြီးသွားပြီးတော့ အဆင့်မြင့် object တွေနဲ့ ဖန်တီးတဲ့ OOP concepts တွေပါဝင် တဲ့ chapter 2 ကို ပို့ချမယ်သော့မှာဖြစ်ပါတယ်။ ဒါတော့ အခြေခြား chapter 1 ဖြစ်တဲ့ Theories and Information of JAVA ဆိုတဲ့ အခန်းကို သေသေချာချာ ပြန်လုန် လေ့လာသားပါ။



လွယ်ပါတယ်။ ဘယ် drive နဲ့ boot လုပ်မယ်ဆိုတာ Bios setting မှာပြောမထားလို့ ဒီပြဿနာနဲ့ ရင်ဆိုင်နေရတာ။ ကွန်ပျူးတာတွေမှာ များသောအားဖြင့် စိုင်းရုံစွဲတော်က က်င်းဝေးအောင် ဖလေ့ရှိကနေ စပြီးမတက်ဖို့ အကာအကွယ်ပြထားလေ့ရှိတယ်။ ဖလေ့ရှိကနေစပြီး boot တက်ချင်ရင် ကွန်ပျူးတာ စဖွင့်ဖွင့်ချင်း CMOS Setup Utility ထဲရောက်အောင် ဝင်လိုက်ပါ။ (ဖွင့်ဖွင့်ချင်း delete key ကိုနှစ်ထားလိုက်ပါ)။ Advance BIOS Feature ကို ထပ်ပြီးဝင်လိုက်ပါ။ ပြီးရင် First Boot Device မှာ Floppy ကို ပေးထားလိုက်ရင်ရပြီ။ နောက်ပိုင်းမှာ ဖလေ့ရှိပါက သိပ်မသုံးကြတဲ့ စီမံတွေ့နဲ့ boot လုပ်လာကြတယ်။ CD-ROM ကနေ boot လုပ်ချင်ရင်လည်း Floppy အစား CDROM လို့ ပြောင်းပေးလိုက်တာနဲ့ရပြီ။

## JAVA Developer Guide

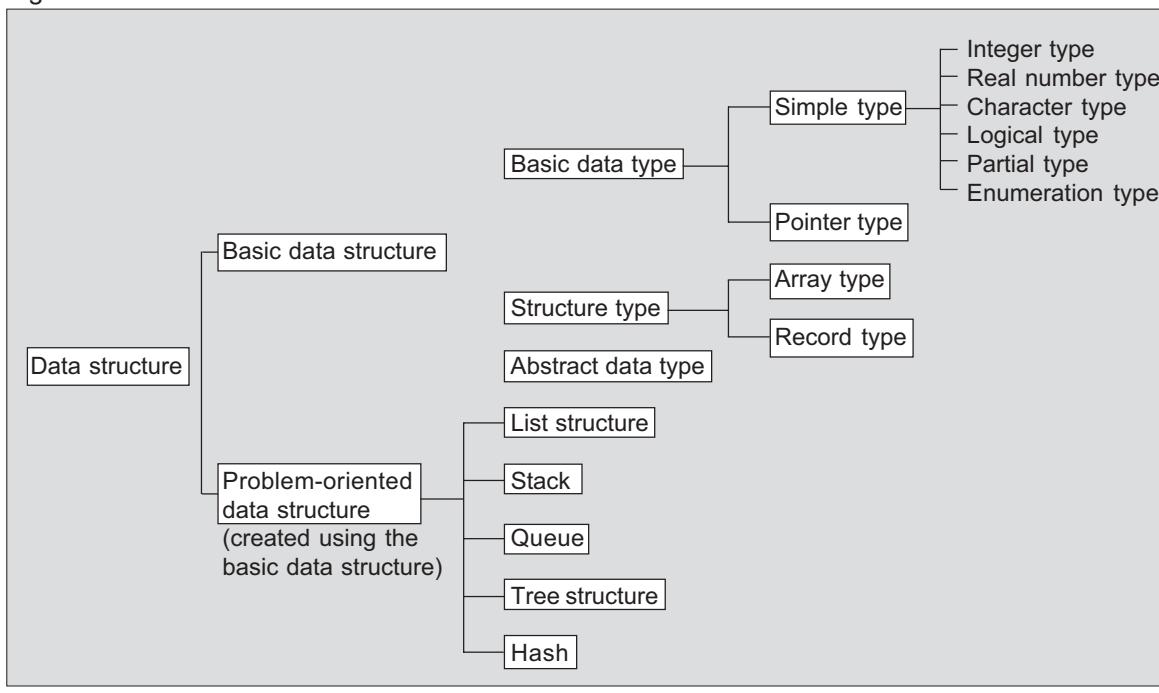
### 1.5 Arrays of JAVA

ကွန်တော်အနေနဲ့ JAVA Developer Guide ကို သင်ကြား  
ပို့ချေရာမှာ chapter 1 ကို Basic Foundation အနေနဲ့ အပြည့်စုံ  
ဆုံး၊ အလွယ်ကူဆုံးဖြစ်အောင် ရေးသားပို့ချေပေးခဲ့ပါတယ်။ JAVA  
ရဲ့ အခြေခံအတဲ့ဖြစ်တွေထဲမှာ Arrays ဟာ မပါမဖြစ်တဲ့ ပင်မ  
မဏ္ဍာင်အတ်မြစ်ကြီးတစ်ခုအဖြစ် ပါဝင်ပါတယ်။ ဒီတော့ အခြေခံက  
စံပြီးတော့ လေ့လာကြတောပေါ့။ JAVA ဟာ နောက်ဆုံးပေါ် Non-  
procedural Language, OOP Language တစ်ခုအဖြစ်ပါတယ်။ ဒီအခါမှာ အရင် Language တွေမှာ မပါဝင်တဲ့ features တွေ  
အများကြီးပါဝင်လာတယ်ဆုံးတာကိုလည်း ကွန်တော် ရှုံးသင်ခန်း  
စာတွေမှာ ပို့ချေပေးခဲ့ပြီ။ JAVA Language ကို အခြေခံပြီး  
တော့ Software Engineering အဆင့်ကို တက်လှုပ်မယ်ဆိုရင်  
ကွန်တော်တို့အနေနဲ့ System Development, Internal Design  
and Programming စတဲ့ အခန်းကဏ္ဍတွေကို မေ့ထားလို့  
မရပါဘူး။ တကယ်တတ်မြောက်အောင် သင်ယူရမှာပါ။ (Com-

puter Science နယ်ပယ်ကနဲ့ NCC Diploma တက်ရောက်နေ  
တဲ့ ကျောင်းသားတွေကတော့ ဒီ Module တွေကို မစိမ်းပါဘူး)  
ဒါတွေကို စနစ်တကျ မတတ်မြောက်ရင် programming မိုးပုံ  
အောင် ရေးတတ်နေလည်း နိုင်ငံတကာ Exam တွေကို ဝင်ရောက်  
ဖော်ဆိုနိုင်မှာ မဟုတ်ပါဘူး။

ကဲ ဒီတော့ Arrays နဲ့ ပတ်သက်ပြီးတော့ အခြေခံကစံပြီး  
တော့ သင်ယူကြရအောင်။ Arrays ဟာ အမှန်တကယ်ကတော့  
data structure အုပ်စုဝင်တစ်ခုပါဝါး။ Data Structure (Data  
type) ဟောနဲ့ ပတ်သက်ပြီးတော့ လက်နဲ့ချုတ်ကိုနိုင်အောင်အထိ  
1.2 Decision, control structure and looping statement  
in JAVA ဆိုတဲ့အခန်းမှာ သင်ကြားပို့ချေပေးခဲ့ပြီ။ အဲဒီမှာတော့  
အခြေခံယေးအနေနဲ့ပါဝါး။ ဒီနောက်တော့ Internal Design  
and Programming ဆိုတဲ့ Module ကို အခြေခံပြီးတော့ Arrays အကြောင်းကို ရှင်းပြုမှာပါ။ ကဲ ဒီတော့ Figure 1.2 Classification  
of data structures ကို လွှေလာကြည့်ပါ။

Figure 1-2 Classification of data structures



ကွန်ပျူတာသင်ခန်းစာ

က အပေါ်မှာရေးပြဿနာတဲ့ Data type ပုံလေးကို လေ့လာဖြို့  
ရင် Arrays ဟာ data structures အပိုဒဝိဖြစ်တယ်ဆိုတာကို  
တွေ့ရမှာပါ။ Data Structures ရဲအောက်မှာ Basic data structure  
တဲ့ problem-oriented data structure (created su-  
ing the basic data structure)ဆိုပြီး အပ်စုနှစ်စု ရှိတာကို  
တွေ့ရမှာပါ။ ဒီအခါ JAVA ရဲ Features တွေထဲမှာတော့ problem-oriented data structure တွေကို JAVA ရဲ package  
class တွေထကနဲ့ ဆွဲခေါ်ယူပြီးတော့ အသုံးပြုလိုရပါတယ်ဆိုရင်  
အခြား ဘယ် Language တွေမှာမူ မပါဝင်သေးတဲ့ features  
တွေ ပါဝင်ပါတယ်ဆိုတာကို လက်ခံနိုင်မှာပါ။ အမျိုးတကယ်က  
တော့ ဒီ problem-oriented data structure တွေဟာ int,  
float, Long, char, double စုတဲ့ Basic data structure တွေကို  
အခြေခံပြီးမှ ပြန်တည်ဆောက်ရတာပါ။ JAVA Language မှာ  
တော့ အချို့သော problem-oriented data structure တွေဟာ  
package တွေထဲမှာ ပါဝင်နေတော့ ဆွဲခေါ်ယူပြီး အသုံးပြုလိုက်ရှိ  
ပါပဲ။ အဲဒီလိုဂါပဲ Arrays ဟာလည်း data structure အပိုဒဝိ  
ဆိုပေမယ့် basic data structure တွေဖြစ်တဲ့ int, float, Long,  
char, double စုတဲ့ data type တွေကို အခြေခံပြီး ပြန်တည်  
ဆောက်ရတာပါ။ Arrays ဆိုတာဘာလ? က သေသေချာချာ  
ရေးပြီး ရေးသားပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

Arrays ဆိတာဟာ multiple data item တွေကို group လုပ်ပြီး အသံးပြုချင်တယ်ဆိုရင် array အနေနဲ့ ဖန်တီးရပါတယ်။ သူမှာ index\_number ရှုပါတယ်။ သူ အနေနဲ့ multiple data item တွေကို common\_name တစ်ခုပေးပါတယ်။ သူရဲ့နောက်မှာ size specification ပါဝင်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ array ကို ကြည်းအခါးမှာ array\_name ရဲ့နောက်မှာ square bracket [ ] ပါဝင်ရပါတယ်။ အဲဒီ square bracket [ ] ထဲမှာမူ non-negative integer ကို ထည့်ပြီး ကြည်းရပါတယ်။ ဒါကို စာဖတ်သူ အနေနဲ့ complex ဖြစ်သွားမှုစိုးလို့ sample အနေနဲ့ ရေးပြပါမယ်။

```
sample : int [] a=new int [5];
```

ဒီပုံစံရေးခြင်းဟာ one-dimensional array ကို အသုံးပြု၍  
ကြေညာထားတာပါပဲ။ ကိန်းသေရထားတဲ့တန်ဖိုးနဲ့ တွက်ချက်မယ်  
ဆိုရင်တော့ ဒီလို Declarer လုပ်ရမယ်ပါ။

```
sample : int [] a={ 22, 10, 18, 9, 27 };
```

ဒါ Array ကို index\_number တွေနဲ့ ညွှန်ပြရမယ်ဆိုရင်  
တော့ ဒီလိုတွေ ရမှာပါ။

```
a [0] = 22  
a [11] = 10  
a [2] = 18  
a [3] = 9  
a [4] = 27
```

Array ကို စီစဉ်တဲ့အခါမှာ index\_number ဟာ [0] ကနေ စပြီးတော့ စီစဉ်ပါတယ်။ အကွဲရာ char data type နဲ့ ကြညာ မယ်ဆိုရင်တော့ ဒီလိုပေးရမယ်ပါ။

```
String [] a={"Hello!","Kyaw Zayar Lay."};  
    a[0] = Hello!  
    a[1] = Kyaw Zayar Lay.
```

က ဒါနိရင်တော့ အခြေခံ array type ကို အခြေခံလောက်တော့ နားလည်သဘောပါက်သွားမှာပါ။ ဒီတော့ arrays မှာ one-dimensional array, Two-dimensional array, Multi-dimensional စသဖြင့် ဂဲပြားခြားနားတာတွေကို တစ်ပိုင်းစီ စနစ်တကျ ရှင်းပြပါမယ်။ သေသေချာချာ အချိန်ယူပြီးတော့လော်လာပါ။

## One-dimensional array

**Figure 1-3 One-dimensional array**

1st	2nd	3rd	.....	i th	.....
Element	Element	Element	.....	Element	.....
A(1)	A(2)	A(3)		A(4)	

က ဒါနိရင်တော့ one-dimensional array အကြောင်းကို  
အခြေခံတော့ ရလောက်ပါ၌။ သတိပြုရမှာကတော့ ကြညာထား  
တဲ့ data type ရဲ့ နောက်က variable name ရဲ့ နောက်က  
square bracket [ ] ထဲမှာ ဒါသမကိန်း (float, double) ကိန်းနဲ့  
(-) negative ကိန်းတွေ မထည့်မိမိပါဘူး။ Square bracket [ ]  
ထဲမှာ မပါမိမိပဲနော်။ Data type အနေနဲ့တော့ Declare လုပ်ပြီ။  
အသုံးပြနိုင်ရပါတယ်။ က အောက်မှာ sample ရေးပြပါမယ်။  
ရှင်းပြတာကို မှတ်စီမံချက်မှာထိုလိုပါ။

```
sample:float[] a={1.2,3.6,2.7};
```

```
a[0] = 1.2  
a[1] = 3.6  
a[2] = 2.7
```

Java ရဲ့ array ပုံစံဟာ C++ မှာပါဝင်တဲ့ array type နဲ့ ဆင်တူ

## ကွန်ပူတာသင်ခန်းဘ

သလိဂုရီပေမယ့် ကွဲပြားစြားနားပါတယ်။ လက်တွေ့အနေနဲ့ ချရောပြရရင်တော့

**sample :** int [] a=new int [5]; //Java Language

int a[5]; //C++ Language

ဒီအခါမှာ Java မှာ အသုံးပြုတဲ့ Array ဖုံးဟာ C++ မှာ အသုံးပြုတဲ့ pointer type နဲ့ သွားတူနေတာကို သတိပြုရမှာပါ။ Java မှာ pointer type ဖုံးဟာ သီးသန် features အနေနဲ့ မပါရှိပါဘူး။ သတိပြုပြီး သိထားရမယ့်အချက်တွေကတော့ Java မှာ poninter type ဖုန်တီးရေးသားခြင်းနဲ့ပတ်သက်တဲ့ method တွေကို ဖော်ရှားသားခြင်းပဲဖြစ်ပါတယ်။ Java ရဲ့ array type ဖုံးဟာ C++ က ဒီလိုပုံစံနဲ့ သွားတူနေပါတယ်။

**sample:** int\*a=new int[5]; //c++ pointer type use

ကဲ ဒါဆိုရင်တော့ Arrays နဲ့ပတ်သက်ပြီးတော့ သေသေ ချာချာ နားလည်းသောာပေါက်သွားလောက်ပါပြီ။ အလွယ်ဆုံး ပြောရရင်တော့ Array ဆိုတာဟာ အလုပ်တစ်ခုကို လုပ်ကိုင်ဖို့ အတွက် same data type ကို အသုံးပြုပြီး multiple data item တွေကို စိုင်ဆိုင်ပါတယ်။ ဒီအခါ ဒီ multiple data item တွေကို ကိုယ်စုံပြုတဲ့ Team\_name ကို စိုင်ဆိုင်တဲ့ work group ဟာ array ပါပဲ။

**Note :** Array ကို Declare လုပ်တဲ့အခါမှာ ကျွန်တော်တို့အနေနဲ့ C & C++ programming language မှာလိုမျိုး ကြော်လို လည်းရပါတယ်။ သတိပြုရမှာကတော့ Array ဟာ same data type ကို အသုံးပြုပါတယ်။ int data type ကို အသုံးပြုခဲ့ရင် အုပ်စုဝင် data item တွေ ဘယ်လောက်ပဲရှိရှိ အားလုံးဟာ int data type ပဲဖြစ်ပြီးတော့ တူညီပါတယ်။ အကယ်၍သာ multiple data type တွေကိုသာ အသုံးပြုမယ်ဆိုရင် array type မဟုတ်ဘဲ Record type ဖြစ်သွားတော့မှာပါ။ အောက်မှာ Record type ရဲ့ပုံစံကို ဥပမာအနေနဲ့ ရေးပြထားပါတယ်။ Array ဟာ same data type ဆိုတာကို အထူးသတိပြုပါ။

### One record (data on a student)

Student number	Name	Marks	Student number	Name	Marks	.....

↑  
Integer type  
Character type  
(character string type)  
Sort type

Figure 1-3.1 Record type

ကဲ Array ရဲ့ အခြေခံအနေနဲ့ Listing 1.16 ကို အသုံးပြုပြီး တော့ Array အကြောင်းကို လက်တွေ့လေ့လာတွေ့ဗြို့ရအောင်။

### Listing 1.16 Using one-dimensional array in JAVA

```

1. class BasicArray {
2.     public static void main (String arg[ ]){
3.         int [ ] a = {1,3,5,7,9};
4.         for (int i=0; i<5; i++){
5.             System.out.println (a[i]);
6.         }
7.     }
8. }
```

### Analysis

၁။ ဒီ program ဟာ Arrays ကို အခြေခံအနေနဲ့ ရေးပြထားတာ ပါ။ လိုင်းနံပါတ် ၁ မှာ Basic Arrays ဆိုတဲ့ class ကို ဖုန်တီးပါတယ်။ Primary class အနေနဲ့ပါ။ လိုင်းနံပါတ် ၂မှာ တော့ main ( ) function ကို ရေးသားထားပါတယ်။

၂။ လိုင်းနံပါတ် ၃မှာမှ int [ ] a ဆိုတဲ့ arrays တစ်ခုကို ဖန်တီးရေးသားပါတယ်။ [ ] တစ်ခုထပ်ပါတဲ့အတွက် one-dimensional array ဆိုတာကို သိထားရမယ်နော်။ သူရဲ့ member တွေကတော့ {1,3,5,7,9};ဖြစ်ပါတယ်။ Arrays member တွေကို ရေးတဲ့အခါ မှာ brace { } နှစ်ခုကြားမှာ comma (,) ခံပြီး ရေးရပါတယ်။ Last arrt index မှာတော့ comma (,) ခံရေးစရာ မလိုပါဘူး။ Brace { } ရဲ့နောက်မှာ semi-colon(;)ပါရပါမယ်။ ကဲ ဒါဟာ Arrays ရဲ့ basic\_form ပါပဲ။

၃။ ဒီ Arrays ရဲ့ member တွေကို one-dimensional array ရဲ့ form အတိုင်း စီစဉ်လိုက်မယ်ဆိုရင်တော့ ဒီလိုတွေ့ရမှာပါ။

```

a[0]=1
a[1]=3
a[2]=5
a[3]=7
a[4]=9
```

ကဲ ဒါဟာ dimensional array ဟာ ဒီအတိုင်းပဲ သွားမှာပါ။ ဒါပေမယ့် ခေါ်ယူအသုံးပြုတဲ့အခါမှာ a[ ] ဆိုတဲ့ Array name အတိုင်းခေါ်ယူရနဲ့ member အုပ်စုဝင်တွေအားလုံး ပါဝင်လာမှာပါ။ ၄။ လိုင်းနံပါတ် ၄ မှာ for looping ကို အသုံးပြုပြီးတော့ looping ပါဝင်ပါတယ်။ ဒီအခါမှာ int i ဆိုတာကို Declare လုပ်လိုက်ပါတယ်။ i ဟာ 5 ထက်တော့ ငယ်တယ်။ ပြီးတော့ i ကို postfix increment operator ကို အသုံးပြုပြီးတော့

## ဂွန်ပျိုတာသင်ခန်းစာ

တစ်ခုချင်းတိုးပြီး looping block ထဲကို ဝင်ပါတယ်။ လိုင်းနံပါတ် ဂုဏ်တော့ System.out.println() ဆိုတဲ့ print statement ကို အသုံးပြုပြီးတော့ output ထုတ်ပါတယ်။ ဒီအခါမှာ a[0] ဖြစ်စဉ်မှာ output 1, a[1] ဖြစ်စဉ်မှာ output 3, a[2] ဖြစ်စဉ်မှာ output 5, a[3] ဖြစ်စဉ်မှာ output 7, a[4] ဖြစ်စဉ်မှာ output 9 ဆိုပြီး output ကို ထုတ်ပေးပါတယ်။ i ရဲ့ တန်ဖိုးဟာ 5 နဲ့ ဆိုစဉ်မှာတော့ looping block ထဲကို ဝင်လိုက်ရတော့ပါဘူး။ i ဟာ 5 ထက်ငယ်မှာသာ လျှင် looping block ထဲကို ဝင်ခွင့်ရမှာဆိုတော့ for ရဲ့ block အပြင်ကို ခုန်စွာကိုပြီး လိုင်းနံပါတ် ၆ က main() function ရဲ့ close brace } ဆိုကို ဆင်းလာပါပြီ။ ပြီးတော့ Basic Arrays ဆိုတဲ့ primary class ရဲ့ close brace } ဆိုကို ဆင်းလာပြီးတော့ program ဟာ ပြီးသွားပါပြီ။

ကဲ ဒီတော့ ဒါ Listing 1.16 ဟာ အနည်းငယ် complex ပြစ်နေနိုင်ရင် Java က လွယ်ကူသွားအောင် သင့်ကို အကူအညီပေး မှာပါ။ JAVA ရဲ့ API (Application Programming Interface) method ဟာ package တွေတဲ့ ကန်လည်း လိုချင်တဲ့ method (or) object component တွေကို ဆွဲခေါ်ယူ အသုံးပြုတဲ့ နည်းပညာပါပဲ။

**Note :** JAVA package ဆိုတာကတော့ C & C++ programmer တွေအတွက် မစိမ်းလောက်ပါဘူး။ C & C++ မှာ သက်ဆိုင် ရာ program အလိုက် header file တွေကို declare လုပ်ပြီး လိုချင်တဲ့ object component တွေကို ဆွဲခေါ်ယူပြီး အသုံးချုတဲ့ နည်းပညာရည်ကို အသုံးပြုဖူး တွေ့ဖူးမှာပါ။ ဥပမာ #include <math.h> လို့ Declare လုပ်ခဲ့ရင် sqrt() square root method, Log() အစရိသဖို့ ကိုယ်တိုင် ဖန်တီးရေးသားစရာ မလိုဘဲ လိုချင်တဲ့ object component method ကို ဆွဲခေါ်ပြီး အသုံးပြုတော်ပါပဲ။ JAVA မှာပါဝင်တဲ့ package ဆိုတာဟာလည်း ဒီအတိုင်းပါပဲ။ C & C++ programming မှာ header file တွေကို ကိုယ်တိုင် ဖန်တီးရေးသားလို့ရာလို Java Language မှာလည်း package တွေလဲက object-component တွေကို ဆွဲခေါ်ယူပြီး အသုံးချိန်ရုံးသာမ်ား ကိုယ်တိုင်လည်း package တွေကို ဖန်တီးရေးသားလို့ရပါတယ်။ ကိုယ်ပိုင် package တွေ ဖန်တီးပြီးတော့ Java.sun.com ကို လှုပ်ပြီး ပေးပို့ Develop လုပ်လို့ ရပါတယ်။ အမှန်တကယ်သာ အသုံးဝင်တဲ့ features ဆိုရင်တော့ နောက်ထွက်လာမယ့် SDK version အသစ်တွေမှာ Tools အသစ်အနေနဲ့ ဖြည့်သွင်းပြီးတော့ ပါဝင်လာမှုပါ။ ဒီအခန်းမှာတော့ package တွေအကြောင်းကို နားလည်အောင် အသုံးဝင်တဲ့ API လေးတွေကို ဆွဲခေါ်ယူပြီး အသုံးချုပါမယ်။ နောက် chapter တွေမှာတော့ package တွေအကြောင်းကို အပြည့်စုံဆုံးဖြစ်အောင် ပို့ချေပေးမှာပါ။

က ဒီတော့ Listing 1.16 မှာ ရေးပြခဲ့တဲ့ program ကို Length API method ကို အသုံးပြုပြီးမှ ဖြန့်ရေးဖွံ့ဖြိုးမယ်။ Array တစ်ခုမှာပါဝင်တဲ့ data item အရေအတွက်ကို သိချင်ရင် ArrayName.Length method ကို အသုံးပြုလို့ရပါတယ်။ အတိအကျမသိရတဲ့ data item အရေအတွက်တွေကို intput အနေနဲ့ လက်ခံရမယ်ဆိုရင်လည်း အသုံးပြုလို့ရပါတယ်။ Sample အနေနဲ့ အရင်ရေးပြပါဦးမယ်။

### sample

```
int[ ] a={1,3,5,7,9};  
for(int i=0;a.length;i++){  
    System.out.println(a[i]);}
```

Listing 1.16 ကို ဖြန့်ရေးထားတဲ့ Listing 1.17 ကို လေ့လာကြည့်၍ ArrayName.Length method ကို အသုံးပြုပြီး ရေးသားထားပါတယ်။

### Listing 1.17 Using API to ArrayName.length method

```
1. class APIClass {  
2.     public static void main (String [ ] args){  
3.         int [ ] a={1,3,5,7,9};  
4.         for (int i=0; i < a.length; i++){  
5.             System.out.println (a[i]);}  
6.     }  
7. }
```

**output**

```
1  
3  
5  
7  
9
```

### Analysis

၁။ ဒါ program ကို compile လုပ်ပြီး Run လုပ်မယ်ဆိုရင် ရယ်တဲ့အဖြေဟာ Listing 1.16 က output နဲ့အတူတူပါပဲ။ လိုင်းနံပါတ် ၄ က for looping ပတ်စဉ်မှာ a.length ဆိုပြီး arrayName.Length method ကို ဆွဲခေါ်ယူ အသုံးချုလိုက်ပါတယ်။

၂။ ပြီးတော့ လိုင်းနံပါတ် ၅ မှာ print statement ကို အသုံးပြုပြီးတော့ output ကို ရိုက်ထုတ်ထိုက်တာပါ။ ဒါကို လေ့လာမယ်ဆိုရင် Java ရဲ့ API method တွေဟာ programming လုပ်ရာမှာ လွယ်ကူစေဖို့ အကူအညီပေးတယ်ဆိုတာကို နားလည်သွားမှာပါ။ Memory & time waste ဖြစ်ခြင်းမှ ကာကွယ်မှုပေးတယ်ဆိုရင်လည်း မမှားပါဘူး။

ဒါ arrayName.Length method ဟာ ကျော်တော့အနေနဲ့ JAVA Developer Guide ကို ပို့ချေတဲ့ ပထမဆုံးအနေနဲ့ စတင်ပို့ချေတဲ့ API method ပါပဲ။ က ဒီတော့ နောက် Listing

## ကွန်ပူးတာသင်နှင့်

တွေ့မှာ API တွေကို ထည့်သွင်းပြီးတော့ သင်ကြားစိုးချွားမှာဖြစ်ပါတယ်။ သေသေချာချာသတိပြုပြီးတော့ လေ့လာပါ။

ကဲ ဆက်လက်ပြီးတော့ string ကို use လုပ်ပြီးမှ text element တွေကို dimensional အနေနဲ့ သုံးကြည့်ရအောင်။ Listing 1.18 မှာ ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။

**Listing 1.18 Using in arrayName.Length method in JAVA API**

```
1. class lengthClass{  
2.     public static void main (String []  
3.         args){  
4.             String []Hello = new string [3];  
5.             Hello [0] ="JAVA Developer Guide";  
6.             Hello [1] ="Kyaw Zayar Lay .";  
7.             Hello [2] ='IT Engineering .';  
8.             for(int i=0; i<Hello.Length; i++){  
9.                 System.out.println(Hello [i]);  
10.            }  
output  
JAVA Developer Guide  
Kyaw Zayar Lay.  
IT Engineering.  
Analysis
```

၁။ ဒါ program လေးဟာ မခက်ပါဘူး။ Primary class နဲ့ main( ) function တွေကို ဖန်တီးခဲ့ပြီး string type ကို one-dimensional array အနေနဲ့ Hello လို့ Declare လုပ်ပါတယ်။ ဒီအခါနာ new ဆိတ် keyword အသစ်တစ်ခု ကို အသုံးပြထားတာကို တွေ့ရမှာပါ။ new keyword ဟာ memory မှာ allocates လုပ်နိုင်ဖို့အတွက် နေရာကို ဖန်တီးပေးတဲ့သေားပါပဲ။ ရှုံးသင်ခန်းစာတွေမှာ မရှင်းခဲ့တာက တော့ Listing နဲ့အတူပဲ ရှင်းပြမယ်လို့ ရည်ရွယ်ခဲ့လို့ပါပဲ။ new keyword ကို အသုံးပြုပြီး Array ကို Declare လုပ်ရင် memory မှာ Array အတွက် နေရာတွေကို ဖန်တီးပေးတယ် ဆိတ် သဘောကိုပဲ အဂ္ဂာယ်မှတ်ထားပါ။

၂။ ပြီးတော့ Array ရဲ့ index တွေအလိုက် string တွေကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် ၇ မှာမ for looping နဲ့ပတ်ပြီး output ကို တစ်ကြောင်းခြင်း ရိုက်ထုတ်ပါတယ်။ Hello.Length ဆိတ် .Length API method ကို အော်မျိုး အသုံးပြုထားတာကို တွေ့ရမှာပါ။ ဒီတော့ output အနေနဲ့ JAVA Developer Guide Kyaw Zayar Lay.IT Engineering. ဆိတ် string text တွေဟာ တစ်ကြောင်းစီမံပြုးပေါ်လာခဲ့တာပေါ့။ Arrays ဟာ နားလည်သဘောပေါက်သွားရင် လွယ်ကြပြီးတော့ အင်မတန် အသုံးဝင်ပါတယ်။ ဆက်လက်ပြီးတော့ one-dimensional array အကြောင်းကို နားလည်သဘောပေါက်သွားရင် Two-dimensional array နဲ့

Multi-dimensional array တွေအကြောင်းကို ဆက်လက် လေ့လာကြတာပါ။

### Two-dimensional array

A data structure in which data is lined up in both vertical and horizontal directions called a two-dimensional array ရဲ့ အမိုးကြည့်ရအောင်။ အဂ္ဂာယ်ဖို့ မြောရင်တော့ one-dimensional array ရဲ့ နောက်မှာ square bracket [ ] တစ်ခုပါပါတယ်။ အဲဒီတော့ နောက်ထပ် square bracket [ ] တစ်ခု ထပ်တိုးလိုက်ခြင်းဟာ Two-dimensional array ဖြစ်သွားပါတယ်။ Two-dimensional array ဟာ data တွေကို Table Form နဲ့ သိမ်းဆည်းနိုင်ပါတယ်။ Table Form ဆိုမှတော့ သိလောင်တဲ့အခါမှာ Row & Column (တန်းနဲ့တိုင် ပုံစံ သိလောင်မှာပါ) ဒါလေမယ့် သိလောင်မှုပုံစံဟာ Language တိုင်းမှာ မတူညီကြပါဘူး။ ဘာကြောင့်လဲဆိုတော့ Two-dimensional array ဆိုလေမယ့် main memory unit မှာ သိလောင်တဲ့ ပုံစံကတော့ one-dimensional array ရဲ့ Form အတိုင်းပဲ သိလောင်တာပါ။ ကဲ အောက်မှာရေးဆွဲပြထားတဲ့ Figure 1.4 ကို လေ့လာကြည့်ပါ။ Two-dimensional array ရဲ့ ပုံစံပါပဲ။

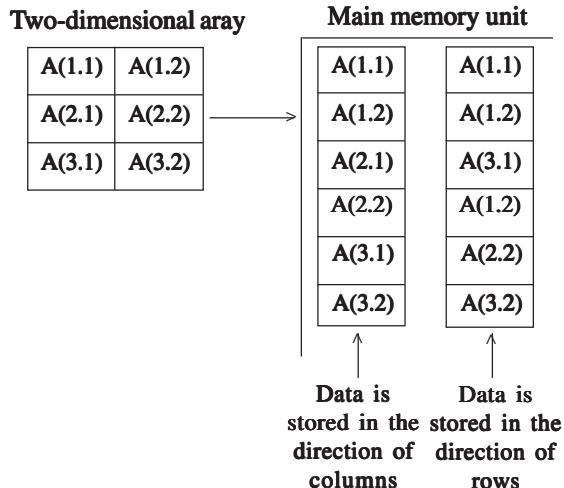
**Figure 1.4 Two-dimensional array (with three columns and two rows)**

		Row j
		↓
Column i →	A(1.1)	A(1.2)
	A(2.1)	A(2.2)
	A(3.1)	A(3.2)

ဒီအခါမှာ အချို့ Language တွေမှာကတော့ A[Column] [Row] လို့ မှတ်ယူပါတယ်။ ဒီတော့ ရှင်းအောင်ပြောရရင်တော့ A[Vertical] [Horizontal] ဆိုပြီးမှ မှတ်သားသိလောင်ပါတယ်။ ဒါလေမယ့် အချို့ Language တွေမှာတော့ A[Row] [Column] ဆိုပြီး သိမ်းဖို့ပါတယ်။ ဘာကြောင့်လဲဆိုတာကို လေ့လာကြည့်ရင် main memory unit မှာ sequentially store လုပ်တဲ့အခါမှာ direction of column type နဲ့ direction of Row type ဆိုပြီး ပုံစံကြေားသွားလို့ပါပဲ။ အဂ္ဂာယ်ဆုံး အရှင်းဆုံးဖြစ်အောင်ပြောရရင်တော့ C & C++, JAVA စဲတဲ့ Language တွေမှာတော့ vertically type store ဖြစ်ပြီး FORTRAN, COBOL အစရှိတဲ့ language တွေမှာတော့ Horizontally type store ဆိုပြီး သိလောင်ပါတယ်။ ပိုမိုးရှင်းသွားအောင်လို့ Figure 1.5 ကို သေသေချာလေ့လာကြည့်ပါ။ Main memory unit မှာ store လုပ်ပုံကို ရေးဆွဲပြထားတာပါ။

## ကွန်ပျူတာသင်ခန်းစာ

**Figure 1.5 How data of a two-dimensional array is stored in a main memory unit**



က ဒါဆိုရင်တော့ Two-dimensional array ရဲ့ memory unit မှာ store လုပ်ပါကို နားလည်သွားလောက်ပါ၍။ Two-dimensional array မဲ့ syntax\_form ကို ရေးပြပါ၍မယ်။ လော့လာကြည့်ပါ၍။

### sample

```
Declare int[ ][ ]a={           primary dimensional
using          {array data},   array of open and
                     {number item}, close brace
                     .....
Array name          };
semicolon
```

က လက်တွေအနေနဲ့ ရှင်းသွားအောင် Listing 1.19 နဲ့ Two-dimensional array အကြောင်းကို လက်တွေရေးပြပါ မယ်။ Table Form ဖို့ပါဝါပါ။ သိပ်မခက်လုပ်ပါဘူး။ လော့လာကြည့်ပါ။

### Listing 1.19 Demonstrate in Two-dimensional array stored data

```
1. class twoArray{
2. public static void main(String[]
args){
3. //Declare two-dimensional array
in 4 Row and 5 Column
4. int[] []a={
5.           {16,3,2,13,9},
6.           {5,10,11,8,7},
7.           {9,6,7,12,31},
8.           {4,15,14,1,8}
9.         };
```

```
10.    for(int i=0;i<4;i++){
11.      for(int j=0;j<5;j++)
12.        System.out.print(a[i][j]+" ");
13.        System.out.println();}
14.    }
15.  }
output
 16 3      2      13      9
 5 10     11      8      7
 9 6      7      12      31
 4 15     14      1      8
```

### Analysis

၁။ Listing 1.19 ၏ program ကို လော့လာကြည့်မယ်ဆိုရင် တော့ primary class နဲ့ main () function ကို ရေးဖို့ပြီး လိုင်းနံပါတ် ၃ မှာတော့ comment ကို ရေးသားထားတာကို တွေ့ရမှာပါ။ ထည့်ရေးရင်လည်း ရပါတယ်။ //Double backslash ခံပြီး ရေးထားတဲ့အတွက် JAVA Interpreter က အသိအမှတ်မပြုပါဘူး။

၂။ လိုင်းနံပါတ် ၄ မှာတော့ int[ ][ ]a ဆိုပြီး Two-dimensional array ကို Declare လုပ်ပါတယ်။ သတိပြုရမှာက တော့ JAVA မှာ Array ကို Declare လုပ်စဉ်မှာ int[ ][ ]a ဆိုပြီး square bracket ကို Array\_name ရဲ့ရှိမှာ ထားရ ပေါ်လုံး looping ပတ်ပြီး ခေါ်ယူ သုံးခွဲတဲ့အခါမှာတော့ Array\_name က အရင်လာဖြုံး square bracket ဟာ နောက်ကို ရောက်နေရပါမယ်။ Declare လုပ်စဉ်မှာ JAVA ရဲ့ array ဟာ တစ်မျိုးဖြစ်နေတယ်လို့ ထင်ရပေမယ့် ခေါ်ယူ သုံးခွဲအသုံးပြုပုံကတော့ C & C++ နဲ့ အတူတူပါပဲ။

Note : Array variable ကို Declare လုပ်တဲ့အခါမှာ int[] a; လို့ ကြညာလို့ရသလို C & C++ programming language မှာလို int a[ ]; လို့ ကြညာလို့လည်းရပါတယ်။ ကြိုးကြုံတဲ့ပုံစံနဲ့ Declare လုပ်လို့ရပါတယ်။

၃။ Two-dimensional array မဲ့ member data တွေကို (=) equal sign ရဲ့ ညာဘက်မှာ { }open and close brace ကြားမှာ သီးသန့် brace တွေခွဲပြီး (,)comma ခံပြီး ရေးပါ တယ်။ လိုင်းနံပါတ် ၅ က ကန် လိုင်းနံပါတ် ၆ အထိဟာ Two-dimensional array ကို Declare လုပ်ထားတာပါ။ လိုင်းနံပါတ် ၉ က အပိုတ် close brace ရဲ့နောက်မှာ semi-colon (;)ပါတာကို သတိပြုပါ။ Array မဲ့ Syntax\_form ကိုလည်း ရေးပြထားတော့ သတိပြုပြီးသားဖြစ်မှာပါ။

၄။ လိုင်းနံပါတ် ၄ မှာ Row အတွက် for looping ကို int i နဲ့ Declare လုပ်ပြီး ပတ်ပါတယ်။ For Looping အတွက် looping block အနေနဲ့ { brace တစ်ခု ဖွင့်ခဲ့တယ်ဆိုတာကို တွေ့ရမှာပါ။ သူ့ရဲ့အတွင်းထဲမှာမဲ့ for looping တစ်ခုကို ထပ်ပတ်ပါတယ်။ Column အတွက်ပါ။ Column အတွက်

## ကွန်ပူတာသင်ခန်းစာ

ကတော့ int j ကို Declare လုပ်ပါတယ်။ အထူးသတိပြုရမှာ ကတော့ looping ပတ်တဲ့ အခါမှာ looping တစ်ခန့်တစ်ခု ထားလို့ရပါတယ်။ ဒါပေမယ့် looping တွေဟာ cross ဖြတ်လို့ မရပါဘူး။ ဒီအချက်ကိုတော့ Listing 1.18 မှာ သာမက ဘယ် program တွေမှာမဆို သတိပြုပြီးတော့ ရေးသာပါ။

ထုံးစံအတိုင်းပါဝါ။ Package တွေထက် API method တွေကို ယူမသုံးတော့ output ရှိရတယ်ချင်တော့လည်း for looping နဲ့ပုံပြန်တိတ်ရမှာပါ။ အဲဒါကြောင့် လိုင်းနံပါတ် ၁၀ နဲ့ လိုင်းနံပါတ် ၁၁ မှာ i နဲ့ j အတွက် looping ပြန်ပတ် ထားတာပေါ့။ အပေါ်မှာ i ဟာ Row, j ဟာ column ဆိုပြီး ညီခြားသားဆိုတော့ လိုင်းနံပါတ် ၁၂ က output statement ကြောင့် အဖြော့က ကဏ္ဍနဲ့တစ်လုံးနဲ့ output ထွေတို့ တိုင်း Double quote " " နှစ်ခုကြားက blank အကွာအဝေး အတိုင်း အနည်းငယ်စီးပွားရေး ပေါ်ပါတယ်။ ပြီးတော့ အောက် တစ်ကြောင်းမှာ print command ကို new line print form နဲ့ ရေးထားတဲ့ အတွက် နောက်တစ်ကြောင်းမှာမ Row ကို ရှိက်ထုတ်တာပါ။

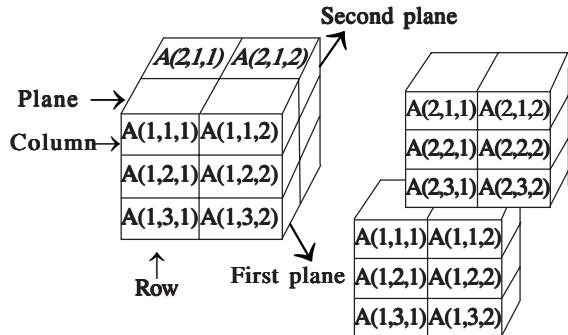
၅။ အလွယ်ပြောရရင်တော့ 1<sup>st</sup> Row အတွက် ပါဝင်တဲ့ column အရေအတွက်မပြည့်မချင်း block ထဲက ထွက်လို့ မရပါဘူး။ ပြီးသားမှ 2<sup>nd</sup> Row အတွက် column ကို လာရှိက်ထုတ်တာပါ။ ဒီလိုနည်းနဲ့ပုံ Table form အတိုင်း Four Row and Five column ပါတဲ့ ပေါ်တာတစ်ခုရလိုပေါ်တာပါ။ အသေးစိတ် သေသေချာချာဖော်ပြီးတော့ trace လှပြီး လေ့လာကြည့်ပါ။

Tow-dimensional array အကြောင်းကို နားလည် သော့ ပေါ်သွားရင် Multi-dimensional array အကြောင်းကို ဆက်လက်ပြီး လေ့လာကြပါနို့။

### Multi-dimensional array

Multi-dimensional arryas such as three-, four-, or n-dimensional arrays can be defined လို့လည်း အလွယ်ပြောရအောင်ပါ။ အတိအကျအဓိပ္ပာယ်ဖွံ့ဖို့တာတော့မဟုတ်ဘူး နော်။ အလွယ်ပြောမယ်ဆိုရင်တော့ Array\_name ရဲ့နောက်မှာ square bracket[ ] တွေ ထပ်တိုးလာတာပါပဲ။ ဥပုံ three-dimensional array မှာဆိုရင် comprising planes, columns and rows ဆိုပြီး ဗုံးတည်ဆောက်သလိုပေါ့။ Three-dimensional array ကို ထိန်းချုပ်တဲ့ အခါမှာတော့ two-dimensional ကို ထိန်းချုပ်ပုံနဲ့ အတူတူပါဝါ။ ကဲ ဒါဆိုရင်တော့ three-dimensional array ကင်း အောက်တဲ့ two-dimensional array form ကို ပြောင်းလဲရေးဖွံ့ထားတဲ့ Figure 1.6 ကို လေ့လာကြည့်ပါ။ အလွယ်တကူ ရှင်းလင်းသွားမှာပါ။

**Figure 1-6 Developing a three-dimensional array into a two-dimensional array**



One-dimensional array, two-dimensional array နဲ့ Multi-dimensional array စာတွေ တွေအကြောင်းကို အခြေခံကျကျ နားလည် သော့ပေါ်သွားရင် Array ရဲ့ အခိုက် ကျတဲ့ method အကြောင်းကို ရှင်းပြပါမယ်။ ကဲ ဒါဆိုရင် Arrays ရဲ့ programming area တွေမှာ လုပ်နိုင်စွမ်းတဲ့ method တွေကို classified လုပ်ကြည့်ရအောင်။ ဒီအခါမှာ - Static Array

- Dynamic Array စာတွေ မှစ်ခုကို တွေ့မြင်ရမှာပါ။ ကျွန်တော့အနေနဲ့ Java ရဲ့ စုစုပြု keyword တွေအကြောင်းကို မရှင်းပြရသေးပါဘူး။ ဒါကြောင့် keyword တွေအကြောင်းကို ရှင်းတဲ့ အခါမှာပဲ ယဉ်တွေပြီးယဉ်ပြပါတော့မယ်။ မပြည့်မစုရှင်းလင်းထားရင် အမှန်တကယ်လိုအပ်တဲ့ အချိန်မှာ သော့ပေါ် နားလည် မှာမဟုတ်တဲ့ အတွက် keyword အခန်းမှာ သေသေချာချာ ရှင်းပြပါမယ်။ ဆက်လက်ပြီးတော့ Arrays class ထဲက API method တွေကို ဆွဲခေါ်ပြီး အသုံးပြုကြည့်ရအောင်။ Listing 1.20 မှာ sort( ) method ကို ခေါ်သုံးထားပါတယ်။ လေ့လာကြည့်ပါ။

**Listing 1.20 Using Sort( ) method in Java API**

```

1. import java.util.Arrays; //import
2. import java.util.*;
3. class sortArray{
4.     public static void main(String[] args){
5.         int[] a={7,18,2,6,9,1,10};
6.         System.out.println("\t Before
7.         Sorting\n");
8.         for(int i=0;i<a.length;i++){
9.             System.out.print(a[i] + " ");
10.            } //print
11.            Arrays.sort(a); //sorting
12.            System.out.println("\t After
13.            Sorting \n");
14.            for(int i=0;i<a.length;i++){
15.                System.out.print(a[i] + " ");
16.            }
17.        }

```

## ကွန်ပူးတာသင်ခန်းဘ

```

13. }
output
    Before Sorting
7   18   2   6   9   1   10
    After Sorting
1   2   6   7   9   10   18

```

**Note :** ဒါ program မှာသုံးတဲ့ array.sort() method ဟာ Quick-Sort Algorithm method ကို အသုံးပြုထားပါတယ်။ ကျွန်တော့ အနေနဲ့ Sort method တွေအကြောင်းကို Sort Algorithm ဆိတဲ့ ခေါင်းစဉ်ကို flowchart, Algorithm နဲ့ ရှင်းပြုးလိုအပ် တဲ့နေရာမှာ Java program တွေနဲ့ ထည့်သွင်း ရှင်းပြုမှာပါ။ ဒီနေရာမှာတော့ Listing 1.20 မှာသုံးတဲ့ sort() method ဟာ QuickSort method ကို သုံးထားတယ်လိုပဲ မှတ်ထားပါ။

### Analysis

၁။ ဒါ Listing 1.20 မှာ Arrays class ထဲက sort method ကို ယူသုံးထားပါတယ်။ ဒီလို ခေါ်ယူအသုံးပြနိုင်ဖို့အတွက် လိုင်းနံပါတ် ၁ မှာ import java.util.Arrays; ဆုံးပြုး ရေးသားထားတာကို တွေ့ရမှာပါ။ Java ရဲ့ package ကို ခေါ်သုံးတာပါ။ Package ရဲ့ အစိမ္ပာယ်ကတော့ အရေ့မှာ ရှင်းပြန်သလိုပဲ C & C++ က header file တွေနဲ့ ဖုန်းတုပါတယ်။ Arrays class ကို ခေါ်သုံးချင်ရင် လိုင်းနံပါတ် ၁ မှာပဲ //Backslash ခံပြီး ရေးထားတဲ့ comment ပုံစံ ရေးထားတဲ့ import java.util.\*; လို့ ခေါ်သုံးရင်လည်း ရပါတယ်။ သူကတော့ Arrays class သာမကပဲ java.util.package တစ်ခုလုံးကို ခေါ်သုံးချင်ရင် (အခြား class တွေအပါအဝင်) ရေးသားအသုံးပြုပါတယ်။

၂။ လိုင်းနံပါတ် ၄ မှာ int[ ]a={7,18,2,6,9,1,10}; ဆုံးပြုး one-dimensional array ကို Declare လုပ်ပါတယ်။ လိုင်းနံပါတ် ၅ မှာတော့ System.out.println("\tBefore Sorting\n"); ဆုံးပြုး print ရိုက်ထုတ်ပါတယ်။ Escape character ဖြစ်တဲ့ (\t 0.5 ခြားရနဲ့) (\n new Line ဆင်းရနဲ့) ခေါ်သုံးထားတာကို လည်း တွေ့ရမှာပါ။ အဓိမ္ပာယ်ကတော့ Sorting မလုပ်ခဲ့ (ငယ်စဉ်ကြီးလိုက်မစဉ်ခဲ့) အရင် မူလ data အတိုင်း ရိုက်ထုတ်ပြုမယ်ပေါ့နော်။ ပြီးတော့ for looping နဲ့ array ကို ပတ်ပြီး ရိုက်ထုတ်ပါတယ်။ ဒီအခါမှာ ArrayName.length() API method ကို ခေါ်သုံးထားတာကို တွေ့ရမှာပါ။ For looping နဲ့ block brace ကိုလည်း တွေ့ရမှာပါ။ increment အနေနဲ့ (1 တိုးတိုင်း) block ထက် ပြန်ဝင်မှာပါ။

၃။ ဒီတော့ အရင်လို နောက်တစ်ကြောင်းမှာ Equals Sign(=) တို့ သုံးပြီး a[i]=i; လို့ ရေးစရာမလိုတော့ပါဘူး။ ကဲ ပြီးတော့ လိုင်းနံပါတ် ၇ မှာ Data တွေကို for looping နဲ့ ရိုက်ထုတ်ပါတယ်။ Data တစ်ခု ရိုက်ထုတ်ပြီးတိုင်း Double quote " "

နဲ့ blank (white) space ခြားပါတယ်။ အမှန်တကယ်က တော့ \t ကို အသုံးပြုပြီးတော့ခြားလည်းရပေမယ့် ဒီအချိန်မှာ complex ဖြစ်သွားမှာစိုးလို့ " " Double quote နဲ့ space ခြားလိုက်ပါတယ်။ ဒီတော့ output အနေနဲ့ Before Sorting ဆုံးပြုး လာပေါ်ပါတယ်။ ဒေါ်အောက်မှာမှ 7 18 2 6 9 1 10 ဆုံးပြုး sort မလုပ်ရသေးဘဲ လာပြုမှာပါ။

၄။ လိုင်းနံပါတ် ၈ မှာတော့ java.util.Arrays package ထဲက sort() API method ကို ယူသုံးပြုးတော့ ငယ်စဉ်ကြီးလိုက် စီစဉ်ပါတယ်။ Comment အနေနဲ့လည်း sorting လို့ ရေးပြထားတာကို တွေ့ရမှာပါ။ လိုင်းနံပါတ် ၉ ကတော့ print() command ကို အသုံးပြုပြီးတော့ After Sorting ဆိတဲ့ စာသားကိုလာပြဖို့အတွက် ရေးထားတာပါပဲ။

၅။ လိုင်းနံပါတ် ၁၀ ကတော့ sorting လုပ်ပြီးသား (ငယ်စဉ်ကြီးလိုက်) စီစဉ်ပြီးသား Arrays ရဲ့ member data တွေကို for looping ကို အသုံးပြုပြီးတော့ output ထုတ်ပါတယ်။ ဒီအခါ မာတော့ output ကို ထုတ်ပေးနိုင် brace { ထဲကို ဝင်ပြီး လိုင်းနံပါတ် ၁၁ မှာ ရေးထားတဲ့ print statement ကို အလုပ်သွားလုပ်ပါတယ်။ ဒီလိုနည်းနဲ့ output တွေကို ရိုက်ထုတ်သွားတာဟာ 1 2 6 7 9 10 18 ဆိတဲ့ sorting လုပ်ပြီးသား data တွေကို လာပြတာပေါ့။ Arrays ရဲ့ member data item တွေကြားမှာ white space ခြားစွဲအတွက် program ရေးစဉ် ထဲက Double quote " " နဲ့ ထည့်ပေါ်လို့ဆိုတာကို သတိပြုမိမှာပါ။

၆။ လိုင်းနံပါတ် ၁၁ က print statement ရဲ့အဆုံးမှာ for looping ရဲ့ block brace ကို လာပိတ်ထားတာကိုလည်း တွေ့ရမှာပါ။ လိုင်းနံပါတ် ၁၂ နဲ့ လိုင်းနံပါတ် ၁၃ ကတော့ main() function ရဲ့ close brace နဲ့ primary class ရဲ့ close brace } တွေဆုံးတာကို သတိပြုမိမှာပါ။

**Note :** ဒါ program မှာ အသစ်အဆန်းအနေနဲ့ကတော့ Java package တွေကို ခေါ်သုံးတာတွေကိုပါပဲ။ မှတ်မိသွားအောင် သေ သေချာချာလေ့လာပါ။ ရှေ့လျှောက် ဆက်လက်အသုံးပြုမယ့် tools တွေဖြစ်လိုပါပဲ။

ဆက်လက်ပြီးတော့ Java Language နဲ့ program တွေကို ရေးသားတည်ဆောက်တဲ့ အခါမှာ API method တွေကိုလည်း ခေါ်ယူအသုံးချင်ရင် တစ်ဖက်ကလည်း ကိုယ်ပိုင် function တွေကို ရေးသားပြီး အလုပ်လုပ်တွေ့ကြရအောင်။ Listing 1.21 က program ဟာဆိုရင် Java package တွေထဲက API method တွေကို ခေါ်ယူအသုံးပြုရှိသာမကပဲ ကိုယ်ပိုင် function တွေကို create လုပ်ထားပြီး အသုံးချထားပါတယ်။ (ရှေ့လ ဆက်လက်ဖော်ပြပါမည်)

ကျော်ဇူးတော်လေး (ကသာ)

## JAVA Developer Guide

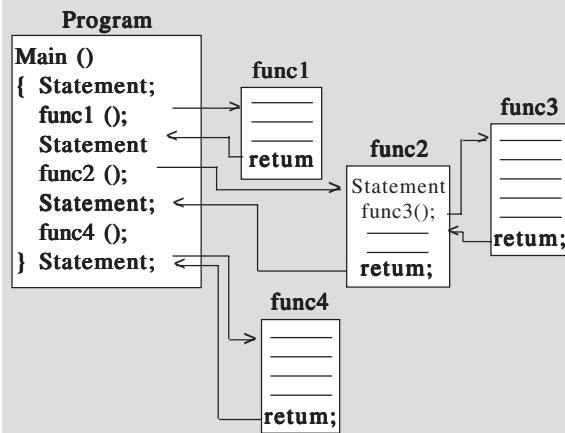
ဒီဇိုင်းရာမှာ functionအကြောင်းကို သိသန့် ပေါင်းစည်အနေ နဲ့ ရှင်းပြုစေယ်။

### Function Method In JAVA

Function( )ဆိုတာကတော့ Java Developer Guide ကို လေ့လာနေသူအနေနဲ့ မစိမ်းပါဘူး။ ဘာလို့ပြောရတာလဲဆုံး တော့ Java ရဲ့ အခြေခံ Listing 1.1 ကနေ စတင်ရေးသားပို့ချေပေး ခဲ့တဲ့ program တွေမှာ ဝါဝင်တဲ့ main( )ဆိုတာဟာ Java ရဲ့ function တစ်ခုဖြစ်လိုပါပဲ။ C & C++ မှာလည်း အတူတူပါပဲ။ အခု ရေးသားပို့ချေမပဲ၏ function() တွေရဲ့ပုံစံနဲ့ main( )ပုံစံဟာ ဆင်တူပါတယ်။ အခုရေးမယ့် function တွေနဲ့ main( ) ဟာ ထပ်တူထပ်ဖူးတူလားဆုံးရင်တော့ စမ်းရည်အနေနဲ့ကတော့ အများကြီး ကွဲပြားမြားနားပါတယ်။ အရေးအကြီးဆုံးအချက်ကို ပြောရရင်တော့ main( ) ဟာ ကိုယ်ပိုင်ရပ်တည်မှုမရှိပါဘူး။ Function( ) တွေဟာ ကိုယ်ပိုင်ရပ်တည်မှုမရှိပါဘူး။ main( ) ဟာ မိန့်မှုက်င်းမယ်ရုံးသာမကပါဘူး။ main( ) ဆိုတဲ့ method မပါလာရင် အဲဒီ program ဟာ compile လုပ်နိုင်စွာတောင် မရှိတော့ပါဘူး။ ဤတော့ function တွေ ဘယ်လောက်ပဲရေးထား ပေမယ့် program compile လုပ်ပြီး Run ပြီဆိုတာနဲ့ main( ) method ကို တွေ့အောင်ရှာပြီး အဲဒီဇိုင်းရာက အပြုံးပြုခြင်းတော့မှ compile လုပ်ပါတယ်။ ဒီတော့ main( ) ဆိုတာဟာ program တစ်ခုရဲ့ ဝင်ပါက်ဖြစ်နေတော့ပေါ့။ ဥပမာအနေနဲ့ပြောရရင်တော့ main( ) function ဟာ အိမ်တစ်အိမ်ရဲ့ဝင်ပေါက် (တံ့ခါးမကြီး)နဲ့ တူဖော်တယ်။ Function( ) ဆိုတာကတော့ အိမ်ထဲမှာရှိတဲ့ အခန်းငယ် တွေ (အိပ်ခန်း/ထမင်းစားခန်း)နဲ့တူပါတယ်။ အိမ်ထဲရောက်နေဖေ မယ့် အိပ်ခန်းထဲ/ထမင်းစားခန်းထဲ ဝင်ချင်မှုဝင်မှာပေါ့။ ဒီသော့ တရားဟာ function( )တွေ ရေးခဲ့ပြီးတော့ အော်သုံးချင်မှ သုံးမယ်။ မသုံးချင်လည်းရပါတယ်ဆိုတဲ့ သေားတရားပဲပေါ့။ က ဒါဆိုရင် main( ) function ကတော့ အိမ်တံ့ခါးမကြီးနဲ့ တူပါတယ်ဆိုမှ တော့ တံ့ခါးမကြီးကို ဖွင့်မဝင်ဘဲ အိမ်ထဲကို ဝင်လိုက် မရတော့ပါဘူး။ အိမ်ထဲကို ဝင်လိုမရတော့ဘူးဆိုမှတော့ function( ) တွေ ဖြစ်တဲ့ အိမ်ခန်း/ထမင်းစားခန်းထဲကိုရော ဝင်လိုရနိုင်ပါဦးမလား? မရနိုင်ပါဘူး။

က ဒါဆိုရင်တော့ main( ) function နဲ့ ကိုယ်ပိုင်ဖန်တီးရေးသားတဲ့ function( ) method တွေ ကွဲပြားမြားနားတာကို သော့ ပေါက်သွားတော့မှာပါ။ C & C++ နဲ့ ကွဲပြားမြားနားတာကတော့ Java မှာ return type ကို ထည့်သွင်းရေးဖွံ့စရာမလိုပါဘူး။ က Figure 1.7 ကို လေ့လာကြည့်ပါ။ Function( ) တွေ အလုပ်လုပ်တဲ့ပုံစံကို Diagram ဆွဲပြထားတာပါ။

**Figure 1.7 When a program calls a function, execution switches to the function and then resumes at the line after the function call.**



ဒါ Figure 1.7 မှာ ရေးပြထားတဲ့အဓိပ္ပာယ်ကတော့ program ဟာ run လုပ်ရင် main( ) ဆိုတဲ့ ဝင်ပါက်က စစ်ရှိပြီးတော့ ခေါ်ယူသုံးစွဲခဲ့ရတဲ့ function( ) တွေဟာ လုပ်ဆောင်ချက် ပြီး မြောက်တာနဲ့ return အနေနဲ့ main( ) ထဲကို ပြန်သွားရပါတယ်။ Function( ) တွေ ဘယ်လိုပဲ ဆင့်ပွားသုံးစွဲပေမယ့် main( ) ထဲကို ပြန်ကိုဝင်လာရမှာပါ။ က ဒါဆိုရင်တော့ function( ) နဲ့ ပတ်သက်ပြီးတော့ အကြမ်းအားဖြင့် နားလည်သေားပေါက်သွားရင် java.util.array package ထဲက Arrays class ကို ခေါ်ယူ အသုံးချဖို့ method မှာ function ငယ်လေးတွေ ရေးသားဖြည့် စွက်ပြီးတော့ စိတ်ကြွောက်ခေါ်ယူအသုံးချတဲ့ပုံစံတို့ Listing 1.21 မှာ ရေးသားပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```

Listing 1.21 Duplication on Array and testing writes the function method
1. import java.util.Arrays;
2. public class duplicate{
3. public static void main(String[] args){
4. int []a={22,14,6,8,44,20};
5. System.out.print("Array a=");
6. print(a); //function call
7. int []b=(int [])a.clone(); //Duplicate a[] in b[]
8. System.out.print("Cloned Array b=");
9. print(b);
10.System.out.println("Arrays.equals
(a,b) = "+Arrays.equals(a,b));
11.int i=Arrays.binarySearch (a,44);
12.System.out.println("Found at position 44 is "+i);
13.Arrays.sort(a);
14.System.out.print("Sorted Array a[]={"); print(a);
15.
16.String []c={"AB","CD","EF"}; //arrays contain three string elements
17.print(c);
18.String []d=(String [])c.clone(); //duplicate c[] in d[]
19.print(d);
20.
21.c[1]="XYZ"; //Change String []c of index 1,CD into XYZ
22.print(c);
23.print(d);
24. } //close brace to main() function
25.
26. public static void print(intp []a){
27. for(int i=0;i<a.length;i++){
28. System.out.print(a[i]+" ");}
29. }
30.
31. public static void print(object []
a){
32. for(int i=0;i<a.length;i++){
33. System.out.print(a[i]+" ");}
34. }
35. } //close brace to primary class output
Array a=22 14 6 8 44 20
Cloned Array b=22 14 6 8 44 20
Arrays.equals(a,b)=true
Found at position 44 is = 4
Sorted Array a[]={6 8 14 20 22 44}

```

AB CD EF

AB CD EF

AB XYZ EF

AB CD EF

**Note:** ဒါ prograrm မှာ function ရေးရတဲ့အကြောင်းရင်းကတော့ Arrays ရဲ့ data member တွေကို output ထုတ်ပဲအပါ မှာ for looping နဲ့ပတ်ပြီး ထုတ်ပါတယ်။ ဒါတော့ Arrays အမျိုးအစားများရင် for looping တွေ အများကြီးပတ်သွေလို main( ) function ဖြစ်စဲ ပင်မ body ကြီးထဲမှာ program code တွေ ရုပ်တွေးသွားနိုင်တဲ့အတွက် print( ) ဆိုတဲ့ function ကို ရေးသားခဲ့ရင်းပဲဖြစ်ပါတယ်။ ဒါအခါမှာ looping ကို ထပ်ကြံမ်တလဲ ပတ်စရာမလိုတော့ဘဲနဲ့ Arrays မှန်သမျှရဲ့ data number တွေကို ရိုက်ထုတ်တော့မယ်ဆိုရင် print() function ကို ခေါ်ယူလိုက်ရုပ်ပေါ်ပေါ်။

#### Analysis

၁။ Listing 1.21 မှာ ရေးသားရင်းလင်းပြထားတဲ့ program ဟာ Java Language ရဲ့ အဆင့်ဖြင့် package တွေထဲက object တွေကို ခေါ်ယူ သုံးစွဲရုံသာမကပဲ function method ကိုလည်း Introduce လုပ်ပြီး ရေးသားထားပါတယ်။ က စတင်ပြီး လေ့လာကြည့်ရအောင်။ လိုင်းနံပါတ် ၁ မှာ ရေးထားတာကတော့ java.util.Arrays package ကို ခေါ်ယူ အသုံးပြထားတာပါပဲ။ လိုင်းနံပါတ် ၂မှာမှ Duplicate class ဆိုတဲ့ primary class ကို ရေ့ဖွံ့ဖြိုးတော့ open brace { ကို ဖွင့်ထားတာကို တွေ့ရှုမှာပါ။ ဒီဇန်ရာမှာ ကျွန်ုင်းတော်ပြောချင်တာ ကတော့ ကျွန်ုင်းတော်အနေနဲ့ main( ) function ကို စတင်မရှင်းလင်းသေးဘဲ လိုင်းနံပါတ် ၂၆ ကနေ လိုင်းနံပါတ် ၃၄ အထိ ရေးသားထားတဲ့ print ဆိုတဲ့ function လေးကိုပဲ အရောင်ရှင်းပြသွားမှာပါ။ ဒါကို ရုပ်တွေးမသွားစေချင်ပါဘူး။

၂။ လိုင်းနံပါတ် ၂၆ မှာ ရေးထားတာကတော့ public static void print (intp []a) ဆိုတဲ့ function လေးကို တွေ့ရှုမှာပါ။ ဒီအခါမှာ ရှုံးယူအသုံးပြထားတဲ့ public ဆိုတဲ့ static keyword လေးကို သတိပြုမှာပါ။ ဒီဇန်ရာမှာတော့ Java Language မှာ static keyword နဲ့ final keyword ဆိုပြီးတော့ နှစ်မျိုး ကဲပြားမြားနားပါတယ်။ အဲဒါတွေကိုတော့ function တွေ ဖန်တီးရေးသားရင် လိုက်နာရမယ့် Rules တွေနဲ့ အတွေ့ဌးမှပဲ သေသေချာချာ ရှင်းပြပါတော့မယ်။ ဒါတော့ print( ) ဆိုတဲ့ function လေးကို ရေ့ဖွံ့ဖြိုးသွားတဲ့အခါမှာ အဖွင့် brace { လေးကို သတိပြုမှာပါ။ အရေးကြီးတဲ့ အချက်တစ်ချက်ကတော့ ဒါ print() function ထဲမှာ int[] a ဆိုတဲ့ arguments ကြောင့် ဒါ function ကို call ခေါ်ပြီး int, string အစရှိတဲ့ arrays တွေကို ထည့်လိုက်ရင် တွက်ချက်ပြီး

တဲ့အခါမှာ pass by valueလုပ်ဖြီးတော့ အဖြေကို main()  
ထကို ပြန်ပို့ပေးမှာပါ။

၃။ လိုင်းနံပါတ် ၂၂ မှာတော့ function ထဲကို ထည့်သွင်းခံရတဲ့ arrays တွေကို print ထုတ်ပေးဖို့အတွက် for looping နဲ့ အရှင်ပတ်ရုပါတယ်။ ဒီလို့ looping ပတ်ပြီးမှာသာ output အနေနဲ့ print ထုတ်လို့ရမှာဖော်။ သူတို့ကို ရှိရှိး၍ print command အနေနဲ့ output ထုတ်လို့ ရုသလားဆိုရင် မရပါဘူး။ သူတို့ဟာ arrays တွေမျိုးပါပဲ။ ရှိရှိး၍ arrays မဟုတ်တဲ့ကိန်း ဆိုရင်တော့ ရမှာဖော့။ ဒီအခါ looping ပတ်ရာမှာ for looping ရဲ့ test expressing နေရမှာ ကိန်းကဏ္ဍးကန်းသတ် ချက်ကို မထည့်သွင်းဘဲနဲ့ i < a.length; ဆိုပြီး ရေးသားထားပါတယ်။ Looping ပတ်ပြီးတိုင်း looping brace ထဲကိုဝင်ပြီး တော့ arrays ရဲ့ data items တွေကို ရိုက်ထုတ်ပါတယ်။ ဒီအခါ data items တွေ တစ်ခုနဲ့တစ်ခုကြားမှာ white space ခြားအောင်လို့ Double quote " နှစ်ခုကြားမှာ blank space ကို ထည့်ရေးထားပါတယ်။ ဒါကြောင့် output ထွက်လာတဲ့အခါမှာ data items တွေ တစ်ခုနဲ့တစ်ခုကြားမှာ Double quote ခြားခဲ့သလောက် အကွာအဝေးခြားနားထားပေါ့။ လိုင်းနံပါတ် ၂၃ မှာတော့ for looping အတွက် close brace } နဲ့ လိုင်းနံပါတ် ၂၄ မှာတော့ print( ) function အတွက် close brace } ကို လာပိတ်ထားပါတယ်။

တွက်ထုတ်ရမှာပါ။ ဒါကြောင့် ကျွန်တော်တို့ ဖန်တီးရေးသား ခဲ့တဲ့ function ထဲမှာ looping ပတ်ပြီး data member items တွေကို ရိုက်ထုတ်ဖို့ သီးသန့်စိစဉ်ရေးသား ခဲ့တဲ့ function ထဲမှာ looping ပတ်ပြီး data member items တွေကို ရိုက်ထုတ်ဖို့ သီးသန့် စိစဉ်ရေးသား ခဲ့တဲ့ ပါ။ ဒါကြောင့် လိုင်း နံပါတ် ၇ မှာ ရေးထားတဲ့ call function method ကြောင့် Array a=22 14 6 8 44 20 ဆိုပြီး output လာပြတာပါ။ ကြားထဲမှာ white space ခြားထားတာကတော့ print() function ကို ရေးသားထားတဲ့ double quote " ခြားပြီး ရေးသား ခဲ့လိုပါပဲ။ ကဲ ဒီအထိရှုပြုနော်။

၆။ လိုင်းနံပါတ် ၇ မှာတော့ Array class ထဲက class() method ကို အသုံးပြုပြီးတော့ int[ ]b ကို ဖန်တီးပါတယ်။ ပြီးတော့ cloned Array b=ဆိုတာကိုတော့ လိုင်းနံပါတ် ၈ မှာ System.out.print( ) method ကို အသုံးပြုပြီးတော့ output ထုတ်ပါတယ်။ Array b ရဲ့ data member တွေကို တော့ ကျွန်တော်တို့ ဖန်တီး ခဲ့တဲ့ print() function ကို ခေါ်သုံးပြီး မှ output ထုတ်ပါတယ်။ ဒီတော့ cloned Array b=22 14 6 8 44 20 ဆိုပြီး output လာပြပြန်ပါပြီ။ လိုင်းနံပါတ် ၁၀ မှာတော့ Array နှစ်ခုဟာ ညီမညီကို java.util.Arrays package ထဲက equals() method ကို ခေါ်သုံးပြီး output ထုတ်ပါတယ်။ အခြေကို လာပြပါတယ်။ လိုင်းနံပါတ် ၁၁ မှာတော့ int i ကို Declare လုပ်ပါတယ်။ ပြီးတော့ binarysearch() method ကို ခေါ်ယူပြီးတော့ Array a[ ] ရဲ့ data member တွေထဲမှာ 44 ဆိုတဲ့ ကိန်းဟာ Arrays ရဲ့ ဘယ် index နေရမှာ ရှိသူလဲ လို့ တွက်နိုင်းပြီးတော့ လိုင်းနံပါတ် ၁၂ မှာမှ System.out.println() method ကို အသုံးပြုပြီးတော့ found at position=4 လို့ လာပြပါတယ်။ Arrays ရဲ့ index တွေကို ပြတဲ့ အခါမှာ a[0] ကစာတယ်ဆိုတာကို မှတ်မိနေရမှာပါ။ 44 ဘာ index[4] ခုမြောက်မှာ ရှိနေလိုပါပဲ။

၇။ လိုင်းနံပါတ် ၁၃ မှာတော့ Sort() method ကို အသုံးပြုပြီး တော့ array a ကို sorting လုပ်ပါတယ်။ ပြီးတော့ လိုင်း နံပါတ် ၁၄ မှာတော့ print command ကို အသုံးပြုပြီးတော့ sorted Array a[ ]= ဆိုတာကို အရင် ရိုက်ထုတ်ပါတယ်။ ပြီးတော့မှ data item တွေကို ကျွန်တော်တို့ ဖန်တီးရေးသား ခဲ့တဲ့ print() function ကို ခေါ်ယူပြီးတော့ output ထုတ်ပေးပါတယ်။ လိုင်းနံပါတ် ၁၅ ကိုတော့ program code ကို ရှင်းလင်းအောင် line space ချုန်ခဲ့တာပါ။ Line Spacing ကို မချုန်လည်းရပါတယ်။ ပြီးတော့ လိုင်းနံပါတ် ၁၆ မှာတော့ String[ ]c={"AB","CD","EF"}; ဆိုပြီး string elements သုံးခုကို Declare လုပ်ပါတယ်။ လိုင်းနံပါတ် ၁၇

မှာတော့ Declare လုပ်ခဲ့တဲ့ arrays တွေကို output ထုတ်ပေးပါတယ်။ ဒီအခါမှာ output အနေနဲ့ AB CD EF လို့ လာပြတာကို တွေ့ရမှာပါ။ လိုင်းနံပါတ် ၁၈ မှာမှ string[ ]c ကို cloned() method နဲ့ string[ ]d ဆိုပြီး duplicate လုပ်ယူပြန်ပါတယ်။ လိုင်းနံပါတ် ၁၉ မှာမှ output ထုတ်ပေးပါတယ်။ ဒီကြောင့် AB CD EF ဆိုပြီး ထင်လာပြပြန်ပါတယ်။ ၈။ လိုင်းနံပါတ် ၂၂ မှာတော့ c[ ] array ထဲက data တွေကို changes လုပ်ပါတယ်။ အလွယ်ပြောရရင်တော့ c[ ] array ရဲ့ ဒုတယ်ပြောက်/ index အနေနဲ့ ဆိုရင်တော့ c[1] ဖြစ်တဲ့ data item ကို change လုပ်ဖို့အတွက် c[1]="XYZ"; ဆိုပြီး တော့ changes declare လုပ်ပါတယ်။ ဒါကြောင့် လိုင်း နံပါတ် ၂၂ မှာ print() function ခေါ်ယူပြီး print ထုတ်ပဲ့အခါမှာ AB XYZ EF ဆိုပြီး လာပြတာဖြစ်ပါတယ်။ လိုင်း နံပါတ် ၂၃ မှာ print(d) ဆိုပြီးမှ string[ ]d ကို output ထုတ်ပါတယ်။ ဒီတော့ output အနေနဲ့ AB CD EF ဆိုတာကိုပဲ လာပြတာပေါ်ပါ။ String[ ]c ကိုသာ data changes လုပ်လိုက်တာပါ။ String[ ]c ကို cloned လုပ်ယူးယူထားတဲ့ string[ ]d ကိုက changes လုပ်တာမှုမဟုတ်ပဲ။ ဒီတော့ မူလယွှေးယူစွဲက ရထားတဲ့ data အတိုင်းပဲ လာပြတာပေါ့။ ဒါကြောင့် output တွေဟာ ကဲ့ပြား ခြားနားတာပေါ့။ ကဲ့ပြား ခြားနားမှန်းသိအောင်လည်း output ထုတ်ပြတာပါ။ Main() function ရဲ့ close brace ကိုတော့ လိုင်းနံပါတ် ၂၄ မှာ လာပိတ်ထားပါတယ်။ Duplicate class ဆိုတဲ့ primary class ရဲ့ close brace ကိုတော့ လိုင်းနံပါတ် ၃၆ မှာ လာပိတ်ထားပါတယ်။ Program ဟာ ပြီးသွားပါပြီ။ ကဲ ဒါဆိုရင်တော့ function() တွေကို ဖန်တီး ရေးသားတဲ့ ဒါကြောင့်လည်းကောင်းလို့လည်း နားလည်း သဘောပေါက်သွားတော့မယ်လို့ ယုံကြည်ပါတယ်။

**Note:** ဒီ Listing 1.21 မှာ function အကြောင်းကို စနစ်တကျနားလည်းအောင် ရေးသားရှင်းပြခဲ့သလို Java ရဲ့ package class တွေထဲက API method တွေကို လိုအပ်ရင်လိုအပ်သလို ခေါ်ယူသုံးစွဲနိုင်အောင်လိုပေါ့နော်။ ဒီတော့ အောက်မှာ API method တွေ အကုန်တော့ မဟုတ်သေးပါဘူး။ လိုအပ်မယ်ထင်တဲ့ method တွေပါပဲ။ ဒီ object class တွေကို ရေးသားတင်ပြတဲ့အခါမှာ ဖြန့်မာလို ရေးသားရှင်းပြခြင်းမရှိပဲ အချို့နေရတွေမှာ Java Guide Book တွေမှာပါတဲ့အတိုင်း English လို ရေးသားထည့်သွေးပေးထားပါတယ်။ ဒါမှာသာ နောက်ပိုင်း ပိုမြီးဖတ်ရလွယ်ကူလာရမှာပါ။ လေ့လာကြည့်ပါ။

JAVA Language နဲ့ ပတ်သက်ပြီး အလွန်လျော့ပြောမှားတဲ့ စကားပုံတစ်ခုရှိပါတယ်။ အဲဒါကတော့ စာဖတ်သူကြားမူးမှာပါ။

"Write Once, Run Anywhere" အမို့ယ်ကတော့ တစ်ကြိုင်ရေး နေဂာတိုင်းသုံးဆိုတဲ့ အမို့ယ်ပါပဲ။ တချို့သူတွေကတော့ ပြောလေ့ရှိပါတယ်။ ဒါဟာ Java Applet တွေနဲ့သာဆိုပါတယ်လို့ ပြောကြပါတယ်။ အမှန်တကယ်ကတော့ Java Applet တွေသာ ကော Java Applet တွေ Java Application မှ အစပြောသာ Java နဲ့ ဆက်နွှယ်နေသော source code အားလုံးပါ အကျိုးဝင်ပါတယ်။ ကဲ စတင်လေ့လာကြပါနို့။ ဒါဟာ Java ရဲ့ စွမ်းရည် တစ်ခုပါလို့ ပြောရင်လည်း မမှားပါဘူး။

### **Introduction to JAVA API (Application Programming Interface)**

JAVA Language ကို လေ့လာတဲ့အခါမှာ အထောက်အကျ ရေစွဲအတွက် အခြေခံအဆင့်မှာ အသုံးဝင်လောက်မယ်ထင်တဲ့ object class တွေထဲက API method တွေကို ရေးသားပို့ချေပေးထားပါတယ်။ သတိပြုရမှာကတော့ အခု ကျွန်တော်တင်ပွဲတဲ့ API method ဟာ Introduction လုပ်တယ်ဆိုတာကို မမွေ့ဖို့ပါပဲ။ ကဲ စတင်လေ့လာကြရအောင်။

### **The Maths Class (java.util.Math Class)**

```
public static double      abs(double x) //absolute value
public static native double aton(double x) //asciiongent
public static native double ceil(double x) //ceiling
public static native double cos(double x)
public static native double sin(double x)
public static native double tan(double x)
public static native double sqrt(double x)//square root
public static native double log(double x) //base e
public static native double exp(double x) //base e
public static native double floor(double x)
public static native double max(double x, double y)
public static native double min(double x, double y)
public static native double pow(double x, double y)
public static long        round(double x)
public static synchronized double random()
```

### **Convert Between Degrees and Radians**

```
public static double toDegrees(double radians)
//converts the specified number of radians to degrees
public static double toRadians (double degrees)
//converts the specified number of degrees to radians
```

### **Methods that return Random Numbers using Random Class**

```
nextBoolean()    //return a random boolean value
```

nextDouble()	//return random double or float values between 0.0 and 1.0
nextFloat()	//return random double or float values between 0.0 and 1.0
nextInt()	//return random int or long value (all possible value)
nextLong()	//return random int or long value (all possible value)
nextGaussian()	//return a double value between -1.0 and 1.0

### **The Array Class (java.util.Arrays class)**

```
public static int      binarySearch()
public static boolean equals()
public static void     fill()
public static void     sort()
```

**Note:** The all the methods in the Arrays Class are static. Thus, they are invoked using the prefix Arrays, instead of with the name of an instance of the class.

### **java.lang.String**

```
Char   charAt(int index)
int   compareTo(String Other)
boolean endsWith(String suffix)
boolean equalsIgnoreCose(String other)
int  indexOf(String Str)
int  indexOf(String Str, int Fromindex)
int  lastIndexOf(String str)
int length()
String replace(char newchar, char newChar)
boolean startsWith(String prefix)
String substring (int beginIndex)
String substring (int beginIndex, int endIndex)
String toLowerCase()
String toUpperCase()
String trim()
```

### **java.swing.JOptionPane**

```
Static String showInputDialog(object message)
```

### **java.lang.System**

```
Static void exit(int Status)
```

### **java.text.NumberFormat**

```
Static NumberFormat getCurrencyInstance()
Static NumberFormat getPercentInstance()
Static NumberFormat getNumberInstance()
String format(double number)
void setMaximumFractionDigits (int digits)
```

"Write Once, Run Anywhere" အမို့ယ်ကတော့ တစ်ကြိုင်ရဲး နေရာတိုင်းသုံးဆိုတဲ့ အမို့ယ်ပါပဲ။ တချို့သူတွေကတော့ ပြောလေ့ရှိပါတယ်။ ဒါဟာ Java Applet တွေနဲ့သာဆိုပါတယ်လို့ ပြောကြပါတယ်။ အမှန်တကယ်ကတော့ Java Applet တွေသာ ကော Java Applet တွေ Java Application မှ အစပြောသာ Java နဲ့ ဆက်နွှယ်နေသော source code အားလုံးပါ အကျိုးဝင်ပါတယ်။ ဂဲ စတင်လောကြပါနို့။ ဒါဟာ Java ရဲ့ စွမ်းရည် တစ်ခုပါလို့ ပြောရင်လည်း မမှားပါဘူး။

### **Introduction to JAVA API (Application Programming Interface)**

JAVA Language ကို လေ့လာတဲ့အခါမှာ အထောက်အကျ ရေစိန်အတွက် အခြေခံအဆင့်မှာ အသုံးဝင်လောက်မယ်ထင်တဲ့ object class တွေထဲက API method တွေကို ရေးသားပို့ချေပေးထားပါတယ်။ သတိပြုရမှာကတော့ အခု ကျွန်တော်တင်ပွဲတဲ့ API method ဟာ Introduction လုပ်တယ်ဆိုတာကို မမေ့ဖို့ပါပဲ။ ဂဲ စတင်လောကြရအောင်။

### **The Maths Class (java.util.Math Class)**

```
public static double      abs(double x) //absolute value
public static native double aton(double x) //asciiongent
public static native double ceil(double x) //ceiling
public static native double cos(double x)
public static native double sin(double x)
public static native double tan(double x)
public static native double sqrt(double x)//square root
public static native double log(double x) //base e
public static native double exp(double x) //base e
public static native double floor(double x)
public static native double max(double x, double y)
public static native double min(double x, double y)
public static native double pow(double x, double y)
public static long        round(double x)
public static synchronized double random()
```

### **Convert Between Degrees and Radians**

```
public static double toDegrees(double radians)
//converts the specified number of radians to degrees
public static double toRadians (double degrees)
//converts the specified number of degrees to radians
```

### **Methods that return Random Numbers using Random Class**

```
nextBoolean()    //return a random boolean value
```

nextDouble()	//return random double or float values between 0.0 and 1.0
nextFloat()	//return random double or float values between 0.0 and 1.0
nextInt()	//return random int or long value (all possible value)
nextLong()	//return random int or long value (all possible value)
nextGaussian()	//return a double value between -1.0 and 1.0

### **The Array Class (java.util.Arrays class)**

```
public static int      binarySearch()
public static boolean equals()
public static void     fill()
public static void     sort()
```

**Note:** The all the methods in the Arrays Class are static. Thus, they are invoked using the prefix Arrays, instead of with the name of an instance of the class.

### **java.lang.String**

```
Char   charAt(int index)
int   compareTo(String Other)
boolean endsWith(String suffix)
boolean equalsIgnoreCose(String other)
int  indexOf(String Str)
int  indexOf(String Str, int Fromindex)
int  lastIndexOf(String str)
int length()
String replace(char newchar, char newChar)
boolean startsWith(String prefix)
String substring (int beginIndex)
String substring (int beginIndex, int endIndex)
String toLowerCase()
String toUpperCase()
String trim()
```

### **java.swing.JOptionPane**

```
Static String showInputDialog(object message)
```

### **java.lang.System**

```
Static void exit(int Status)
```

### **java.text.NumberFormat**

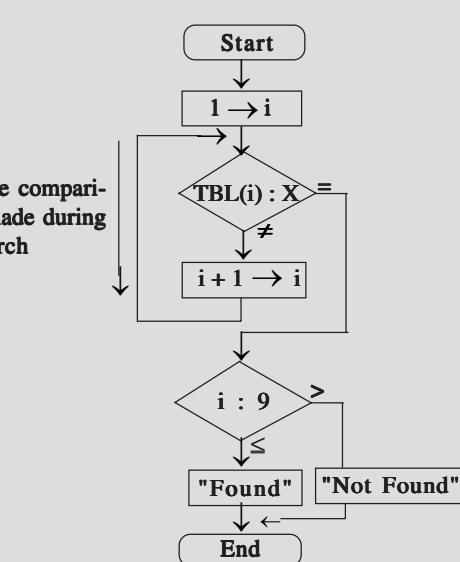
```
Static NumberFormat getCurrencyInstance()
Static NumberFormat getPercentInstance()
Static NumberFormat getNumberInstance()
String format(double number)
void setMaximumFractionDigits (int digits)
```

ကဲ Figure 1.7.3 မှာတော့ Linear(or Sequential) search method ဒဲ Flowchart ကို ရေးဆွဲပြထားပါတယ်။ လေ့လာ ကြည့်ပါ။ အကျယ်ချုပြုး မရင်းသေးပါဘူး။ ဂိုအပ်လာရင် စနစ် တကျ အပြည့်စုံး ပို့ချေပေးမှာပါ။ လေ့လာကြည့်ပါဦး။ အလွန်

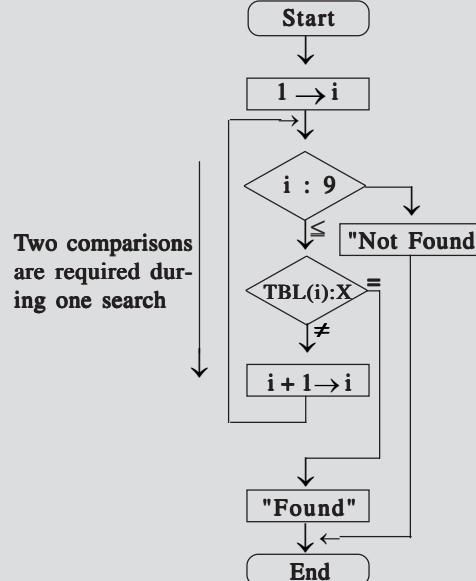
လွယ်ကူပါတယ်။ ဒီအခါမှာ Sentinel search method ကို အသုံးပြုထားတဲ့ Flowchart နဲ့ Sentinel search method ကို အသုံးမပြုဘဲ ရေးဆွဲထားတဲ့ Flowchart ဆိုပြီး Diagram ကို J မျိုးခွဲပြီး ဆွဲပြထားပါတယ်။

**Figure 1.7.3 A comparison of the case where the sentinel search method is used and the case where it is not used**

<The case where the sentinel search method is used>



<The case where the sentinel search method is not used>



ကဲ Linear (or sequential) search method အကြောင်းကို သဘောပေါက် သွားရင်တော့ Binary Search method အကြောင်းကို ဆက်လက် လေ့လာကြည့်ပါ။

### Binary Search method

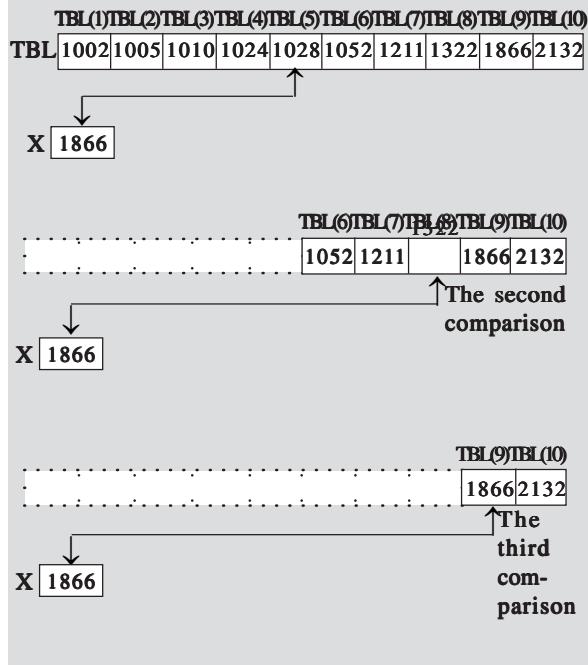
Binary search method ဒဲ search method ကတော့ search လုပ်လိုတဲ့ data နဲ့ table တုစ္ဆိတုလုံးကို dividing into two parts လုပ်ပြီးတော့ compare လုပ်ပြီး ရှာဖွေတာပါပဲ။ သူ့၏ method ဟာ Linear search method နဲ့ ဆင်တူသလိုရှိပေမယ့် ကွာဟာချက်ရှိပါတယ်။ အဲဒါကတော့ binary search method နဲ့ search လုပ်တဲ့အခါမှာ table မှာ ပါဝင်တဲ့ elements data တွေဟာ ascending or descending order လုပ်ထားရပါ မယ်။ အမိဘာယ်ကတော့ sort လုပ်ပြီးမဲ့ search လုပ်လို့ရတဲ့ အမိဘာယ်ပါပဲ။ ကဲ ရှင်းသွားအောင် Figure 1.7.4 အနေနဲ့ ရေး ပြထားတဲ့ Algorithm နဲ့ compare Technique ကို လေ့လာ ကြပါနို့။

**Figure 1.7.4 Algorithm for the binary search method**

- Step 1: A total of the value of a subscript representing the top of a table and that of a subscript representing the end of a table is divided by 2.
- Step 2: The elements having the value obtained in step 1 as a subscript are compared with a target element.
- Step 3: If there is an element that matches a target element, the search is successful.
- Step 4: If the value of a target element is smaller than that of an element in a table, 1 is subtracted from the current subscript and the value is used as a subscript for representing the end of a table.  
If the value of a target element is larger than that of an element in table, 1 is added to the current subscript and the value is used as a subscript for representing the top of a table.
- Step 5: Step 1 through step 4 is repeated. If an element matching a target element cannot be found at the point where the value of a subscript

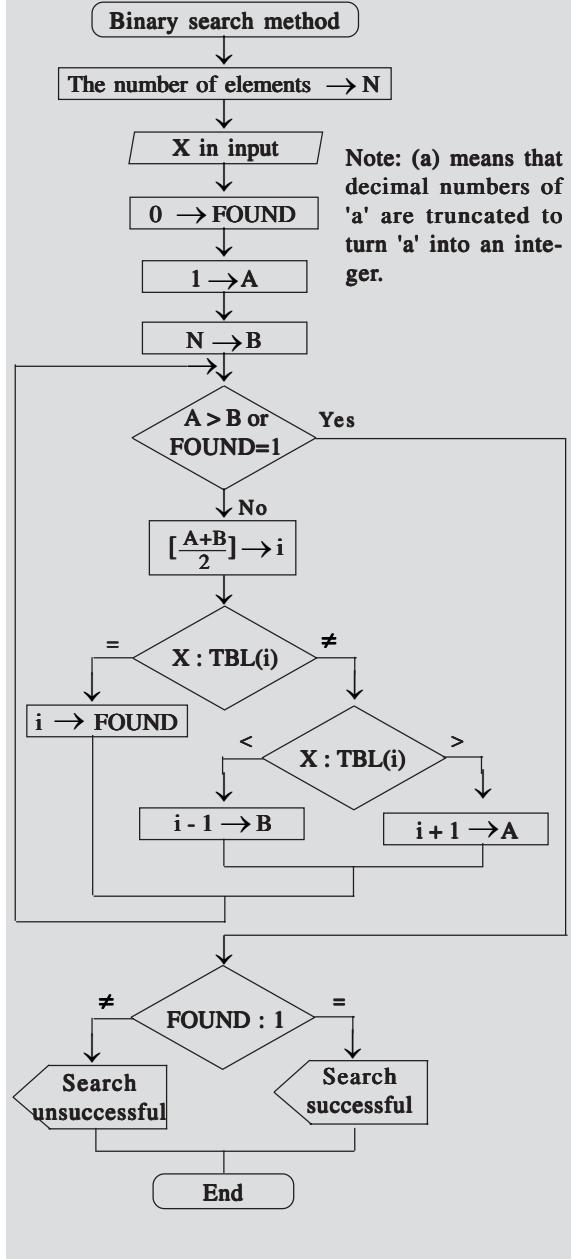
ဒီအထောက် Figure 1.7.4မှာ ရေးပြခဲ့တဲ့ Algorithm statements တွေကို တစ်ကြောင်းဆီ ဖော်ကြည့်ပြီး နားလည် သဘောပေါက်သွားရင် factorials in compare technique ကို ဆက်လက် လေ့လာကြည့်ပါ။ Continue in Figure 1.7.4 ဆိုပြီး ထည့်သွင်း ရေးဆွဲပေးထားပါတယ်။ Figure 1.7.4အနေနဲ့ ရေးပြထားတဲ့ Algorithm Statements တွေနဲ့ compact technique တွေကို တွေ့တာနဲ့ Binary search method ကို နားလည်ရင်းလင်းသွားမှာပါ။ ပိုပြီးပြည့်စုံသွားအောင် Figure 1.7.5 အနေနဲ့ ရေးပြထားတဲ့ Flowchart ကို လေ့လာကြတာပေါ့။ အောက်မှာ Diagram ရေးပြထားပါတယ်။

Continue in Figure 1.7.4



က ဒါဆိုရင်တော့ Algorithm ရဲ့ title တစ်ခုဖြစ်တဲ့ search Algorithm အကြောင်းကို အကြမ်းဖျင်းအားဖြင့် ရှင်းလင်းခဲ့ပြီ။ ဒီတော့ အမှန်တကယ်ရည်ရွယ်ထားတဲ့ sort Algorithm အကြောင်းကို အပြည့်စုံစုံဖြစ်အောင် 1.5.1 Sort Algorithm ဆိုပြီး sub-title အနေနဲ့ ရေးသားပို့ချပေးထားပါတယ်။ ဒီအခန်းက လူတော်မူတဲ့ အလွန်အရေးကြီးတဲ့ အတွက် သေသေချာချာ လေ့လာပါလို့ သတိပေးချင်ပါတယ်။ အခ ကျွန်ုတ် ရေးသား တင်ပြခဲ့တာတွေကို အမှန်တကယ် နားလည်မှသာလျှင် / အမှန်တကယ် တတ်မြောက်ကျမ်းကျင်မှသာလျှင် programming နယ်

Figure 1.7.5 Flowchart of the binary search method



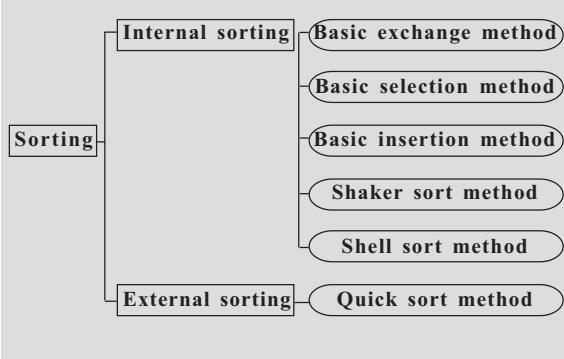
ပယ်မှာ လူတန်းစွဲနိုင်ပြီး အဆင့်ဆင့် ဆက်လက်တက်လမ်းနှင့်မှာ ဖြစ်တဲ့အတွက်ကြောင့် ဖြစ်ပါတယ်။ က 1.5.1 Sort Algorithm ဆိုတဲ့ အခန်းက လူတော်မူတဲ့ စတင် လေ့လာကြပါစို့။  
ကျော်ဇူးတော်မူတဲ့ (ကသာ)

## JAVA Developer Guide

### 1.5.1 Sort Algorithm

Sorting data လုပ်တယ်ဆိတာဟာ အလွယ်တွေ့ညွှန်မယ်ဆိုရင် small to large values ဆိုတော့ ascending order sorting လို့ သတ်မှတ်ပါတယ်။ vice versa သဘောတရားအားဖြင့်တော့ large to small value ဆိုရင် descending order sorting လို့ သတ်မှတ်ရမှာပါ။ ဒီလို့ sorting လုပ်ခြင်းဟာ programming လောကမှာ အလွန်အရေးပါတဲ့ features တွေပါပဲ။ ဒါကြောင့် အလွန်အရေးပါတဲ့ Sort Algorithm အပိုင်းကို ရှင်းပြမှာပါ။

**Figure 1.8 Sort methods**



အပေါ်မှာဆွဲပြထားတဲ့ Figure 1.8 ကိုတွေ့ရရင် sorting လုပ်တဲ့ method မှာ Internal sorting နဲ့ External sorting ဆိုပြီး method နှစ်ခုကဲပြားနေတာကို တွေ့ရမှာပါ။ Internal sorting ကို ပြန်တွေ့တဲ့အခါမှာတော့ အောက်မှာပြထားတဲ့ အတိုင်းပဲ တွေ့ရမှာပါ။

- Internal Sorting
  - Basic exchange method
  - Basic selection method
  - Basic insertion method
  - Shaker sort method
  - Quick sort method

ဒီတော့ Internal sorting မှာ Basic exchange, Basic selection, Basic insertion, shanker sort, shell sort နဲ့ Quick sort method ဆိုပြီး method (6) မျိုးကို တွေ့ရပြီဖော်။ ကဲ ဒီလို့ အလွယ်မှတ်တွေ့ရှိနော်။

- Basic exchange method (Bubble sort)
- Basic selection method (Smallest data is selected and then compare)
- Basic insertion method (like Bubble sort)
- Shaker sort method (like Bubble sort)
- Shell sort method (extended version of the Basic insertion) GAD = Data size/2 method
- Quick sort method (include JAVA util parkage)

သားမှာ bracket ( ) ထဲက ရေးထားတဲ့ Notes တွေကတော့ သိသာထင်ရှားအောင် မှတ်ရလွယ်အောင် ရေးပေးထားတာပါ။ ဒါဆိုရင် Internal sorting နဲ့ ပတ်သက်ပြီးတော့ method (6) မျိုး ကဲတာကို သဘောပေါက်ရင် External sorting ကို ဆက်လေ့လာတွေ့ရအောင်။

External Sorting ——————> Merge Sort Method (Divide - and conquer method)

Internal sorting နဲ့ External sorting အလုပ်လုပ်ပုံတဲ့ ကွဲပြားစွာ နားမှု ရှိပါတယ်။ အခြေခံအနေနဲ့ ပြောရရင်တော်မှ internal sorting တာ main memory unit မှာ အလုပ်လုပ်ပြီး တော့ External sorting ကတော့ magnetic disk and other auxiliary storage unit တွေမှာ အလုပ်လုပ်ပြီး ဆိုလောင်ပါတယ်။ ဘယ် sorting method ကိုပဲရွေးရွေးပါ။ အဓိကအကျိုးကျေးဇူး ကတော့ အချိန်ကုန်သက်သာဖေတယ်။ တွက်ချက်မှာ စွမ်းရည်မှာ လိုအပ်သလို ကူညီပေးတယ်။ ဒါကြောင့် arrays အခန်းကဏ္ဍတွေကို သင်္ကားပြီဆိတာ နဲ့ Sort Algorithm ကို မေတာလို့ မရပါဘူး။ ကိုယ်ရွေးချယ်တဲ့ sort method ကိုလဲ အလုပ်လုပ်သွားပဲ the time of calculation (computational complexity) method နဲ့ ချတ်ပြီး လေ့လာသင့်ပါတယ်။ ကျွန်တော့အနေနဲ့ ကတော့ programming လောကမှာ နာမည်ပြီး method ဖြစ်တဲ့ Bubble

Sort (OR) Basic exchange method ကိုပဲ သင်ကြားပို့ချေပါမယ်။ ကျွန်တဲ့ Sorting method တွေကို လိုအပ်ရင် လိုအပ်သလို ပို့ချေသွားပါမယ်။

**Note:** Bubble Sort method ကိုက ရွှေ့ချယ်ရုံးလဲဆိုတော့ Bubble Sort (OR) Basic exchange method ဟာ အခြေခံ အကျော်း method ဖြစ်ပါတယ်။ ဤော်လည်း Exam Question တွေမှာလဲ အမေးအများဆုံး Method type ဖြစ်နေလိုပါပဲ။ ကျွန်တဲ့ method တွေကိုလဲ လိုအပ်ရင်လိုအပ်သလို လေ့လာရမှာ ပါ။ Java.util.\*; Package မှာ ပါဝင်တဲ့ Sort( ) API method ရဲ့ အခြေခံ Quick Sort method ကိုလည်း algorithm & Flowchart ကိုလည်း ထည့်သွင်းပေးထားပါတယ်။ အလွန်အရေးပါတဲ့ Array class တွေရဲ့ ပင်မဖြစ်တဲ့အတွက် သေသေချာချာ လေ့လာပါ။ ကဲ စတင်လေ့လာကြပါစို့။

### Basic exchange Method (Bubble Sort)

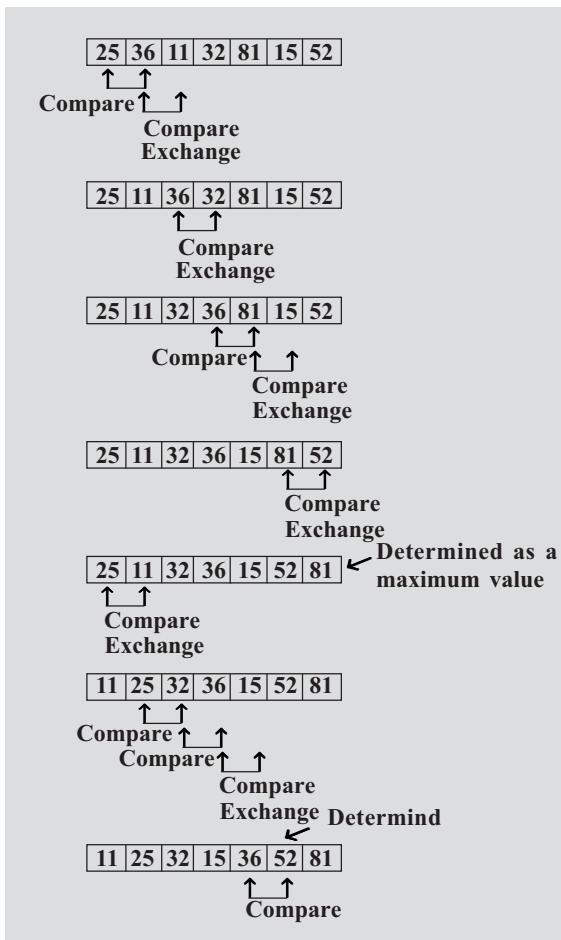
Basic exchange method (Bubble Sort) ဆုံးတာကတော့ data တွေကို Sequentially used to compare လုပ်တာပါပဲ။ Ascending order အရ ပြောရရင်တော့ Array ရဲ့ data တွေကို index of 1<sup>st</sup> array (Array ရဲ့ စစ်ချင်း data တွေကာနေ စပြီးတော့ မှ) နှစ်လုံးတစ်တွဲစီ compare လုပ်ပြီးတော့ ယောက်တွေက Array ရဲ့ head of index ကို ရွှေ့သွားတဲ့ထံနဲ့ အလုပ်လုပ်တာပါ။ နှစ်လုံးတစ်တွဲ နှင့်ယဉ်ရာကနေ ယောက်တွေ data ဟာ ရှေ့ကို ရွှေ့သွားတဲ့ method နဲ့ Array ရဲ့ head to tail ကို ဖြည့်ဖြည့်ချင်း ရွှေ့သွားပါတယ်။ Array ရဲ့ test index ကို ရောက်သွားရင်တော့ returns to repeated လုပ်ပါတယ်။

ဒီလိုနည်းနဲ့ပဲ Array ရဲ့ data item တွေကို exchange လုပ်ပေးသွားပါတယ်။ ဒီလို အလုပ်လုပ်သွားတဲ့ Method ကို Basic exchange method (OR) Bubble sort method လို့ ခေါ်ပါတယ်။ အလွန်ရှိုးရှင်းတယ်လို့ နာမည်ကြီးတဲ့ simplest, best known sort method တစ်ခု ဖြစ်ပါတယ်။ Bubble sort လို့ ခေါ်ခေါ်ရတဲ့ အကြောင်းရင်းကတော့ movement of maximum or minimum data တွေကို နေရာရွှေ့ပြီး အလုပ်လုပ် တာဟာ Looks like bubbles popping out on the surface of the water လို့ အဓိုာတ် ဖွှဲ့ဆိုထားလိုပါပဲ။ မှတ်သားရ လွယ်ကူတဲ့ method တစ်ခုပါပဲ။

အောက်မှာ Figure 1.9 အနေနဲ့ Bubble sort (OR) Basic exchange method ရဲ့ Algorithm နဲ့ compare exchange step တွေကို ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။ Algorithm ကို သေသေချာချာဖတ်ပြီး နားလည်သွားပြီ့ဆိုရင် တော့ compare exchange step တွေကို လေ့လာပါ။

**Figure 1.9 Steps of the basic exchange method (Bubble sort)**

Algorithm for sorting data in the ascending order  
 Step 1. The first and second elements in a table are compared.  
 Step 2. If the first element is larger than the second, the first is exchanged with the second  
 Step 3. If the second element is larger than the first, no exchange occurs.  
 Step 4. The second and third elements are compared and step 2 and 3 are repeated.  
 Step 5. This routine operation is repeated to the last element in a table. As it reaches the last element, a maximum value is stored in the last element in that table.  
 Step 6. Steps 1 thorough 4 and 5 are executed until the operation reaches the last element but one.  
 Step 7. Steps 1 through 6 are repeated until only the first and second elements in a table remain.  
 This completes data sorting.



<Characteristics>

- One of the simplest sort methods
- The efficiency is low since data item are unconditionally compared even if they are sorted correctly.
- If the volume of data is large, the process is time-consuming.

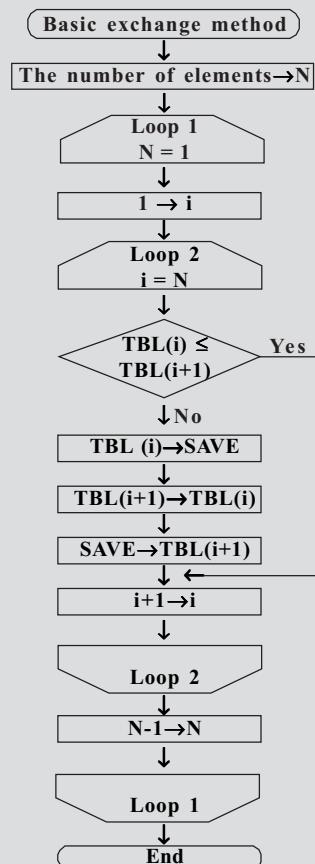
<Computational complexity>

- Maximum computational complexity:  $O(n^2)$
- Average computational complexity:  $O(n^2)$

ဒါ Figure 1.9မှာ ရေးသားပြထားတဲ့ Algorithm statements တွေဟာ ကျွန်ုတ် computer science နယ်ယယ်ကို ရောက်စဉ်ကတည်းက (စတင်ဝင်ရောက်စဉ်ကတည်းက) လေ့လာခဲ့သူမျှ စာအုပ်တွေထဲမှာ အခါ ကျူးယူဖော်ပြထားတဲ့ method တဲ့ အကောင်းဆုံးနဲ့ အပြည့်စုံဆုံးဆိတ်တော်ရဲ့ သင်တြားမှာ အတွေ့အကြုံကိုပါ အမြဲခြီး အမခံပါတယ်။ လေ့လာကြည့်ပါ။

**Figure 1.10 shows the flowchart of the basic exchange method. (Bubble sort)**

**Figure 1.10 Flowchart of the basic exchange method**



က ဒါတော့ Figure 1.9မှာ ရေးဆွဲပြထားခဲ့တဲ့ Algorithm and compare exchange step တွေကို နားလည် သဘောပါက သွားပြီဆိုရင်တော့ Figure 1.10မှာ ရေးဆွဲပြထားတဲ့ Basic exchange Method (Bubble sort) ရဲ့ Flowchart ကို လေ့လာကြည့်တာပါ။ အလွန်စနစ်ကျွဲ့ Method တစ်ခု ဖြစ်တာ ကြောင့် Program ရေးမပြခင်မှာ အပြည့်စုံဆုံးဖြစ်အောင် ရှင်းပြန်တော်ပါ။ က ဒါတော့ Bubble sort (Basic exchange method) ရဲ့ Flowchart ကို လေ့လာကြည့်ရအောင်။

**Note :** Algorithm & Flowchart ဆိုတဲ့ အခေါ်အဝေါ်တွေ ကို ဘဏ်တဲ့ အနေနဲ့ စိမ်းကောင်း စိမ်းနေနိုင်းပါတယ်။ ဒီနေရာမှာ ကျွန်ုတ်အနေနဲ့ introduce လုပ်ပြီးရှင်းပြထားပါတယ်။ အမှန်တကယ် တတ်မြောက်အောင် သင်ကြားပို့ချပေးမှာ ဖြစ်တဲ့ အတွက် စိတ်အေးပောက်အေးနဲ့သာ ဆက်လက် လေ့လာသွားပါလို့ သာ ပြောချင်ပါတယ်။

**Listing 1.22 Illustrate of the Basic exchange method (OR) Bubble sort method**

```

1. import java.util.*;
2. class bubblesort{
3.     public static void main (String [ ] args){
4.         int[ ] a={25, 36, 11, 32, 81, 15, 52};/
5.          /Declare of seven arrays data item
6.         Random myRand=new Random( );// creates a random objects
7.         for(int i = 0: i<7; i++){
8.             a[i] = myRand.nextInt( );
9.             System.out.println ("Before Sorting");
10.            for (int i =0; i<7; i++) {
11.                System.out.Println("a["+ + i+" ]=" +
12.                +a[i]);
13.                boolean sorted = false;
14.                while (!sorted){
15.                    sorted = true
16.                    for (int i=0; i < 6; i++ ){
17.                        if (a[i] > a [i+1]){
18.                            int temp = a[i];
19.                            a[i] = a[i+1];
20.                            a[i+1] = temp;
21.                            sorted = false;
22.                        }
23.                    }
24.                    System.out.println("After Sorting");
25.                    for (int i=0; i < 7; i++) {
26.                        System.out.println("a["+ + i+" ]= "+a[i]);
27.                    }
28.                }
29.            }
30.        }
31.    }
  
```

```

output
Before Sorting
a[0] = 25
a[1] = 36
a[2] = 11
a[3] = 32
a[4] = 81
a[5] = 15
a[6] = 52
After Sorting
a[0] = 11
a[1] = 15
a[2] = 25
a[3] = 32
a[4] = 36
a[5] = 52
a[6] = 81
Analysis

```

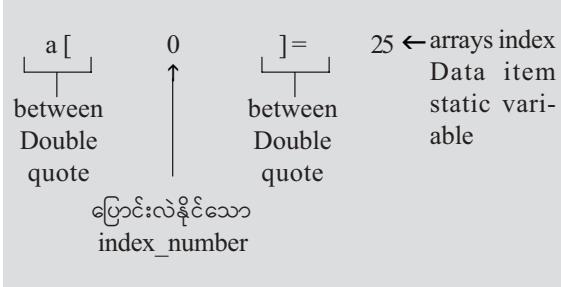
- ၁။ လိုင်းနံပါတ် (၁) မှာ Arrays နဲ့ ပတ်သက်တော့ကို ခေါ်ယူ အသုံးပြုနိုင်ခဲ့အတွက် import java.util.\*; ဆိုတဲ့ Java ရဲ့ package class ကို ခေါ်ယူထားတာကို တွေ့ရမှာပါ။ အဲဒီနောက်မှာမှ bubblesort ဆိုတဲ့ primary class ရယ် main () function ရယ်ကို လိုင်းနံပါတ် (၂) နဲ့ လိုင်းနံပါတ် (၃) မှာ ကြေညာပါတယ်။
- ၂။ အဲဒီနောက် int [] a ဆိုပြီးမှ one-dimensional array ကို အသုံးပြုပြီးတော့ data item 7 ခုကို Declare လုပ်ပါတယ်။ Random class ကို အခြေခံပြီးတော့မှ object တဲ့ ခုကို create လုပ်လိုက်တာကို လိုင်းနံပါတ် (၅) မှာ တွေ့ရမှာပါ။ အမှန်တကယ်ကတော့ nextInt (), nextBoolean (), nextFloat () အစရှိတဲ့ method တွေကို ခေါ်ယူထားတာပါ။ Bubble sort နဲ့ ကိန်းတွေကို စီစဉ်တဲ့အေါ်မှာ nextInt () method နဲ့ ခေါ်ယူရမယ် မဟုတ်လား။ ဒါ Random Class နဲ့ ပတ်သက်ပြီး အခြေခံအနေနဲ့ သိထားသင့်တဲ့ JAVA ရဲ့ API method တွေတိပါ listing 1.21 အပြီးမှာ ပို့ချေပေးခြေားပါပြီ။ လိုအပ်ရင်ဖြစ်စေ JAVA နဲ့ ပတ်သက်တဲ့ Certified JAVA programmer တစ်ယောက်၊ Certified JAVA Developer တစ်ယောက် ဒါမှမဟုတ် Certified JAVA Master တစ်ယောက်အထိ မျှော်လင့်ရည်မှန်းထားရင် ဒါ API method တွေကို နှိမ်နှိမ်ချတ်ချတ် သိထားသင့်ပါတယ်။ ဒါကြောင့် ထပ်မံပါတယ်လဲ ဖတ်မှတ်လေ့လာပါ။ ထပ်မံပြီး ဖြည့်စွက်ချက်တွေအနေနဲ့ API method တွေကို ပို့ချေပေးလို့မှာပါ။
- ၃။ လိုင်းနံပါတ် (၆) မှာတော့ for looping ကို အသုံးပြုပြီးတော့ looping ပတ်ပါတယ်။ ဒါဟာ 0 က စတယ် 7 ထက် ငယ်စဉ်

မှာ အလုပ်လုပ်မယ်။ ပြီးတော့ i ကို postfix ပုံစံနဲ့ increment လုပ်ပြီး တိုးသွားပါတယ်။ ပြီးတော့ for looping ရဲ့ block ထဲကို text expression လုပ်တဲ့အခါတိုင်း ဝင်ပြီး တော့ a[i]=myRand.nextInt(); ဆိုပြီးမှ ဒါ တန်ဖိုးနဲ့ ရှိနိုင်ပါတယ်။ Dimensional array အနေနဲ့ ခေါ်ယူတဲ့ အခါမှာ i ကို အသုံးပြုရမှာ ဖြစ်လိုပါ။

- ၄။ လိုင်းနံပါတ် (၈) မှာ System.out.println("Before Sorting"); ဆိုပြီးမှ output အနေနဲ့ print ထုတ်ပေးပါတယ်။ လိုင်းနံပါတ် (၉) မှာတော့ sorting မလုပ်ခင် array ရဲ့ data item တွေကို print ထုတ်ပေးလို့ looping ပတ်ပါတယ်။ For looping ရဲ့ block brace ထဲကို တစ်ကြိမ်ဝင်တိုင်း System.out.println () method ကို အသုံးပြုပြီးတော့ output ကို ထုတ်ပေးမှာပါ။ ဒီနေရာမှာ သတိပြုရမှာကတော့ output ထုတ်တဲ့အခါ arrays ရဲ့ index\_number တွေကို i အနေနဲ့ equally လုပ်ပြီးတဲ့အခါမှာ "a[" + index\_number + "]="+a[i] အနေနဲ့ ရေးထားတာကို တွေ့ရမှာပါ။ ဒီအခါမှာ ရှင်းအောင် ရေးပြရရင်တော့ "a[" ဟာ Double quote နှစ်ခုကြား ရေးထားတဲ့အတွက် a[ လို့ output ထုတ်တဲ့အခါ ပေါ်လာပြီးတော့ ပြောင်းလဲနိုင်တဲ့ index\_number ကိုတော့ +i+ လို့ ရေးထားတာကြောင့် value ကို increment လုပ်တိုင်း 0,1,2,... အစရှိသဖြင့် တိုးတိုးပြီး လာပြနေတာပေါ့။ ပြီးတော့ "]=" လို့ ရေးထားတော့ Double quote နှစ်ခုကြားရေးထားတဲ့ ]= ဟာ output အနေနဲ့ လာပြတာပေါ့။ ဒီဘက်မှာ ရေးထားတဲ့ +a[i] ကတော့ အမှန်တကယ် arrays ရဲ့ ရောက်ခဲ့ data item တွေပါ။ ဒါကြောင့် output ထွက်လာတော့ a[0]=25 လို့ လာပြတာပေါ့။ ဒါကို စာဖတ်သူရဲ့ မျက်စိတ်မှာ ရှင်းအောင် ရေးပြရရင်တော့ ဒီလိုပါ။

- ၅။ ဒါ အဲဒီလိုနည်းနဲ့ ရေးပြီးတော့ sorting မလုပ်ရသေးတဲ့ (မစီရသေးတဲ့) Arrays data item တွေကို output ထုတ်ပြလိုက်ပါပြီ။ ဒါတော့ Before sorting ဆိုတဲ့ စာသား ခေါင်းစဉ်ရဲ့အောက်မှာ 25, 36, 11, 32, 81, 15, 52 ဆိုပြီး Line by Line ပေါ်လာပြမှာပါ။ ကဲ့ . ဒီအထိတော့ ရှင်းသွားပြီနော်။ လိုင်းနံပါတ် (၁၁) မှာ boolean data type ကို အသုံးပြုပြီးတော့ boolean sorted = false; လို့ Declare လုပ်ပါတယ်။ အမိုးယ်ကတော့ false ဖြစ်ရင် အလုပ်ဆက်လုပ်ဖို့ ကြေညာတာပါ။ လိုင်းနံပါတ် (၁၂) မှာ while control statement ကို အသုံးပြုပြီးတော့ false ဖြစ်ခဲ့ရင် (ရှင်းအောင်ပြောရရင်တော့ sorting မစီရသေးရင်ပေါ့) while ရဲ့ looping block ထဲကိုဝင်ပြီးတော့ အလုပ်လုပ်ပါမယ်။ block ထဲကို ရောက်တာနဲ့ sorted ဟာ true ဆိုပြီး

ပြောင်းလဲ ခံရပါတယ်။ trueဖြစ်မှုလဲ အလုပ်ကို ဆက်လည်လို့ ရှုမှုလေး။



- ၆။ လိုင်းနံပါတ် (၁၄) မှာတော့ for loopingကို ပတ်ပြီးတော့ loopingရဲ့ blockထက်ကို ဝင်တဲ့အခါမှာတော့ Bubble sortနဲ့ ကိန်းစဉ်စီမံခြောင်းဖြစ်တဲ့ numerical methodနဲ့ ပြီး စိပါပြီ။ ဒီအခါမှာ for loopingကို ပတ်တဲ့အခါမှာ Arrays တွေရဲ့ စစ်ချင်း ကိန်းတစ်လုံးကို ကိန်းသေထားမှာ ဖြစ်တဲ့အတွက် မူလရှိတဲ့ Data itemsအရေအတွက်ထက် တစ်ချောက်ပြီး loopingပတ်ထားပါတယ်။ ဒါကို သတိပြုထားပါ။ အောက်မှာ ရေးထားတဲ့ Bubble sortနဲ့ numerical methodကလည်း Algorithm & Flowchart မှာ ပြုခဲ့တဲ့အတိုင်း ကိန်းနှစ်လုံးဟာ တစ်ခုနဲ့တစ်ခု compareလုပ်တဲ့ methodနဲ့ testလုပ်ပါတယ်။
- ၇။ အမိုးပွားယ်ကတော့ Arraysရဲ့ 1<sup>st</sup> index ဟာ Arraysရဲ့ 2<sup>nd</sup> index ထက်ကြီးခဲ့ရင် 1<sup>st</sup> index ကိုပဲ temp လို့ မှတ်ထားမှာပါ။ ပြီးတော့ 1<sup>st</sup> index ဟာ 2<sup>nd</sup> indexနဲ့ နေရာရွှေ့ဖို့အတွက် tempကိုပဲ ယာယိသိလောင်ပြီး နောက် ကိုရွှေ့လာမှာပါ။ အမိုးပွားယ် ကတော့ ငယ်တဲ့ကိန်းဟာ 1<sup>st</sup> index အဖြစ် နေရာရွှေ့လာပြီးတော့ ကြိုးတဲ့ကိန်းဟာ 2<sup>nd</sup> index အဖြစ် နေရာရွှေ့လာမှာပါ။ ဒီလိုနည်းနဲ့ Looping တစ်ပတ်ပတ်တိုင်း 2<sup>nd</sup> index နဲ့ 3<sup>rd</sup> index နောက် တစ်ကြိမ်ကျတော့ 3<sup>rd</sup> index နဲ့ 4<sup>th</sup> index စသဖြင့် compare testလုပ်ပြီး ငယ်စဉ်ကြီးလိုက် နေရာရွှေ့မှာပါ။ ကဲ သေသေချာချာ လက်နဲ့ချွောက်ကြည့်ပါ။ ရှင်းသွားမှာပါ။ လိုင်းနံပါတ် (၁၉) က close brace} က လိုင်းနံပါတ် (၁၅) က if blockရဲ့ အပိုက် brace} ပါ။ လိုင်းနံပါတ် (၂၀) ကတော့ for loopingရဲ့ close brace} ပါပဲ။ လိုင်းနံပါတ် (၂၁) ကတော့ while loopရဲ့ close brace တွေပါပဲ။ Looping တွေ တစ်ခုနဲ့တစ်ခု crossဖြတ်လို့ မရပေါယ့် looping တွေဟာ တစ်ခုနဲ့တစ်ခု အချင်းချင်း ငုံထားလို့ ရပါတယ်။ ဒါကြောင့် loopingကို ငုံပြီးတော့ ပတ်ထားတာပေါ့။

၈။ ဒီလိုနည်းနဲ့ပဲ Bubble sort methodနဲ့ ကိန်းတွေကို ascending orderအတိုင်း စီမံပြုဆိုကြပါနဲ့။ လိုင်းနံပါတ် (၂၂) မှာ print statementကို အသုံးပြုပြီးတော့ After sortingဆိုတဲ့ စာသားကို အရင်ရှိက်ထုတ်ပါတယ်။

၉။ လိုင်းနံပါတ် (၂၃) မှာ for loopingနဲ့ ပတ်ပြီးတော့ ascending orderနဲ့ စီမံပြုသားဖြစ်တဲ့ Arrays data item တွေကို outputအဖြစ် ရှိက်ထုတ်ပါတယ်။ ဒီအခါမှာတော့ loopingပတ်ဘဲအခါမှာ for loopingနဲ့ initializeနေရမ 7ထက် ငယ်တဲ့အထိဆိုပြီးမဲ့ loopingပတ်ခွင့်ပြထားတာကို တွေ့ရမှုပါ။ ဒီလိုနည်းနဲ့ outputကို ရှိက်ထုတ်ပါတယ်။ လိုင်းနံပါတ် (၂၄) မှာပဲ for loopingနဲ့ blockကို လာပိတ်ထားပါတယ်။ လိုင်းနံပါတ် (၂၅) ကတော့ main() functionအတွက် close brace} ဖြစ်ပြီး လိုင်းနံပါတ် (၂၆) ကတော့ Bubble sortဆိုတဲ့ primary classနဲ့ close brace တွေပါပဲ။

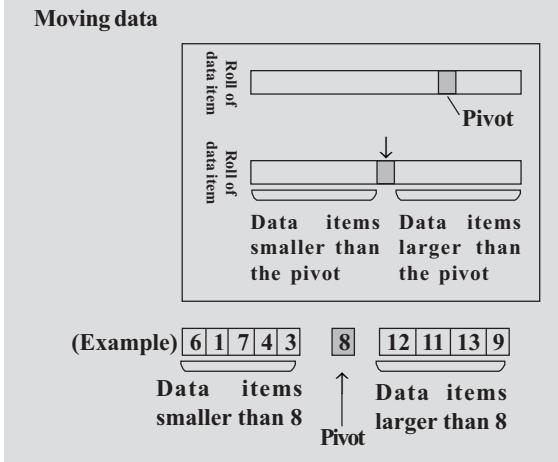
ဒီ Liting 1.22ကို လေ့လာကြည့်ရင်တော့ programming လောက / IT Exam လောကမှတ် အလွန်နာမည်ကြေးတဲ့ Bubble sort (sorting method)ကို သဘောပါက်သွားမှာပါ။ Bubble sortနဲ့ program code တွေကိုသာမက Algorithm, comparative style and flowchart တွေကိုပါ အထပ်ထပ် အခါမှာ လေ့လာပါ။ အလွန်အရေးကြီးပါတယ်။ Bubble sortအကြောင်းကို လေ့လာလိုပြီးသွားရင်တော့ Java languageမှာ java.util packageမှာ objectအနေဖြင့်လည်တဲ့ sort()ဆိုတဲ့ method ကို Quick Sort Methodနဲ့ ရေးသားထားပါတယ်လို့ ကျွန်တော်ရှုံးမှာ တင်ပြခဲ့ပြီးပါပြီ။ ဒီတော့ Quick sort method ကို ဆက်လက် လေ့လာကြတာပေါ့။ သေသေချာချာ လေ့လာကြည့်ပါ။

#### Quick Sort Method (include JAVA util parkage)

Quick Sort Method ကို design ချွဲတဲ့ သူကတော့ Hoareပါ။ သူအနေနဲ့ recursive method ကို အသုံးပြုပြီးတော့ data တွေကို sortလုပ်ရမှာ အလွယ်ကူးနဲ့ အပြည့်စုံဆုံးဖြစ်အောင် စီစဉ်ခဲ့ပါတယ်။ Quick sort method အနေနဲ့ sorting လုပ်မယ်ဆိုရင် data တွေကို pivot or pivotal value ကို အရင်ရှာပါတယ်။ အလွယ်ပြောရရင်တော့ median or intermediate value ကို ရှာဖွေလိုက်ပါတယ်။ Median valueနဲ့ ပတ်သက်ပြီးတော့ Figure 1.11ကို လေ့လာကြည့်ပါ။

Figure 1.11 Pivot (Pivotal value)

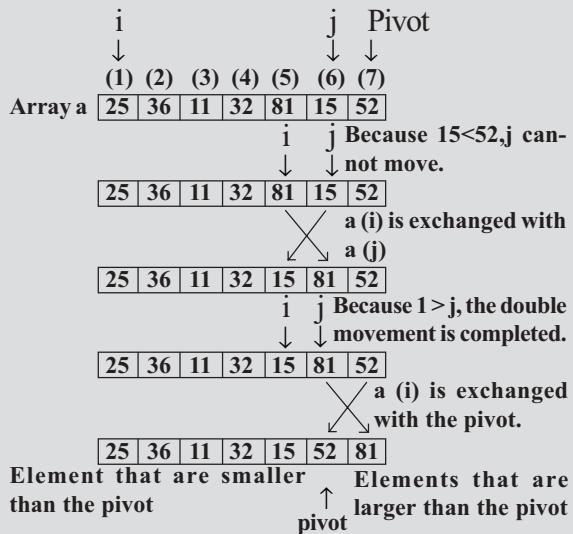
Row of data **6 | 11 | 7 | 4 | 3 | 8 | 12 | 1 | 13 | 9**  
↑ pivot (pivotal value)



Pivot ရဲ့ ဘယ်ဘက် မှာရှိတဲ့ data items တွေဟာ ငယ်နေပြီးတော့ pivot ရဲ့ ဥဘက်မှာရှိတဲ့ data items တွေဟာ ကြိုးနေတာကို တွေ့ရပါတယ်။ ဒါဟာဆိုရင် sorting လုပ်မယ့် data items တွေကို divided into two sections (division) လုပ်ခြင်းပါပဲ။ ကဲ ဒီအထိရပြီးနောက် ဒီလို data items တွေကို group နှစ်အပ်စု ခွဲပြီးတဲ့အခါမှာတော့ sort လုပ်ပါတယ်။ Sorting လုပ်တဲ့အခါမှာ သာမန် sort operation တွေထက် လွယ်ကူလျင်မြန်စွာ သွားနိုင်ပါတယ်။ Figure 1.12 ကို လေ့လာကြည့်ပါ။ Quick sort method ရဲ့ Algorithm နဲ့ calculation factorials ကို ရေးပြထားပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

Figure 1.12 Algorithm of the quick sort method

- Step 1:** As a preparatory step, pointers i and j must be initialized.
- i is at the left end, that is 1.
  - j is at the last position but a (7) of the pivot, that is 6.
- Step 2-1:** i is moved to the right until an element larger than the pivot is detected.
- Step 2-2:** J is moved to the left until an element smaller than the pivot is detected.
- Step 2-3:** a (i) is exchanged with a (j).
- Step 2-4:** Step 2-1, 2-2 and 2-3 are repeated until  $i \geq j$ .
- Step 2-5:** a (i) is exchanged with the pivot at the right end.



#### Algorithm for calculating factorials

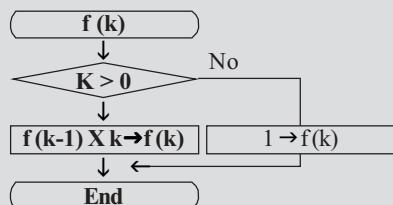
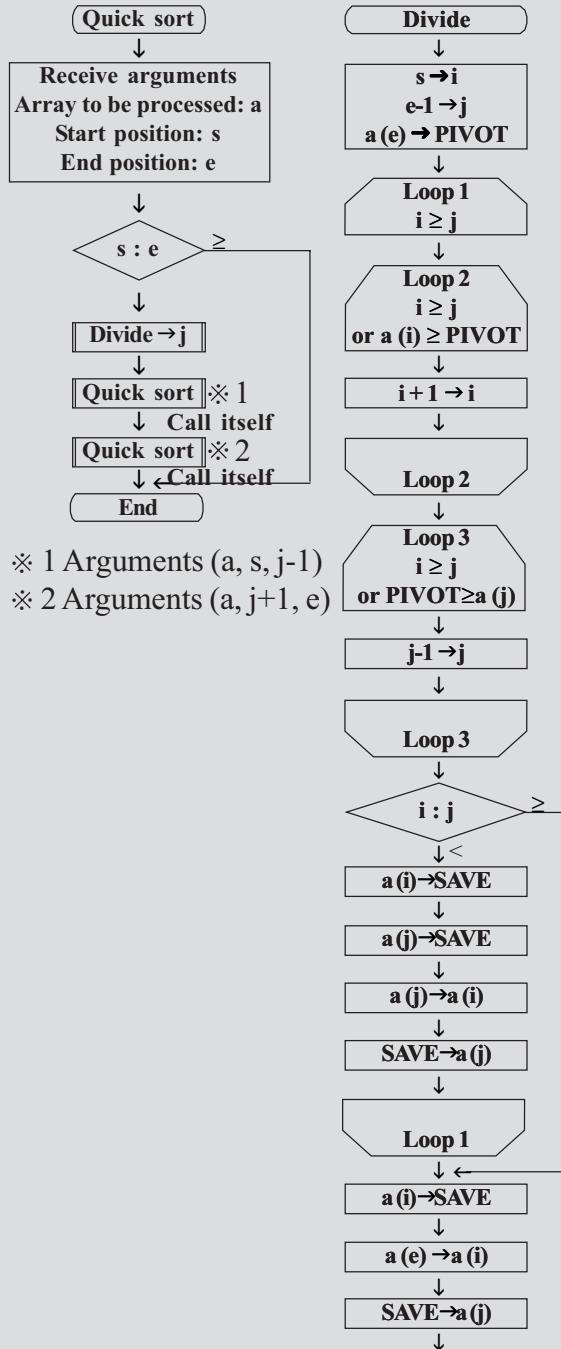


Figure 10.13 Algorithm of the quick sort method



\* 1 Arguments (a, s, j-1)  
\* 2 Arguments (a, j+1, e)

Figure 10.13 shows the flowchart of the quick sort method.

က ဒါဆိုရင် Quick sort method ရဲ့ Algorithm ရဲ့ calculation factorials ကို နားလည် သဘောပါက်သွားပြီ ဆိုရင်တော့ Figure 1.13 အနေနဲ့ ရေးဆွဲပြထားတဲ့ Quick sort method ရဲ့ flowchart ကို လေ့လာကြည့်ပါ။

က ဒါဆိုရင်တော့ Quick sort method နဲ့ ပတ်သက်တဲ့ Agorithm, Data တွေကို compare လုပ်ပြီးနေရာရွှေ့တဲ့ calculation factorials နဲ့ flowchart တွေကို လေ့လာပြီးသွားရင် တော့ Java.util parkage မှာ ပါဝင်တဲ့ sort() method ကို used လုပ်တဲ့အခါမှာ based on design လုပ်ထားတဲ့ Quick sort method ကို နားလည် သဘောပါက်သွားမှုပါ။ Quick sort method ကို Bubble sort method လိုတော့ program ရေးပြီး မရင်ပြတော့ပါဘူး။ Package ထဲမှာ ပါဝင်ပြီးသားမြို့ပါပဲ။ ဒါပေမယ့် Alogrithm & Flowchart တွေကိုတော့ အထပ်ထပ် အခါ လေ့လာပါ။

ကျွန်တော့အနေနဲ့ Arrays နဲ့ ပတ်သက်ပြီးတော့ ဒီအခန်းမှာ အပြည့်စုစုံပြစ်အောင် ပို့ချုပါတယ်။ ဒီအခါမှာ အချို့သော အဆင့်မြင့် arrays ပုံစံတွေကို ရှင်းပြုတဲ့အခါမှာ Algorithm & Flowchart ဆိုတဲ့ System Development နဲ့ Internal Design ပိုင်းဆိုင်ရာ နည်းပညာတွေ ပါဝင်လာတာကို စာဖတ်သူ တွေ့ရမှုပါ။ ဒီတော့ programmer အဆင့် (အခြေခံကဗျာပြုတော့) လေ့လာသင်ယူတဲ့အခါမှာ ဒီနည်းပညာတွေကို ချိန်ထားခဲ့လို့ မရပါဘူး။ ဒီ System Development နဲ့ Internal Design ပိုင်းဆိုင်ရာ method တွေကလည်း Java language တစ်ခုထဲမှာသာ မဟုတ်ပါဘူး။ C & C++, Ada, Java, Fortran, ..... အစရိုး သဖြင့် ဘယ် Language တွေမှာမဆို အသုံးဝင်ပါတယ်။ Program တစ်ခုရဲ့ Flow ဘွားကို program flowchart နဲ့ တွက်ချက် ရေးဆွဲပြရပြီးတော့ System Development ပါဖြစ်ဖြစ် အလုပ်လုပ်သွားတဲ့ Flow လိုတော့ System Flowchart နဲ့ အလုပ်လုပ် ရေးသားပြရ ပါတယ်။ ဒီတော့ ဒီ Flowchart method ဘာ အသုံးဝင်တယ် ဆိုတာကို အခြေခံလောက်တော့ သဘောပါက်လာလောက်ပါပြီ။ ဒါကြောင့် လုံးဝ မတတ်မြောက် / မသင့်ကြားဘူးသေးတဲ့ သူတွေက စပြီးတော့ တတ်မြောက်ထားသူတွေပါ နားလည်သဘောပါက်နိုင် ဖို့အတွက် Java ရဲ့ Sample Program ငယ်လေးတွေကို အသုံးပြုပြီးတော့ Algorithm statement တွေ ရေးသားတည်ဆောက် နည်း Flowchart တွေ ရေးဆွဲနည်းကို သေသေချာချာ လေ့လာကြည့်ပါ။ အလွန် အသုံးဝင်တဲ့ System Development Method ဖြစ်ပါတယ်။ အလွယ်တကူ တတ်မြောက်နိုင်ကြလိမ့်မယ်လို့ ယုံကြည်ပါတယ်။

ကျော်ဇူးတော်လေး (ကသာ)



## 1.6 Requirement Analysis and Design Method

System Development မှန်စုစု၍ project တစ်ခုကို စတင်ရေးဆွဲတော့မယ်ဆိုရင် အဘီ Development ဖုံးပါ၍ project အတွက် analysis requirement များဖို့ လိုအပ်မှာပါ။ ဒီတော့ ပုံစံပြဿနာတဲ့ typical analysis and design method ဆိုရင် ခွဲ့ခြားပြီး စနစ်တကျ လေ့လာကြတောင်း။ ကဲ ဒီတော့ Requirement Analysis and Design method ဆိုရင် စနစ်တကျ လေ့လာကြမယ်ဆိုရင် ဒီအတိုင်း တွေ့ရမှာပါ။

1. Requirement Analysis and Design Method
  - 1.1 Diagramming Methods
  - 1.2 Analysis / Design Diagramming

ကဲ ဒီလိုတွေ့ရပြီနောက်။ ဒီတော့ 1.1 Diagramming Methods ကို ထပ်မံပြီး အပ်စွာ လေ့လာကြည့်ရအောင်။

### 1.1 Diagramming Methods

- 1.1.1 DFD (Data Flow Diagram)
- 1.1.2 ERD (Entity Relationship Diagram)
- 1.1.3 STD (State Transition Diagram)
- 1.1.4 UML (Unified Modeling Language)

ကဲ ဒီတော့ Diagramming Methods အောက်မှာရှိတဲ့ ခေါင်းစဉ် တစ်ခုချင်းစီကို စတင်ပြီးတော့ လေ့လာကြပါနို့။

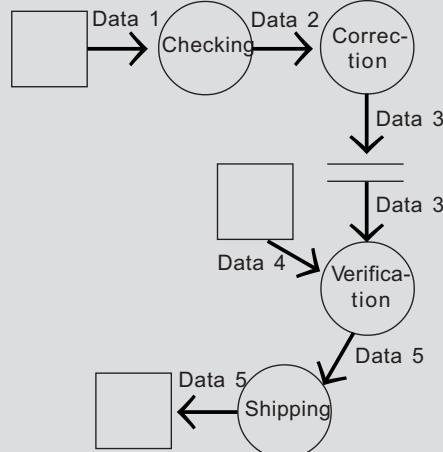
#### 1.1.1 DFD (Data Flow Diagram)

DFD Method ကို စီးပွားရေးလုပ်ငန်းစဉ်တွေမှာ visually to aid understanding အောင် ရေးဆွဲတဲ့အခါမှာ အသုံးပြုတဲ့ analysis method ဖြစ်ပါတယ်။ အောက်မှာ ရေးဆွဲပြဿနာတဲ့ Figure 1.14 မှာ DFD နဲ့ Sample Flowchart တို့၏ လေ့လာကြည့်ပါ။

Figure 1-14 DFD

Symbol	Name	Meaning
→	Data flow (An arrow with a name)	The flow of data is indicated
○	Process (Bubble)	Processing/ conversion of data is indicated
==	Data storage (Two parallel lines)	Stored data are indicated (Files, databases and others)
□	External (A square)	A source or destination of data is indicated (Ordinarily, persons or organizations outside of analysis)

An image of DFD

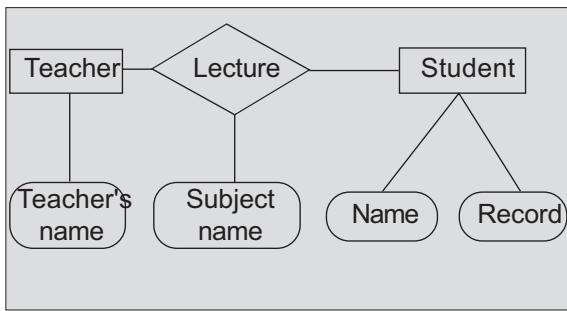


#### 1.1.2 ERD (Entity Relationship Diagram)

ERD ဆိုတာကတော့ Data models တွေကို objects concepts တွေအလိုက် ညွှန်ပြပေးတာပဲ ဖြစ်ပါတယ်။ Business

IT လောကမှာတော့ ERD method ဟို business operation model လို့လည်း ခေါ်ပေါ်ကြပါတယ်။ ဘာမြတ်နဲ့ဆိုတော့ business operation model တွေမှာ ERD method ကိုပါ

အများဆုံး ပုံစံလုပ်လိုပါပဲ။ ကဲ Figure 1.15 ရေးဆွဲပြထား တဲ့ ERD sample Flow Diagram ဖော်လောက်ည့်ပါ။



ကဲ RED sample Flow Diagram ဖော်လောကြီးတဲ့အခါမှာတော့ သူ့ကိုဆွဲသားစဉ်မှာ အသုံးပြုရတဲ့ Symbols တွေကို လောက်ည့်ပါ။

• Entity	Entity ကို ရေးဆွဲတဲ့အခါမှာ Rectangle symbols ဖို့ အသုံးပြုပြီး ရေးဆွဲရပါတယ်။ Entity ဟာ targeted for management ပိုင်းကို indicates လုပ်ပါတယ်။
• Relationship	Relationship တော့ diamond symbols ဖို့ ရေးဆွဲပါတယ်။ Relationship Entity နှစ်ခုကြားမှာ ဆက်သွယ်ပေးတဲ့အရာ အဖြစ် Declare လုပ်ပြီး အသုံးပြုပါတယ်။
• Attribute	Attribute ဟာ Entity နဲ့ Relationship တို့၏ Characteristics and properties တွေကို indicates လုပ်ပါတယ်။

ကဲ ဒါဆိုရင်တော့ ERD ၊ Symbols တွေကို အမော်ခံတော့ နားလည်သွားပြီနော်။ ဒီတော့ အပေါ်က Figure 1.15 ရေးဆွဲပြခဲ့တဲ့ Sample Flow Diagram ဖော်အသေးစိတ် လောက်ည့်ရအောင်။

- Teacher(ဆရာ) ဆိုတဲ့ Entity နဲ့ Student ကျောင်းသား) ဆိုတဲ့ Entity ဟာ lecture(စာသင်ချိန်) ဆိုတဲ့ Relationship ရဲ့ Connection လုပ်ပေးမှုကြောင့်သာ တွေ့ဆုံရတာပါ။
- Teacher ဆိုတဲ့ Entity မှာ Teacher's name (ဆရာ၊ အမည်) ဆိုတဲ့ characteristics ကို လာပြေးနိုင်ဖို့ Attribute ကို အသုံးပြု အောင်ပြုပါတယ်။

- Students ဆိုတဲ့ Entity မှာလ Characteristic and properties တွေကို လာပြေးနိုင်ဖို့ Name and Record ဆိုတဲ့ Attributes ကို အသုံးပြုရပါတယ်။
- Lecture(စာသင်ချိန်မှုလ) သင်ကြားမယ့် ဘာသာရပ် Subject Name ကို လာပြေးနိုင်ဖို့ Relationship ကို လာပြုလေးရပါတယ်။

Relationship လုပ်တဲ့အခါမှာ (၃) မျိုး ကဲပြား မြားနားနိုင်ပါတယ်။ အဲဒါတွေကတော့

- “1 to 1” relationship
- “1 to N” relationship
- “N to N” relationship

Note : N ဆိုတာကတော့ integer ကိန်းဖြစ်ပြီး 1 ထက်ကြီးတဲ့ type တွေကို ဆိုလိုပါတယ်။

ကဲ ဒီတော့ ဥပမာအနေနဲ့ ရေးပြရရင်တော့ “ဆရာတစ်ယောက်ဟာ lecture ချိန်တွေကို တက်ရောက်ခြင်းအားဖြင့် ဆရာတစ်ယောက်မက တွေ့ဆုံးတော့ ဘာသာရပ်အမြောက်အမြားကို သင်ယူနိုင်ပါတယ်။”

ဒီပိုစိတ်ဟာ “N to N” relationship တို့ Example အနေနဲ့ ရှင်းပြတာပါ။ ကဲ STD method ဟာ ဆက်လက်လောက်ပါတယ်။

### 1.1.3 State Transition Diagram

STD method မှာ တစ်ချိန်တိုးကတော့ Object-oriented-modeling မှာ အလွန်အရေးပါပြီး နာမည်ကြီးခဲ့ပေမယ့် ဒီနေ့စောင်မှာတော့ STD method ဟာ သိပ်ပြီးတော့ အသုံးမပြုကြတော့ပါဘူး။ ဒါကြောင့် အသေးစိတ် အကျယ်ချွဲပြီးတော့ မရှင်းပြတော့ပါဘူး။ Diagram Example လိုပဲ လောက်ည့်ပါ။

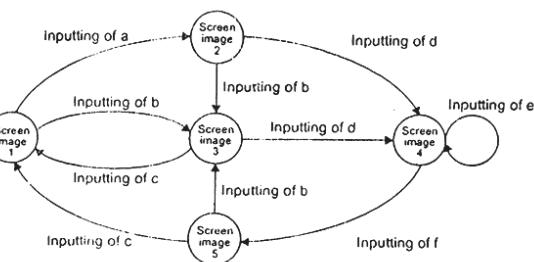
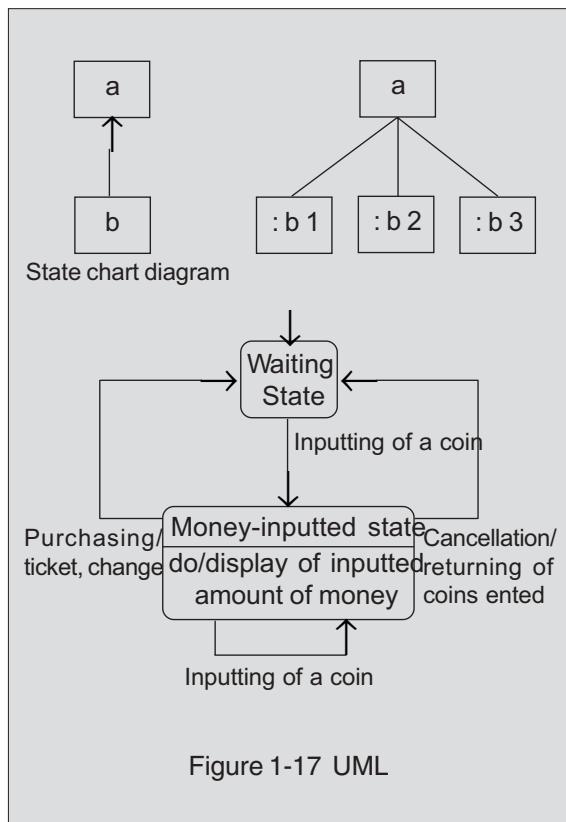


Figure 1-16 A State transition diagram example

#### 1.1.4 UML (Unified Modeling Language)

UML Diagramming method<sup>ဗျာများ</sup> ဒီနေ့ကတ် object oriented concept<sup>ဗျာများ</sup> analysis<sup>လုပ်ရာမှာ</sup> အလွန်အရေးပါတဲ့ စံအဖြစ် (Standard)<sup>လာတ်မှတ်ပြုခြင်း</sup>တော့ အသုံးပြုရတဲ့ diagramming method<sup>ပါ</sup> ဒါကြောင့် နောက်ပိုင်းမှာ Java ရဲ့ Listing<sup>တွေနဲ့အတူ</sup> ရေးဆွဲပုံစံပြုခြင်း သင်ကြားချုပ်ပါမယ်။ အသေးစိတ် ရှင်းလင်းပြောပါမယ်။ အခုတော့ introduce<sup>လုပ်တဲ့</sup> အနေနဲ့ Diagram Example<sup>ပိုပဲ</sup> လေ့လာကြည့်ပါ။



ကဲ ဒါဆိုရင်တော့ Requirement Analysis and Design method<sup>ပိုတဲ့</sup> ခေါင်းစဉ်အောက်မှာ ခွဲဖြာကျနေတဲ့ အပ်စုနှစ်စုအနက်က ပထမဆုပ်စုဖြစ်တဲ့ Diagramming Methods အနကြောင်းကိုတော့ introduce<sup>လုပ်ပြုခြင်း</sup>တော့ ရှင်းလင်းပြုး သွားပါပြီ။ ဒီတော့ Requirement Analysis and Design method<sup>ပိုတဲ့</sup> ခေါင်းစဉ်အောက်မှာရှိတဲ့ ခုတိယအပ်စုဖြစ်တဲ့ 1.2 Analysis/Design Diagramming<sup>ဒါ</sup> စုတင်လေ့လာကြပါစိုး။

#### 1.2 Analysis/Design Diagramming

Program <OR> Software Project<sup>လုပ်ချက်</sup> ဖုန်းစာ ဖုန်းတီးရေးသားတဲ့အခါမှာ လိုအပ်ချက်အရှင်းအမြစ်ကို ရှာဖွေတဲ့အခါမှာ

ဖြစ်စေ Program Process<sup>လွှာကို end user support</sup> အောင် ဖုန်းတီးရေးတဲ့အခါမှာဖြစ်စေ Flowchart Method<sup>လာ</sup> the most typical diagramming method<sup>ပါ</sup> သူကို ရေးဆွဲတဲ့ standard symbols<sup>ဗျာများ</sup> အသုံးပြုခြင်း logical structure<sup>တွေနဲ့ နှုန်းမှန် လမ်းမှန် ရေးဆွဲရပါတယ်။ ကျွန်တော့ အနေနဲ့ကတော့ Java Language<sup>လို</sup> စတင် လေ့လာတဲ့ Programmer/Developer<sup>လုပ်ယောက်အနေနဲ့</sup> အလွယ်တကူ နားလည်း သဘောပေါက်သွားအောင် Program<sup>လွှာ</sup> ဖုန်းတီးရေးသားတဲ့အခါမှာ complex problem<sup>တွေကို processing procedure</sup> အလိုက် သဘောပေါက်အောင် Flowchart<sup>လို</sup> အခန်းကဏ္ဍအနေနဲ့ပါ ရေးသားတွေပြောပါမယ်။ အခြေခံ Flowchart<sup>လို</sup> ဆွဲမှု type<sup>တွေကို ရေးဆွဲရပါမယ်။</sup> ရှင်းလင်းပြုးတဲ့အခါမှာတော့ တွက်ချက်ပို့ချုပြုးတဲ့ Listing<sup>လွှာ</sup> အပါအဝင် sample<sup>တွေ</sup> အတူ Flowchart<sup>တွေကို လေ့လာကြည့်တာပေါ့။</sup></sup>

#### Flowcharts

Flowcharts<sup>ရေးဆွဲတယ်ဆိုတာ မိမိ တည်ဆောက်မယ့် program [Complex problem]<sup>]</sup> definition<sup>ကို အရင်သီ</sup> အောင် Anslysisလုပ်ရပါတယ်။ ပြီးတဲ့အခါမှာ program<sup>၊</sup> processing procedure<sup>နဲ့ နားလည်အောင် algorithm statements<sup>၊</sup> solveလုပ်ရပါတယ်။</sup> အဲဒါတွေပြီးမှ Flowchart<sup>လို</sup> တွက် step<sup>ကို ရေးဆွဲည့်ကြားတာပါ။</sup> Flowchart<sup>ဆိုတာ Soft-</sup>ware Engineering<sup>အဆင့်ရဲ့ System Development Methodologies</sup> မှာ အရေးပါတဲ့ Method<sup>တစ်ခု ဖြစ်တာမို့</sup> သေသေချာချာ မှတ်သားလေ့လာကြတာပေါ့။ Java Language<sup>ကို learnလုပ်တဲ့အခါမှာ အထောက်အကူ ဖြစ်အောင်လိုပါပဲ။</sup> စာဖတ်သူအနေနဲ့ system Development<sup>ခန်းကဏ္ဍ [နိုင်ငံတကာ IT Examလောကမှာ မတတ်မဖြစ်တဲ့ အခန်းကဏ္ဍ]</sup> ကို သင်ကြားဖို့ ဆန္ဒရှိတယ်ဆိုရင် ဆက်သွယ် တောင်းဆိုနိုင်ပါတယ်။ ကျွန်တော့အနေနဲ့လည်း လိုအပ်တယ်ဆိုရင် ပို့ချုပ်သွားမှာပါ။ အနာဂတ်မှာတော့ system Development Method<sup>တစ်စိတ်တစ်ခေါ်အားဖြစ်တဲ့ Flowchart<sup>ပိုပဲ</sup> ရေးသားပို့ချုပ်သွားမှာပါ။</sup> Flowchart is the necessary operations of the system Development methods<sup>လုပ်ချက်မှတ်တဲ့ မှတ်ထားပါ။</sup> Flowcharts<sup>တွေကို ရေးဆွဲတဲ့အခါမှာ သူကို standard<sup>သားတဲ့</sup> unified symbols<sup>တွေ ရှိပါတယ်။</sup></sup></sup>

ကျွန်တော့လေ့လာခဲ့သမျှတွေထဲမှာ JIS X 0121 standard အနေနဲ့ ပြဋ္ဌာန်းထားတဲ့ symbol<sup>တွေဟာ အပြည့်စုံဆုံးဖြစ်တာ</sup> ကြောင့် ထည့်သွင်းပေးထားပါတယ်။ ကဲ ဒီတော့ symbols<sup>တွေ</sup> ကို လေ့လာကြည့်ကြတာပေါ့။

Symbol	Name	Descriptions
	Data	Indicates data for which media is not specified.
	Stored data	Indicates data stored in formats suitable for processing (media are no specified)
	Internal storage	Indicate data stored in an internal storage medium.
	Sequential access storage	Indicate only sequentially accessible data, such as those on magnetic tapes, and cassette tapes.
	Direct access storage	Indicates directly accessible data, such as those on magnetic disks, and floppy disks.
	Documents	Indicates data on media human beings can read.
	Manual input	Indicate data on any medium that is inputted by hand-operated means, such as online keyboards, switches, push buttons or bar code.
	Display	Indicates data on any medium that display information used by human beings, such as those on display screens or online indicators.
	Process	Indicates any type of processing functions.
	Predefined process	Indicates processing composed of one or more operations or instructions defined, for example, a subroutine.
	Manual operation	Indicates any operation handled manually.
	Preparation	Indicates the modification of an instruction or a group of instructions that, like the setting-up of switches or the initial setting-up of a routine, affects later operations.
	Decision	Indicates a decision function that, with an input port and output ports only one of which can be selected at a time, selects one output port depending on the result of the conditional evaluation depicted in the symbol.
	Loop limit	Composed of two parts; conditions for termination for termination and a loop name are written in the symbols indicating the start and the end of the loop, respectively.
	Line	Indicate a flow of data or control.
	Communication link	Indicate that data is transferred on a communication line.
	Connector	Indicates an output port to another place on a flowchart or an input port from another place.
	Terminator	Indicates an output port to, or an input port from, an external environment.
	Comment, annotation	Used to add explanations or notes for clarification.
	Ellipsis	Used instead of a line symbol, and indicates that a symbol or a group of symbols are omitted.

Figure 1-13 The symbols used in flowcharts

အပေါ်မှာ ရေးပြခဲ့တဲ့ Symbolsတွေကို အလွတ်ရအောင် လေ့လာထားပါ။ က ဒီတော့ ဒီ symbols တွေကို အသုံးပြုပြီးတော့ Flowcharts အနေနဲ့ ရေးဆွဲမယ်ဆိုရင် လေ့လာရမှာ အပ်စွဲပြီး လေ့လာရမယ့် group item တွေ ရှိပါတယ်။ ဒါကြောင့် လေ့လာကြည့်ရအောင်-

## 1.2 Analysis/Design Diagramming Method

### 1.2.1 Flowcharts

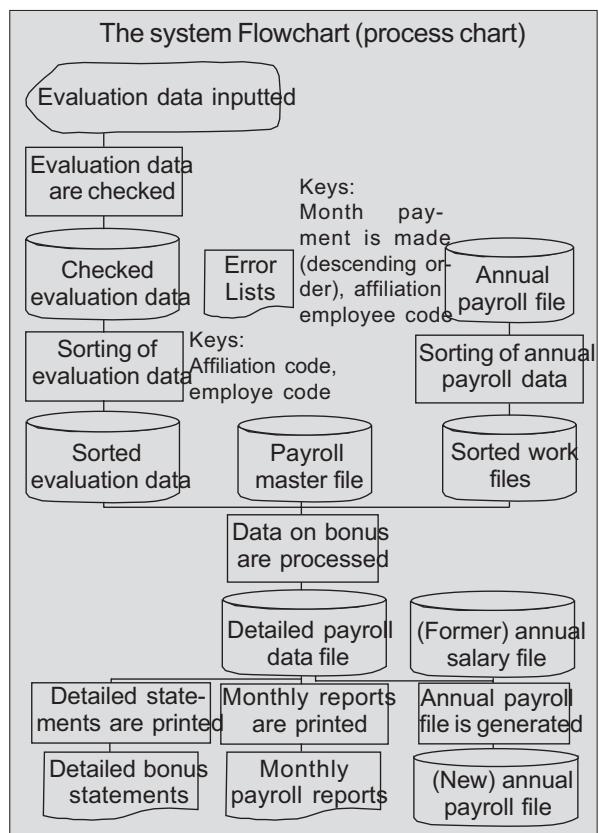
#### 1.2.2 Structured chart တို့မြို့ တွေ၊ ရမှာပါ။ ဒီတော့

1.2.1ကို အပ်စုံ ဆက်ခွဲကြည့်ရင်

### 1.2.1 Flowcharts

#### 1.2.1.1 The system flowchart (process chart)

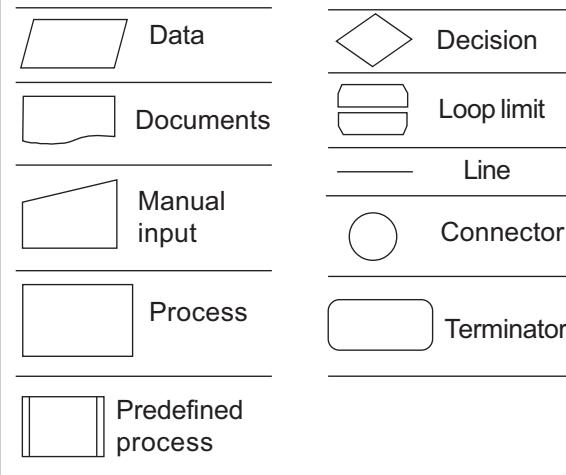
1.2.1.2 The program flowchart တို့မြို့ တွေ၊ ရမှာပါ။ ဒီတော့ program flowchart တို့တဲ့ အပိုင်းပါပဲ။ ဒါပေမယ့် စာဖတ် သူအနေနဲ့ ရှိသက်လေ့လာမယ်ဆိုရင် Family ဖြစ်အောင်လို့ အပေါ်မှာ ရေးပြခဲ့တဲ့ system Flowchart နှင့် program Flowchart တို့ရိုင်းစလုံးကို အရင် ရှင်းပြခဲ့ပါမယ်။ Program Flowchart အပိုင်းကိုတော့ JAVA Laguage တဲ့ တွဲဖက်ပြီးတော့ သေ သေချာချာ သင်ကြားပို့ဆေးထာပါတယ်။ က ဒီတော့ တစ်ရိုင်းစီ ခွဲဗြားပြီး လေ့လာကြည့်ကြရအောင်။



အပေါ်မှာ ရေးဆွဲပြခဲ့တဲ့ System Flowchart တို့မြို့ တွေ၊ ဒီတော့ ဒီ symbols တွေကို အသုံးပြုပြီးတော့ Flowchart အမို့ပုံကို ဖွင့်ဆိုထားပါတယ်။ အလွန်တရာမှ ကျယ်ပြန်လွှာ DataBase System မှာ အစပြုပါ ရေးဆွဲရတဲ့ စနစ်ပါပဲ။ က ဒီတော့ System Flowchart တွေကြောင်းကို နားလည် သဘောပေါက်သွားရင် program Flowchart တွေကြောင်းကို ဆက်ပြီး လေ့လာကြည့်ရအောင်။

### The Program Flowchart

Program Flowchart တို့တာကတော့ အခါ ကျွန်တော်တို့ လေ့လာရမယ့် အမိုက အခန်းကဏ္ဍပါပဲ။ Program Flowchart ဟာ Program တစ်ခုကို တည်ဆောက်မယ့်အခါမှာဖြစ်စေ program တစ်ခု၏ Flow တို့ သိအောင်လုပ်တဲ့အခါမှာဖြစ်စေ အလွန် အရေးပါတဲ့ Method တစ်ခုပဲ ဖြစ်ပါတယ်။ က ဒီတော့ Program Flowchart တို့ sample အနေနဲ့ ရှင်းမပြခင်မှာ Program Flowchart တို့ ဆွဲသားတဲ့အခါမှာ အသုံးပြုပဲ။ Symbols လေးတွေကို အရင် လေ့လာကြည့်ကြရအောင်။ ဒီအခါ Figure 1.18 ရှင်းပြခဲ့တဲ့ Symbols တွေကို အကုန်လေ့လာစရာ မလိုပါဘူး။ စတင် လေ့လာ ရေးဆွဲတဲ့အခါမှာ မှတ်သား အသုံးပြုရတဲ့ Symbols တွေကို ရှင်းပြ ရေးဆွဲပေးထားပါတယ်။ က လေ့လာလိုက်ပါပြီး။ မှတ်မိအောင်ပေါ့။



အခြေခံ program Flowchart တို့ ဆွဲသားတဲ့အခါမှာ အသုံးပြုရမယ့် Symbols ဟာ အခါ (j0) ပဲ ရှိပါတယ်။ က ဒီတော့ Sample Program Flowchart တစ်ခုကို ရေးပြပါမယ်။ ကျွန်တော်တို့ လေ့လာခွဲတဲ့ symbols တွေကိုပဲ အသုံးပြုထားပါတယ်။ ဒီတော့ လေ့လာကြည့်လိုက်ပါပြီး။

Figure 1-20 A program flowchart example

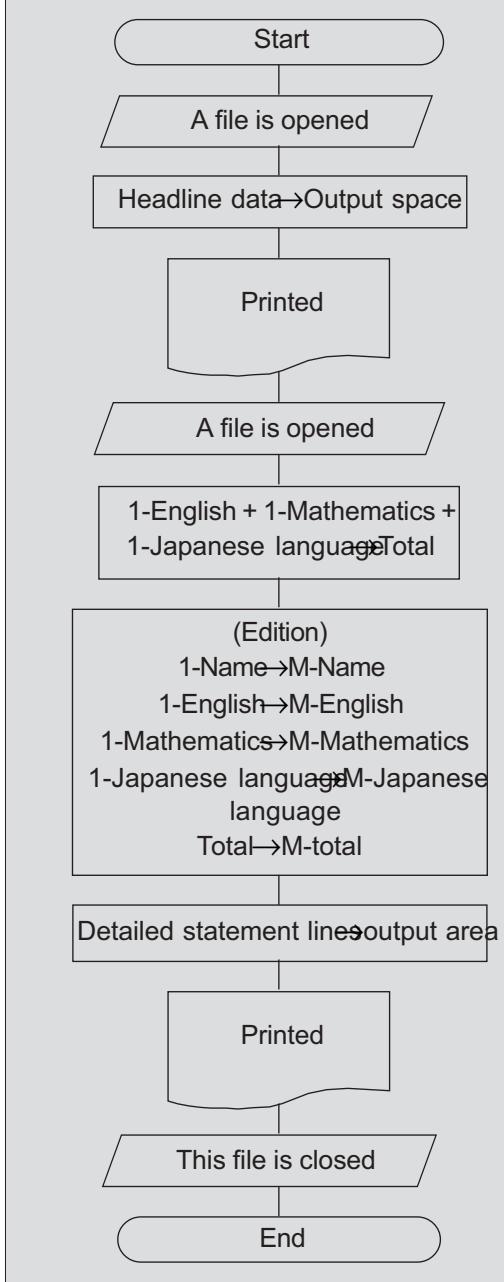


Figure 1.20 ရေးဆွဲပြုခဲ့တာဟာ Sample program Flowchartပါပဲ။ ဒါဆိုရင် symbolsတွေကို မှတ်စီအောင် လေ့လာခိုင်းထားတာကို သဘောပေါက်လောက်ပါပြီ။ ဒါဆိုရင် တော့ Flowchartပါဝေးလက်တွေ့ရေးဆွဲကြည့်ရအောင်။ 1.5 Arrays of JAVAဆိုတဲ့ အခန်းကဏ္ဍမှာ Flowchartရေးဆွဲပြုခဲ့တာကို မှတ်စီမှာ Algorithms statementsတွေ ရေးဆွဲရတာကို မှတ်စီမှာ

ပါ။ ဒီတော့ Algorithm statementsတွေ ရေးမယ်။ ပြီးရင် Flowchartတွေ သက်ဆိုင်ရာ Algorithm statementsအရ ဆွဲသားကြတာပေါ့။

#### Algorithm 1.1 (Sum)

1. Read : FIRST . [input an integer]
2. Read : SECOND. [input next integer]
3. SUM := FIRST + SECOND
4. Write "Sum of two numbers is :", sum.
5. Exit.

Note: Algorithm ကို ရေးတဲ့အခါမှာ Readနေရာမှာ input/writeနေရာမှာ outputကို အသုံးပြုပြီး ရေးသားနိုင်ပါတယ်။

Algorithm 1.1ကို ရေးသား ဖန်တီးထားတာဟာ အလွန် ရှိရင်း စွာနဲ့ ရေးဖွဲ့ထားပါတယ်။ ဒီအခါမှာ စာဖတ်သူအနေနဲ့ 1.5 Arrays of Javaဆိုတဲ့ အခန်းကဏ္ဍမှာ ရေးသားခဲ့တဲ့ typeနဲ့ ပုံစံ အနည်းငယ် ကွဲပြားတာကြောင့် အနည်းယယ် complexဖြစ်ကောင်း ဖြစ်နေနိုင်ပါလိမ့်မယ်။ နောက်စိုင်းကိုတော့ familyဖြစ်ပြီးတော့ ဘယ်လို့ statement formပုံပြုဖြစ်ဖြစ် နားလည်လာမှပါ။ ကဲ ဒါဆိုရင် Alogrithm statementsတွေကို လေ့လာပြီးပြုဆိုရင် တော့ စတင်ပြီးတော့ Flowchartဟို ကိုယ်တိုင် ရေးဆွဲကြည့်ကဲတာပေါ့။ ကဲ စတင်ပြီး လေ့လာကြပါမို့။ ရေးဆွဲ ပြထားတဲ့ Flowchartကို သေသေချာချာ လေ့လာကြည့်ပါ။ Figure 1.21အနေနဲ့ ရေးဆွဲပြထားပါတယ်။

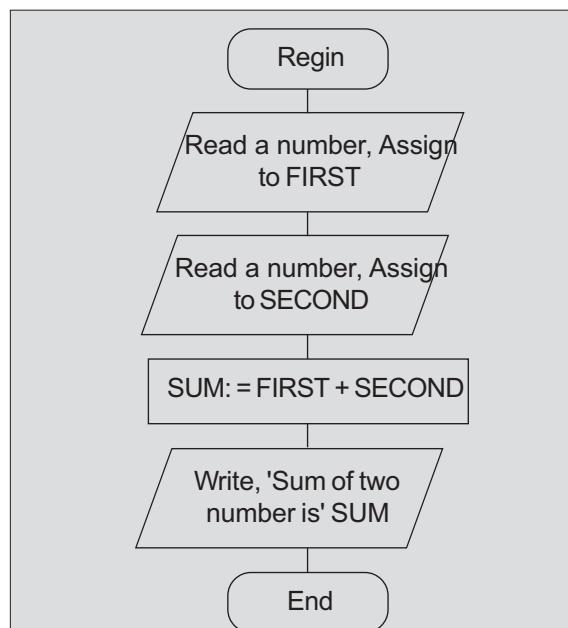


Figure 1.21 Described Algorithm statements in flowchart

Note : Flowchart ဟို ရေးဆွဲတဲ့အခါမှာ Terminator ဟာ Flowchart ၏ အစနဲ့ အဆုံးမှာပါ ပါဝင်ပါတယ်။ Flowchart ၏ အစမှာ ရေးဆွဲထားတဲ့ Terminator ဟာ start/Begin သဖြင့် သုံးသလို အဆုံး Terminator ဟိုတော့ End လို့ အသုံးပြု ရေးဆွဲပါတယ်။

ကဲ အပေါ်မှာ ရေးဆွဲပြဲခဲ့တဲ့ Basic Flowchart ဟို နားလည်သွားရင် အခြေခံ program structure ၏ အသုံးပြုရမယ့် structure type တွေနဲ့ ဆွဲသွားပြီး လေ့လာကြတာပေါ့။ ဒါတော့ if condition ၏ Flowchart ဆွဲသားနည်းကနေ စတင်ပြီးတော့ လေ့လာကြရအောင်ပါ။

#### If condition (single Alternative)

If condition ဟို single alternative ဖို့လဲ ခေါ်ပေါ်ကြပါတယ်။ 1.2 decision, control structure and looping statement ဘာစတင်ပြီးတော့ ပို့ချွဲတဲ့ Structure type တွေကို လေ့လာခဲ့မယ်ဆိုရင် အလွယ်တကူ နားလည်နိုင်မှာပါ။ ကဲ လေ့လာကြည့်ပါဦး။

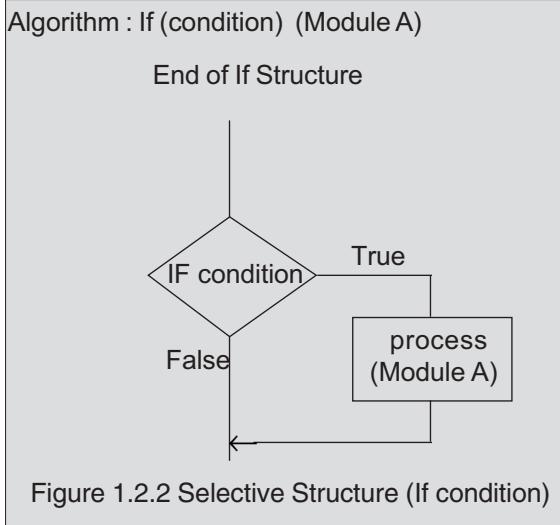


Figure 1.2.2 Selective Structure (If condition)

program statement တွေက selective structure ၏ ခေါ်တဲ့ if condition ပါဝင်စေဖို့ ရေးဆွဲရတဲ့ Method ကို နားလည်သွားရင် Double Alternative ဖို့ခေါ်တဲ့ if - Else condition ကို လေ့လာကြည့်ရအောင်။

#### Double Alternative

##### Selective structure (if - Else type)

Selective Structure ၏ ပုံစံကတော့ ဟုတ်ခဲ့ရင် [Processing A] မဟုတ်ခဲ့ရင် [Processing B] ဖုံးမယ်ဆိုတဲ့ ပုံစပါပဲ။ ကဲ ဒီတော့ အဒီ type ပုံစံကို လေ့လာကြည့်ရအောင်။

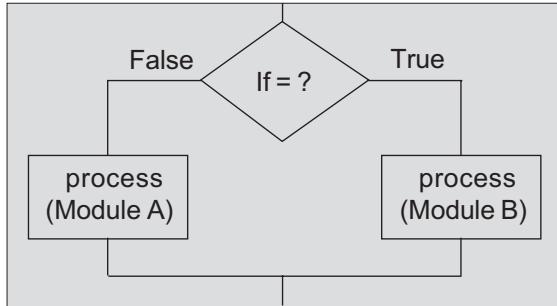


Figure 1.2.3 The Selective Structure

#### Repetitive Structure [Looping Structure]

Repetitive structure ၏ လေ့လာမယ်ဆိုရင်တော့ အောက်ပါအတိုင်း အပိုစွဲပြီး လေ့လာလို့ ရပါတယ်။ အဲဒါတွေကတော့

While loop [indeterminate loops] မှန်းဆ၍ မရသော looping များ

- Do - while type, until type [indeterminate loops] မှန်းဆ၍ မရသော looping များ
- For structure [Determinate loops] မှန်းဆ၍ ရသော Looping များ

စသည်တို့ဖြစ်ပါတယ်။ ဘေးမှာ square bracket ၏ ရေးထားတာတွေကတော့ အခြား အခေါ်အဝေါ်တွေပါပဲ။ ကဲ ခေါင်းစဉ်တစ်ခုချင်းစီ လေ့လာကြည့်ရအောင်။ Sample\_Flow\_type တွေကို သေသေချာချာ လေ့လာပါ။ Sample type တွေကို နားလည်သောပေါက်မှာသာ program နဲ့ လက်တွေ၊ ရေးဆွဲတဲ့အခါမှာ အမှန်တကယ် ရေးဆွဲနိုင်မှာပါ။

#### Looping Structure for while structure type (Repeat while structure)

While structure ဟို လက်တွေ၊ အနေနဲ့ Program တွေ ရေးပြီးတော့ ရှင်းလင်းပြုခဲ့ရုံးသာမကပဲ control statement အနေးကဏ္ဍမှာလ သေသေချာချာ ရှင်းပြုပြီးပါပြီ။ အခါ Flowchart အနေးကဏ္ဍအနေနဲ့ ထုတ်ပြုး ရှင်းပြုရင်တော့ -

- condition ကို checkလုပ်ပါတယ်။
- condition သည် true ဖြစ်နေသရွေ့ instruction ကို executeလုပ်ပါတယ်။
- ပြီးတဲ့အခါမှာ entire procedure ကို Repeat လုပ်ပါတယ်။
- အကယ်၍ condition သည် false ဖြစ်သွားသည်နှင့် တစ်ပြိုင်နက် Looping ဟာ End ဖြစ်သွားပြီး
- While loop ရဲ့ အောက်မှာရှိတဲ့ program ၏ အခြား statements တွေကို အလုပ်ဆက်လုပ်ပါတယ်။

က ဒီတော့ အောက်မှာရေးထားတဲ့ while loop ရဲ့ sample type Flowchart ဖို့ လေ့လာကြည့်ပါ။

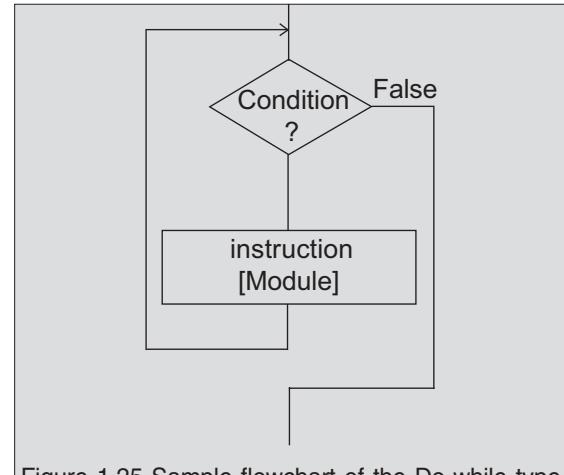
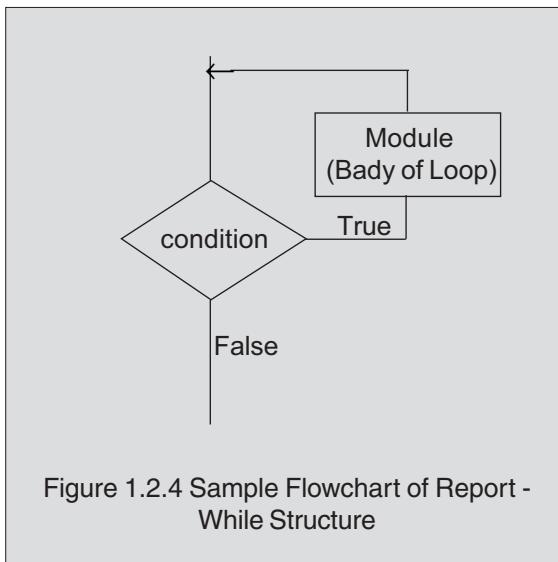


Figure 1.25 Sample flowchart of the Do-while type

Note : while နဲ့ Do - while type တို့၏ Flowchart များ ဆင်တူတာကြောင့် နှစ်သက်ရာ type ရေးဆွဲ ရေးသားနိုင်ပါတယ်။ ကွဲပြားခြားနားတာကိုတော့ မှတ်မှုနားလည်အောင် လေ့လာထားပါ။

#### Until Structure

Until Structure မှာ flow type ဟာ While, Do - While type စသည်တို့၏ type ပုံစံနဲ့ လုံးဝ ကွဲပြားခြားနားပါတယ်။ က သေသေချာချာ လေ့လာကြည့်ရအောင်။

- Condition ကို checkလုပ်တဲ့ အခါမှာ false ဖြစ်နေ သရေး၊ အပေါ်က processing Module ဖို့ ပြန်ရောက် လာပြီး အလုပ်လာလုပ်နေမှာပါ။
- အကယ်၍ တစ်ချိန်ချိန်မှာ condition သည် true ဖြစ်ခဲ့ ရင်တော့ looping ကနေ ခုနှစ်ကိုပြီးတော့ program ရဲ့ အခြား statement တွေကို အလုပ်လာပြီး ဆက်လုပ် တော့မှာပါ။ Sample Flow Diagram ဖို့ လေ့လာကြည့်ပါ။

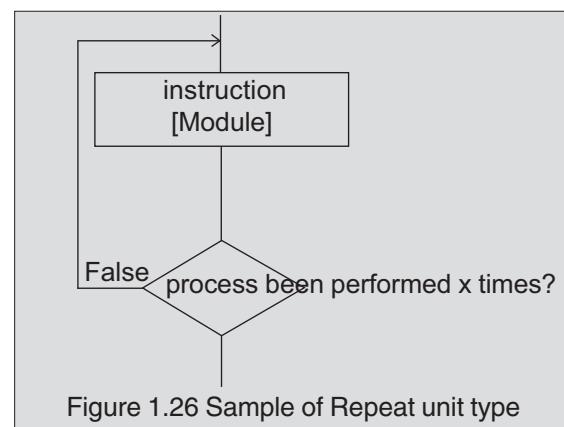


Figure 1.26 Sample of Repeat unit type

က ဒီ Sample Flow Diagram လေးကို လေ့လာကြည့်  
ရင်တော့ if <ဖြစ်ခဲ့လျှင် > break ဆိုတဲ့ program statement  
လေးကို သွားသတိရမိမှာပါ။ ကဲခါဆိုရင်တော့ until structure  
type ရဲ့ flowchart ရေးဆွဲပုံကို နားလည် သဘောပေါက်သွား  
လောက်ပါပြီ။

Note : Flowchart ရေးဆွဲ Trueနေရာမှာ <Yes>နဲ့ False  
နေရာမှာ <No>ဆိုတော့တွေကို အစားထိုး သုံးစွဲနိုင်ပါတယ်။ အတူ  
တူပါပဲ။ ခွင့်ပြုနိုင်ပါတယ်။

#### For Structure type

For looping structure ကြောင်းကိုတော့ အသေးစိတ်  
အသုံးပြုရာတဲ့ increment operator (+) Decrement op-  
erator (-)တွေကနေ အစပြုပြီးတော့ ရေးသား ရှင်းပြခဲ့တာ  
ဖြစ်တာမို့ အားလုံးတို့ နားလည် သဘောပေါက်ပြီးသား ဖြစ်နေမှာ  
ပါ။ ဒါကြောင့် Basic Sample Flowchart \_form ပါပဲ လေ့  
လာကြည့်ပါ။

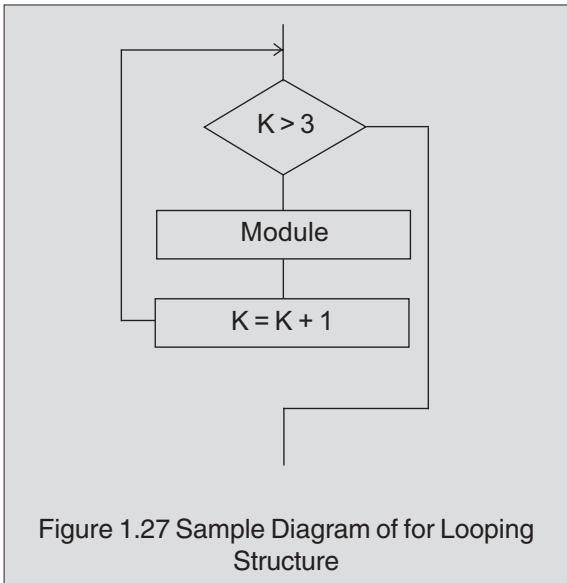


Figure 1.27 Sample Diagram of for Looping Structure

က ဒါကို လေ့လာကြည့်ရင် looping Structure ရဲ့ Flow\_  
type တွေဟာ ဆင်တူတာမို့ အလွယ်တကူ နားလည်သော  
ပေါက်သွားမှာပါ။ သေသေချာချာ လေ့လာထားပါ။

#### Multi-branch structure <CASE>

Multi-branch structure <CASE> ကြောင်းကတော့  
programming Basic မှာတုန်းက Sample တွေနဲ့ ရေးသား  
ရှင်းပြခဲ့တာမို့ နားလည်သောပေါက်ပြီးသား နေမှာပါ။ လိုအပ်  
ရင် Lisiting 1.6ကို ပြန်လှန် လေ့လာနိုင်ပါတယ်။

#### Case ကို Form ဟာ

- Function ထဲက ဖြစ်စေ
  - Multi-type results ထွေထဲက ဖြစ်စေ
  - define check လုပ်ပြီး ရွေးချယ်မှု လုပ်တာပါပဲ။
- Sample flowchart ကို လေ့လာကြည့်ပါ။

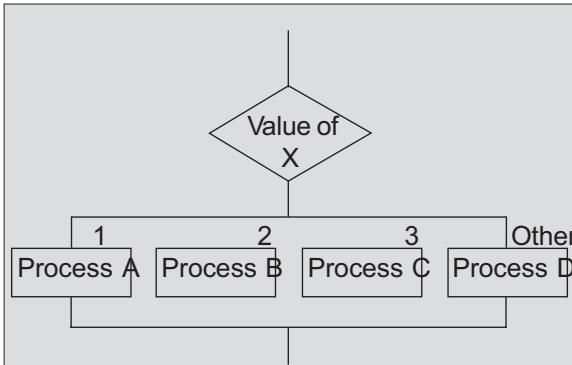


Figure 1.28 Sample flowchart of Multi-branch (case) structure

ကျွန်တော် အခါ ရွှေမှာ သင်ကြားပို့ချဲခဲ့တဲ့ အချက်အလက်  
တွေဟာ program Flowchart ဟဲ ရေးဆွဲရင်း အသုံးပြုရမယ့်  
symbols ထွေ ဆွဲသားနည်း Methods ပါပဲ။ ဒါတွေကို အရင်  
သေသေချာချာ လေ့လာသောပေါက်ထားမှာသာလျှင် Lisiting  
တွေနဲ့ ယဉ်တွဲလေ့လာတဲ့ အခါမှာ လွယ်လွယ်ကူကူ သင်ယူနိုင်မှာ  
ဖြစ်ပါတယ်။

ဒီအခါမှာ ကျွန်တော်အနေနဲ့ ရှင်းပြစ်ရနိုတာကတော့ 1.2 Analysis/ Design Diagramming Method အပ်စွဲပြီး  
လေ့လာခဲ့စဉ်က Sub-title 1.2.1 Flowchart တဲ့ ခေါင်းစဉ်နဲ့  
1.2.2 Structure Chart ပြီး အပ်စုနစ်စွဲပြီး တွေ့ခဲ့ရတာဆို  
တော့ Lisiting တွေနဲ့ မရေးပြခင်မှာ Structure Chart ဟို  
အရင်ဆုံး လေ့လာကြတာပေါ့။ ဒါပေမယ့် ကျွန်တော် ဆက်လက်  
ပို့ချာမယ့်အပိုင်းမှာ Structure Chart ဟို introduce အနေနဲ့  
ရှင်းပြမှာကိုတော့ သတိပြုမိစေချင်ပါတယ်။ ဒါကို introduce  
လုပ်ပြီးမှာသာလျှင် program Flowchart ကြောင်းကို ဆက်လက်  
လေ့လာကြတာပေါ့။

#### 1.2.2 Structure Charts

Flowchart ဆိုတာကတော့ programming theory  
အတိုင်း ဖြစ်နိုင်စွမ်းတဲ့ algorithm နဲ့ base လုပ်ပြီးတဲ့ အခါမှာ  
ဖြေရှင်းရတာပါ။ ဒါပေမယ့် Flowchart ကို ရေးဆွဲတဲ့ အခါမှာ  
arrow symbol ဟို အသုံးပြု ရေးဆွဲရပါတယ်။ Structure chart  
ဆိုတာကလည်း တစ်နည်းပြောရရှင်တော့ Flowchart Method  
နဲ့ တစ်စိတ်တစ်အောင်ပါပဲ။ ဒါပေမယ့် Structure chart ဟို

ရေးဆွဲတဲ့အခါမှာ arrow symbol ဟို အသုံးပြုခွင့် မရှိပါဘူး။ ပိုပြီးရှင်းလင်းအောင်、programme တွေ နားလည်အောင် ပြောရ ရင်တော့ procedure language အသုံးပြုတဲ့ type အတိုင်းပဲ structure theorem ရဲ့ possible ဖြစ်တာကိုပဲ ရေးဆွဲ ဖြေရှင်းရတာပါပဲ။ Flowchart ဆိုတာကတော့ non-structural algorithm ဖြစ်တယ်ဆိုတာကိုလည်း မမေ့ပါနဲ့။

Note : GOTO statements သာ Java Language မပါဝင်ပါဘူး။ ဒါကို သတိပြုမှတ်သာစိုးပါပဲ။

ဒီအချက်တွေဟာ Flowchart နဲ့ structure chart တို့၏ ကွဲပြားမြားနားမှပါပဲ။ structure chart ဟို စနစ်တကျ အပ်စွဲပြီး လေ့လာတဲ့အခါမှာ Methodologies ခဲ့အပ်စွဲပြုးမြား မှတွေ ရှိလာပါတယ်။ အဲဒါတွေကတော့ အောက်မှာ စနစ်တကျ Group\_type ဖြောထားပါတယ်။ လေ့လာကြည့်ပါ။

\* NS chart (Nassi - shreiderman Chart)

\* PAD chart (Problem Analysis Diagram) with tree structure

\* SPD chart (Structured Program Diagram Chart)

\* HCP chart (Hierarchical and Compact description chart)

\* YAC II chart (Yet Another Control chart II)

က ဒီတော့ ခေါင်းစဉ်ပေါ်လေးတွေကို တစ်ခုချင်းစီအလိုက် Sample Diagram ထောင်းတွေနဲ့ ရှင်းပြုပါမယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

\* NS Chart (Nassi - Shreiderman chart)

NS Chart တိုတာကတော့ Special Symbol တွေကို အသုံးပြုပြီးတော့ သာမန် Control Structure တွေအတိုင်း စိစဉ် ရေးဆွဲတာပဲ ဖြစ်ပါတယ်။ symbols ထောင်းကွဲပြားမြားနားတာကလွှဲလို့

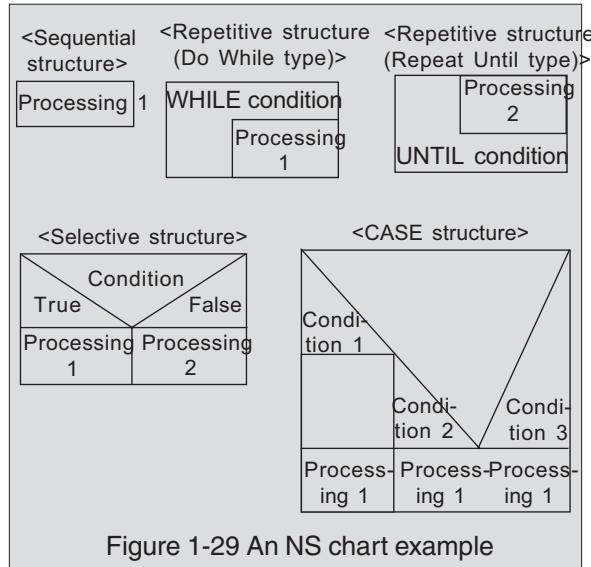


Figure 1-29 An NS chart example

ဒီ Structure charts တို့တွေဟာ algorithms ခဲ့ပါ၊ program Flowchart အတူတူပါပဲ။ ဒီတော့ အထွေအတူးရှင်းပဲစရာ မလိုဘူးပေါ့နော်။ အောက်မှာ NS chart example ရေးဆွဲပြထားပါတယ်။ လေ့လာကြည့်ပါ။

\* PAD chart (Problem Analysis Diagram) with tree structure

PAD chart တိုတာကတော့ Logical Structure ဖြေရှင်းတဲ့ အခါမှာ problem - oriented data structure ဖြေရှင်းတာပါပဲ။ ဒါကြောင့်မို့ ခေါင်းစဉ်ရဲ့ ဘေးမှာ with tree structure တို့ ရေးသား ရှင်းပြထားပါပဲ။ Sample Diagram အနေနဲ့ PAD chart တို့ Structure chart type ပုံစံကို ရေးဆွဲပြထားပါတယ်။ လေ့လာကြည့်ပါ။

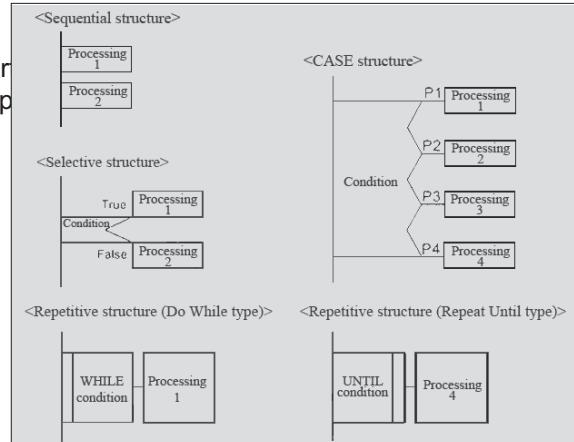


Figure 1-30 A PAD example

\* SPD chart (Structured Programming Diagram)

SPD chart တာလည်း PAD chart တို့ပါပဲ။ သူ့ကို ရေးဆွဲတဲ့အခါမှာ tree structure တိုင်း ရေးဆွဲဖြေရှင်းတာပါပဲ။ သေသေချာချာ လေ့လာကြည့်ပါ။

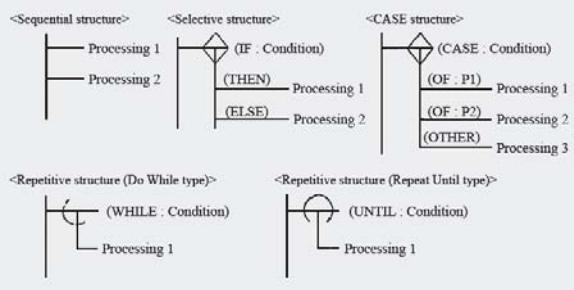


Figure 1-31 An SPD example

\* HCP chart (Hierarchical and Compact description chart)

HCP chartအကြောင်းကို ရှင်းပြရမယ်ဆိုရင်တော့ described of logical structures of Algorithms, in the sam

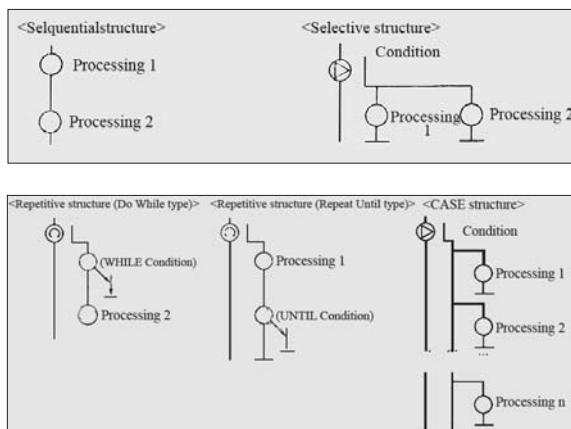


Figure 1-32 A HCP example

way of PAD and SPD၏ ပြောရင်လ ရပါတယ်။ ဘာကြောင် လဲဆိုတော့ HCP chartအနေနဲ့ ရေးဆွဲရတဲ့အဓိကမှာလ true structure typeဆို အသုံးပြုပြီး ရေးဆွဲရတာဖြစ်လိုပါပဲ။ ပုံကို လေ့လာကြည့်ပါ။

\* YAC II chart (YetAnother Control chart II)

YAC II chartဟာလ PAD, SPD and HCP၏လိုပ် tree structureအို အသုံးပြုပြီး ရေးဆွဲတာဖြစ်တဲ့အတွက်

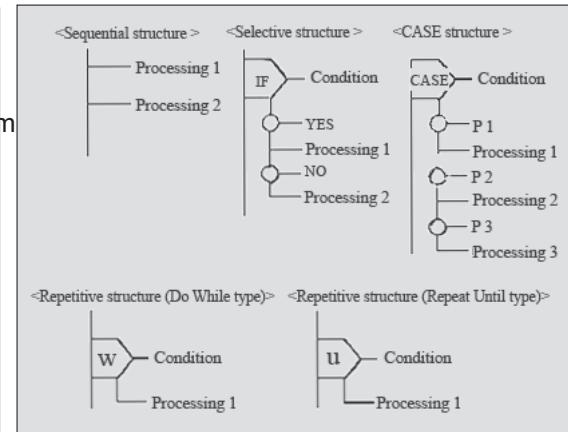


Figure 1-31 A YAC II example

အကျယ်ချုံပြီး မရင်းပြတော့ပါဘူး။ ပုံကိုပဲ သေသေချာချာ လေ့လာကြည့်ပါ။ Figure 1.33အနေနဲ့ ရေးဆွဲပြထားပါတယ်။

Flowchartနဲ့ structured chartအကြောင်းတွေကို ရေးသား ရှင်းပြုပြီးတဲ့အဓိကမှာတော့ Decision Tableဆိုတဲ့ အခန်းကဏ္ဍတို့ ချုပ်လုပ်ထားခဲ့ပါတော့မယ်။ Java Language သင့်ယူတဲ့အဓိကမှာ အထောက်အကြဖြစ်မယ့် Flowchartsအကြောင်း (ပိုပြီး တိတိကျကျ ပြောရရင်တော့ Flowchartဆိုတဲ့ ခေါင်းစဉ်အောက်မှ System Flowchart Program Flowchartဆိုပြီး အပိုင်းနှစ်ပိုင်းကဲဖော့အထဲမှ Program Flowchartအကြောင်း ကိုပဲ စနစ်တကျ Exampleတွေနဲ့အတူ ရေးသား ရှင်းပြမှာပါ။ က ဒီတော့ Example Programတွေနဲ့အတူ Flowchartတွေကို လက်တွေ့ရေးဆွဲကြပါမို့။

ကျော်ဖော်ပေး (ကသာ)

### မိနစ်ပိုင်းအတွင်း အားပြည့်စေမည့် ဘက်ထရီ

တစ်မိနစ်အတွင်းမှာပဲ ဘက်ထရီ၍ ၁၀ ရာခိုင်နှုန်းကို အားပြန်ပြည့်စေတဲ့ လစ်သီယံအိုင်းယုန်းဆုပါဘက်ထရီတို့မျိုးကို တိရိုက်ဘကေနေ ထုတ်လုပ်တော့မယ်လို့ ကြော်လားပါတယ်။ အဲဒါ ဘက်ထရီဟာ အီလက်ထရီတို့တွေကို အားပြည့်နေစဉ်အတွင်းမှာ တစ်ခုနဲ့တစ်ခု 100 nanometer ဟာ လစ်သီယံအိုင်းယုန်းတွေအကြေားမှာ အမြန်ဆုံးနှုန်းနဲ့ ကူးပြောင်းစေတယ်လို့ ဆိုပါတယ်။ တကယ်တစ်း ဘက်ထရီတို့လုံး အားပြန်ပြည့်စွဲကတော့ ၁၀ မိနစ်လောက် ကြာမယ်လို့ ဆိုပါတယ်။ ဒီကနေ့စေတုံး ဘက်ထရီတွေဟာ အချေယူအစားအလေးချိန်နဲ့ power sourceအသုံးချိန်မှာတွေ အပေါ်မှာ မှတ်ပြုပြီး အရေးပါလောတော့လို့ ဖော်တော်ဆိုင်ကယ်မှာအ လျှပ်စစ်ဓာတ်ရရှိနိုင်တဲ့ source အမျိုးမျိုးကနေ ပါဝါ ပြန်ဖြည့်နိုင်တဲ့ အခုလိုဆုပါဘက်ထရီမျိုးဟာ အနေဂတ်မှာ စိုးမိုးနေရာယူသွားလိမ့်မယ်လို့ ခန့်မှန်းထားပါတယ်။ လာမယ့်နှစ်များမှာ ရွေးကွက်ထဲကို ရောက်မယ်လို့ သိရပါတယ်။



## 2.2 Objects Concept of Development tools in JAVA

\* Reference on C& C++ programmer

\* Examining the Address, the stack and the Free Store

C++ programmer တွေအတွက်ကတော့ new keyword ကို အသုံးချဖြီး program ကို ရေးသားမဲ့ယင်းရင် pointer type ဖုန်းဖြစ်သွားပြီနော်၊ ဒီအခါ မူလိုက်နည်းစိန်းရင် memory managing data on the free store အနေနဲ့ သွားလှပ်ပါတယ်။ အဒီအခါမှာ ဒီ memory ကို new keyword ကို အသုံးပြုပြီးတော့ manage လုပ်တာတွေကို programmer ကိုယ်တိုင်က ဖျက်သိမ်းပေးရပါတယ်။ အဒီအခါမှာ အသုံးပြုရမယ့် keyword ကတော့ delete keyword ကို useလုပ်ရမယဲ့။ ဘာကြောင့် ဒီအကြောင်းကို ရှင်းပြန်ရသလဲဆိုတော့ စာဖတ်သူဟာ ကျွန်ုပ်တော် ရေးသားပို့ချနေတဲ့ JAVA Developer Guide ကို လေ့လာနေတဲ့အတွက် အနည်းငြားတော့ C&C++ programming language အကြောင်းကို လေ့လာဖူးမှာပါ။ ဒီအခါမှာ new keyword ဟာ pointer type မှာမူ Indirection operator (\*) နဲ့ယုံ့တွဲပြီး အသုံးပြုရမယ့်။ အသစ်ဖန်တီးတဲ့ memory management မှာလည်း အသုံးချဖြီးရင် အသစ်ဖန်တီးလိုက်မိတဲ့ Free Store စာတွေကို programmer ကိုယ်တိုင် delete keyword ကို အသုံးပြုပြီး ပြန်ဖျက်သိမ်းခဲ့ရမှာပါ။ ဒီအခါ မလိုလားအပ်တဲ့ memory leaks ဖြစ်တာတွေလည်း ဖြစ်လာနိုင်ပါတယ်။ Stay pointer အနေနဲ့ ဖန်တီးတဲ့အခါမှာလည်း programmer ကိုယ်တိုင်က မရေးတတ်၊ မကျွမ်းကျင်ရင် crash ဖြစ်သွားနိုင်ပါသေးတယ်။ ဒါတော့ C&C++ programmer သမားတွေ အများဆုံး တွေကြုံရပဲ ဖြေသာရာပါပဲ။ ဒါအပြင် pointer method ကို new keyword နဲ့ ယုံ့တွဲပြီး အလုပ်လုပ်နေစဉ်မှာ တစ်ခုတိုးပြီး [များယွင်းရေတွက်မိတာ] တွေကြောင့် အရင်ဖန်တီးထားတဲ့အရာတွေ ဖျက်စီးတာမျိုးကို C++ programmer တိုင်း တွေကြုံဖူးပါတယ်။ Java မှာတော့ ဒီဖြေသာရာ မရှိတော့ပါဘူး။ C++ language မှာ new & delete keywords ကိုသုံးစွဲတာနဲ့ ပတ်သက်တဲ့ crash ဖြစ်သွားနိုင်တဲ့ အနေ အထားမျိုးကို လက်တွေနားလည်းသောပေါက်အောင် Sample program ထုတေသနလေးနဲ့ ရေးပြထားပါတယ်။ ဘာကြောင့် ဒီလိုဖြစ်ရ

တယ်ဆိုတာကို တတ်နိုင်သလောက် လေ့လာကြည့်ပါ။ ဒါဟာ Java ရဲ့ အားသာချက်နဲ့ C++ ရဲ့ အားနည်းချက်ကို လက်တွေ ကျွဲ့ လေ့လာကြည့်တာ ဖြစ်ပါတယ်။ ကဲ လေ့လာကြည့်လိုက် ပါဦး။

### Sample program in C++ programming language

When you call delete on a pointer, the memory it points to is freed. Calling delete on that pointer again will crash your program! When you delete a pointer, set it to zero (null). Calling delete on a null pointer is guaranteed to be safe.

#### For example:

```
Animal *pDog = new Animals ;
delete pDog; // frees the memory
pDog = 0; //sets pointer to
           null
//.....
delete pDog; // harmless
```

#### Allocating, using, and deleting pointers.

```
1: // Listing 8.4
2: // Allocating and deleting
   a pointer
3:
4: # include < iostream.h>
5: int main ( )
6: {
7: int local Variable = 5;
8: int * plocal = & local Variable;
9: int * pHeap= new int;
10: if (pHeap ==NULL)
11: {
12: cout << "Error! No memory
      for pHeap!!" ;
13: return 0;
14: }
15: * pHeap = 7;
16: cout << " Local Variable :
```

```

    <<Local Variable<<"/n" ;
17: cout << "* plocal:" << *
     plocal <<"/n";
18: cout<< " *pHeap:" <<* pHeap
     << "/n" ;
19: delete pHeap;
20: pHeap = new int;
21: if (pHeap == NULL)
22: {
23: cout << "Error ! No memory
     for pHeap!!";
24: return 0;
25: }
26: * pHeap = 9;
27: cout << "* pHeap: " <<* pHeap <<
     <<*pHeap<<"/n";
28: delete pHeap;
29: return 0;
30: }

```

Java Language ရဲ new keywords ကို အပေါ်က C&C++ language မှာလို့ ပြောသောတွေမဖြစ်စေသေ စိတ်ချွဲလွယ်ကွ္ခာ သုံးစွဲနိုင်ပါတယ်။ ဒါဟာ Java language ရဲ အားသာချက်ပါပဲ။

**Note:** Java ဟာ C&C++ထက်သာပါတယ်။ ပိုကောင်းပါတယ်လို့ ကြားမှာသီဖူးရဲ့ ပြောနေရင် ရှုက်စရာကောင်းပါတယ်။ ဒိုလို compareလုပ်ခြင်းဟာ C++ ကို ပစ်ပယ်နိုင်းတာ မဟုတ်ပါဘူး။ ကျွန်တော်ရဲ့ Computer Science နယ်ပယ်ထဲမှာတွေ့ရတဲ့ အတွေအကြုံအရ C&C++နယ်ပယ်မှာပဲ ကျွန်တော်နေထိုင်ခဲ့ရတယ်ဆိုရင် မမှားပါဘူး။ ယနေ့အချိန်ထိလည်း C++ ကို အသုံးပြုနေလဲပါ။ ကျွန်တဲ့ language တွေဖြစ်တဲ့ VB, BASIC, Ada, COBOL တို့ စတာတွေထက်တောင် C++ ကို familyဖြစ်တယ်လို့ ပြောရင်ရပါတယ်။ Java ကိုစတင်လေ့လာစဉ်ကတော့ Java ဟာ C++ ထက်ဘာများသာပြီး ပိုကောင်းနေလို့လို့တောင်တွေးယူမြှုပါသေးတယ်။ Language တတ်ခဲ့ရဲ့ အကြောင်းကို ထဲထဲဝင်ဝင် မလေ့လာမသင်ယူဘဲနဲ့တော့ Java ဟာ C++ ထက်သာတယ်လို့ ပြောနေရတာဟာ ရှုက်စရာကောင်းပါတယ်။ C++ ရဲ features တွေကိုသာ အမှန်တကယ် တတ်ကျွမ်းမယ်ဆိုရင်လဲ ကိုယ်လိုချင်တဲ့ program တွေကို ဖန်တီးယူလို့ ရနိုင်တာပဲ မဟုတ်လဲး။ အဲဒီလိုပဲ Java ရဲ အားသာချက်တွေကို လက်ခြေးလည်း ကိုယ်ရဲ့ အသုံးချုပ်ပယ်မှာ အဆင်ပြုအောင် အံဝင်အောင်လည်း တည်ဆောက်နိုင်ရမှာပေါ့။ Language တတ်ချုပ်းစီရဲ့ အားနည်းချက်၊ အားသာချက်တွေကို လက်ခံနိုင်ရမှာ ဖြစ်ပါတယ်။

Java Language ကိုလေ့လာနေတဲ့ programmer တိုင်းသိတားရမယ့် အချက်ရှိပါတယ်။ Java မှာလည်း အချို့သော

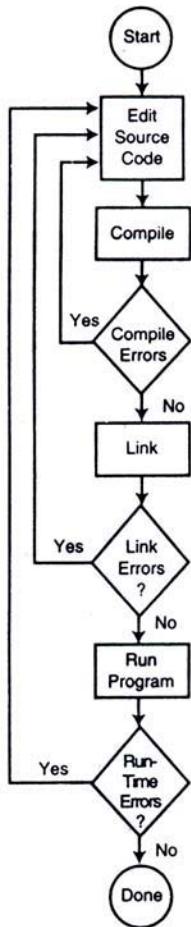
အားနည်းချက်တွေရှိပါတယ်။ C++ နဲ့ Java Language တွေဟာ compile လုပ်ပဲခြင်း မတူညီကြတဲ့ဘူး။ Java Language ဟာ Java Virtual Machine (VJM) ကို အမိုးပြုပြီးတော့ Java ရဲ bytecode တွေကို execute (Run) လုပ်ပေးတာပါ။ ဒီအချက် ကြောင့် Java Language ဟာ အလွန်လျင်မြန်စွာ အလုပ်လုပ် နိုင်သလို တစ်ဖက်မှာလည်း အလွန်နေးကျွေးလေးလွှာ အလုပ်လုပ်ပါတယ်လို့ပြောရင် မမှားပါဘူး။ ဒါကို programmer တိုင်းသိရှိလောက်စံထားရမယ့် အချက်ပါပဲ။

**Note:** Java လို့ Virtual Machine ကို အသုံးပြုပြီးတော့ Run လုပ်နိုင်တဲ့ စွမ်းရည်မြှင့် Language အဖြစ် Microsoft company ကနေ J++ ဆိုတဲ့ Language ကို ဖန်တီးခဲ့ပါတယ်။ ဒါပေမဲ့ အငြင်းမွားမှုတွေ ဖြစ်ဖွားပြီးတဲ့နောက် C# ဆိုတဲ့ Language ကို တိတုတွေခဲ့ပြီး Java ကို အနိုင်ယူလို့ ကြိုးစားတယ်လို့ 1.1 Fundamental of JAVA မှ ကျွန်တော်ရေးသားခဲ့ပါတယ်။ ဒီနေရာမှာ လေ့လာကြည့်ရင် သာမန် user တွေအတို့ Java Language ကို လွယ်လွယ်ကုတ် Install လုပ်ဖို့ program ရေးရှိဟာ သိပ်မလွယ်ကူလှပါဘူး။

Microsoft ရဲ့ C # Language ကတော့ C++ ရဲ့ ပတ်ဝန်းကျင်မှာ ကြိုးပြုပဲခဲ့တဲ့ ကျွန်တော်တို့အနဲ့ အလွယ်တက္က support ဖြစ်နေပါတယ်။ ဒီအချက်ကိုတော့ Sun Microsystem ရဲ့ Java Language ကို ဖန်တီးခဲ့ကြတဲ့ Designer Group အဝါအဝင် CEO အဆုံး လက်ခံကြပါတယ်။ Microsoft ရဲ့ ထုတ်ကုန်တွေဟာ End-user Support ဖြစ်တယ် ဆိုတာကို ကျွန်တော်တို့ ငြင်းလိုမရပါဘူး။ ဒါကြောင့် အချို့နေရာ တွေမှာ Java မှ Java ဖြစ်နေရာက C# ကို ပြောင်းလဲလာတာကို တွေ့လာရပါတယ်။ အချို့နေရာမှာ Net ပေါ့။

ဒါခဲ့ရင် ကျွန်တော် တို့အနေနဲ့ Java Language နဲ့ ကျွန်တော်တို့ ရင်းနှီးကျမ်းဝင်တဲ့ C++ Language နှစ်ခုရဲ့ program Executable flow ကို Diagram လေးရဲ့ လေ့လာကြရအောင်။ C++ Language ကတော့ အားလုံးလိုလို ပိုကောင်းတိုင်း တွေ့ဖူးရင်းနှီးတာကြောင့် [ Java နဲ့လည်း ထပ်တူထပ်မှုနဲ့ပါး နှီးစပ်တာကြောင့် ] Developement Cycle ကို ရေးပြထားပါတယ်။ ကဲ လေ့လာကြည့်ရအောင်။ C++ ရဲ့ program Developement Step ကို Write the program, Compile the Source code, Link the program, and run it လို့ အဆိုပြထားပါတယ်။ ရှေ့အခန်းတွေမှာ program flowchart အပိုင်းတွေကို သင်ကြား ပို့ချုပ်းတာကြောင့် အလွယ်တက္ကနားလည်သောပေါက်သွားပါလိမ့်မယ်။ ဒီ Developement Cycle ရဲ့ အဆိုပြုချက်လေးဟာ အင်မတန်အမေးများတဲ့ Exam Question လေးတစ်ရွေးလည်း ပြောရင်လည်း မမှားနိုင်ပါဘူး။ ကဲ

Figure 2.1 The steps in the development of a C++ program



Development Cycle ကို လေ့လာကြည့်ပါ။ နားလည် သဘော ပေါက်သွားပါလိမ့်မယ်။

ကျွန်ုတ်အနေနဲ့ Java Language ရဲအားသာချက်၊ အား နည်းချက်တွေကို တင်ပြပြီးတဲ့ အခါမှာ Java Language မှာ အထွန်အရေပါတဲ့ copy constructor method ကို လေ့လာ ကြည့်ရအောင်။ Listing 2.5 နဲ့ ရေးသားပို့ချ ရှင်းပြပေးထားပါ တယ်။ ကဲ လေ့လာကြည့်ရအောင်။ သေသေချာချာ လေ့လာကြည့် ရအောင်နော်။

### Listing 2.5

#### Illustrate the copy constructol method

```

1. import java.awt.*;
2. import java.io.*;
3. public class Point { // open
   brace for primary class
4. private double x,y;
  
```

```

5. public point (double a,
   double b) {x = a; y = b;}
6. public Point (Point p) { //copy
   constructor (pass with object)
7. { x = p.x ; y = p.y;}
8. public double x ( ) {return
   x;}
9. public double y ( ) {return
   y;}
10. public boolean equals (Point
   p) { return ( x == p.x & &
   y == p.y);}
11. public String toString ( )
12. {return new String (" (" + x
   + ", " + y + ")");}
13. // write main ( ) function
14. public static void main
   (String [Jargs) {
15. Point p = new point (2,3) / /
   use constructor with parameter
16. System.out.println ("p = "
   + p); / / output p= (2.0, 3.0)
17. Point q = new Point (p); / /
   use copy constructor
18. system.out.println ("q = "
   + q); / / output q= (2.0,3.0)
19. if (q.equals (p)) System
   .out.println ("p equal q");
20. else System.out.println
   (" P does not equal q");
21. } // close brace in main (
   ) function
22. } // close brace for primary class
  
```

### Output

p= (2.0,3.0)

q=(2.0,3.0)

p equal q

### Analysis

- ၁။ လိုင်းနံပါတ် (o) မှာ import.java.out.\*; package နဲ့ လိုင်းနံပါတ် (j) မှာတော့ import.java.io.\*; package တို့ ကြညာထားပါတယ်။ ဒါလို ပေါက်တွေကို ကြညာ မင်းဟာ store လုပ်ထားတဲ့ API function ( ) တွေကို import Statement အနေနဲ့ ယူသုံးဖို့ပဲဆိုတာကို သိထား ရမယ်နော်။ ဒါ ပေါက်တွေအကြောင်းကို ရှုသင်ခန်းစာ ဖြစ်တဲ့ 1.5 Arrays of Java ဆိုတဲ့ အခန်းကဏ္ဍမှာ အတတ်

- နိုင်ဆုံး: ပြည့်စုအောင် ရေးသားရှင်းပြခဲ့ဖြီးသားပါ။ လိုင်းနံပါတ် (၃) မှာတော့ public class point ဆိုပြီး စတင် Declare လုပ်ပါတယ်။ ဒါ Class ထဲမှာသာ ချွဲပြုနိုင်စေမယ့် private data တွေကို Declare လုပ်ပါတယ်။ x နဲ့ y အနေနဲ့ ပါ။ လိုင်းနံပါတ် (၆) မှာတော့ constructor function ကို ရေးဖို့ထားတာ တွေ့ရမှာပါ။ ဒီအခါ သတိပြုရမှာကတော့ အခုရေးလက်စ class [object ဖြစ်လာမယ့် class] ကို parameter အနေနဲ့ ခေါ်ယူထားတာပါ။ Main () function ထဲမှာမှ ဖန်တီးမယ့် object ကို ရေးလက်စမှာတင် parameter arguments အနေနဲ့ ခေါ်ယူလိုက်တာနော်။ ဒါဟာ object ကို passed လုပ်ခွင့်ပေးပြီးတွေ့မှ constructor ကို copy ကူးယူတာဝါပဲဆုံးရင် မမှားပါဘူး။ ဒါ function ရဲ့ body ထဲမှာတော့ private data အဖြစ် Declare လုပ်ခဲ့တဲ့ x နဲ့ y ကို အခြားနေရာက ခေါ်ယူတဲ့ အခါမှာ object Name.(dot)ဆိုပြီး ခေါ်ယူရမှာ မဟုတ်ဘူးလေး? ဒါတော့ p.x; p.y; လို့ ခေါ်ယူပြီး ကြေညာထားတဲ့ private data တွေနဲ့ ညွှေယူပါတယ်။ ဒါကြောင့် X=P; Y=P.Y; ဆိုပြီး ခေါ်ယူထားတာပါ။ private keyword ကြောင့် အခြားနေရာက ခေါ်သုံးလိုမရနိုင်တာကြောင့် ကိုယ်ပေးမယ့် object Name နဲုပ်တွဲပြီး ခေါ်ယူဖို့တာပေါ်နော်။ main () function ထဲမှာလည်း object Name ကို P လို့ပေးမှုဖြစ်တာကြောင့်ပါပဲ။
- JII လိုင်းနံပါတ် (၈) မှာတော့ x ကို main () function ထဲကို return ပြန်ပေးပို့အတွက် ရေးသားထားပါတယ်။ public keyword ကို သတိပြုပါ။ လိုင်းနံပါတ် (၉) မှာတော့ y ကိုလည်း ထိုအတွဲပဲ main () function ထဲကို return ပြန်ပေးပို့အတွက် ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၁၀) က equals () function ရဲ့ parameter arguments မှာတော့ class ကို object အဖြစ် ဖန်တီးပြီး ခေါ်ယူထားပါတယ်။ ဒါမှာလည်း function ရဲ့ body ထဲမှာ မူလ Declare လုပ်ခဲ့တဲ့ x နဲ့ object ရဲ့ number data အဖြစ်ဖန်တီးထားတဲ့ (တစ်နည်းပြောရရင်တော့) constructor function ကို copy ကူးယူထားတဲ့ object function ထဲက x နဲ့ y ကို p.x နဲ့ p.y လို့ ချွဲနိုင်တော်တို့ သတ်မှတ်ထားတာနဲ့ ညွှေယူပေးပို့ ရေးသားထားပါတယ်။ ပြီးတော့ လိုင်းနံပါတ် (၁၁) မှာတော့ toString () method ကို package ထဲ ကနေ ခေါ်ယူပြီးတော့ အသုံးပြုထားပါတယ်။ ဒါဟာ Output အဖြစ် ထုတ်ပေးတဲ့အခါမှာ အဖွင့်နဲ့အပိုတ် bracket () နှစ်ခုနဲ့ data နှစ်ခုကြားက comma, အတွက် ကြိုတင်ရေးသားထားတာပါပဲ။ ဒါဟာလည်း main () function ထဲကို ခေါ်ယူအသုံးပြုမယ့် [object အဖြစ် ဖန်တီးမယ့်
- class ရဲ့] number function နဲ့ data တွေ အကြောင်းပါပဲ။ သေသေချာချာ trace လုပ်ပြီးတော့ လေ့လာ ပါ။ Package function တွေနဲ့ ကိုယ်ပိုင် function တွေကို ပေါင်းစပ်ပြီး အသုံးပြုထားတာပါပဲ။ အကုန် နားလည်သဘော ပေါက်သွားရင်တော့ main () function တွေကို ဆက်လက်လေ့လာကြတာပေါ့။
- ၃။ လိုင်းနံပါတ် (၁၄) မှာတော့ main () function ကို ရေးသားထားပြီး အဲဒီထဲမှာမှ ချွဲနိုင်တော်တို့ဖန်တီးထားတဲ့ point ဆိုတဲ့ class ကို object အနေနဲ့ပြန်ပြီး ဖန်တီးယူပါတယ်။ ဒီအခါမှာ constructor function ဖန်တီးစဉ်က ထည့်သွင်းခဲ့တဲ့ parameter arguments တွေကိုအသုံးပြုထားတယ်ဆိုတာကို သတိပြုမှုမှာပါ။ ဒါကြောင့် parameter value အနေနဲ့ 2,3 ဆိုပြီး input လုပ်ပြီးထည့်သွင်းထားတာနော်။ ပြီးတော့ System.out.print - ("p = " + p); ဆိုပြီး print statement ကိုအသုံးပြုပြီး Output ထုတ်လိုက်တဲ့အခါမှာ p = (2.0,3.0) ဆိုပြီး output ထွက်လာတာပါ။
- ၄။ လိုင်းနံပါတ် (၁၇) မှာတော့ object passing လုပ်ပြီး copy constructor အဖြစ် ရေးဖို့ထားတဲ့ object ကို အခြား Name တစ်ခုနဲ့ ဖန်တီးပြီး လုမ်းခေါ်ယူလိုက်ပါတယ်။ ပြီးတော့ print statement ကို အသုံးပြုပြီးတော့ output ထုတ်ပါတယ်။ ဒီအခါမှာ output ဟာ q = (2.0,3.0) လို့လာပြတာပေါ်နော်။
- ၅။ လိုင်းနံပါတ် (၁၉) မှာတော့ boolean equals () method ကိုအသုံးပြုပြီးတော့ if (g.equals (p)) ဆိုပြီး logical statement အနေနဲ့ test လုပ်ပါတယ်။ ဒီအခါမှာ if (ဖြစ်ခဲ့လျှင် တော့) "p equal q" ဆိုတဲ့ အဖြစ်ကို output ထုတ်ပေးမှာဖြစ်ပြီး else (မဟုတ်ခဲ့ရင်တော့) "p does not equals q" ဆိုတဲ့ output ကို လာပြပေးမှာပါ။ equals () function နဲ့ test လုပ်တဲ့အခါမှာ (ဖြစ်ခဲ့တဲ့အတွက်) "p equals q" ဆိုတဲ့ အဖြစ်ကို output လာထုတ်ပေးပါတယ်။ လိုင်းနံပါတ် (၂၂) မှာတော့ main () function ရဲ့ close brace နဲ့ လိုင်းနံပါတ် (၂၂) မှာတော့ primary class ရဲ့ close brace} ကို လာထုတ်ပေးထားပါတယ်။ output နှစ်ဦးယူဉ်ကြည့်ပြီး code တွေကို သေသေချာချာ trace လုပ်ကြည့်ပါ။ နားလည်သဘော ပေါက်သွားပါလိမ့်မယ်။
- က ချွဲနိုင်တော်တို့ class ကိုစတင် ရေးဖွဲ့ အသုံးပြုလိုက်ပြီ ဆိုတာနဲ့ High performance features တွေကို စတင်အသုံးပြုလေ့လာနေရပါပြီ။ ဒီအခါမှာ OOP concepts တွေမှာ အလွန် နာမည်ကျော်ကြားတဲ့ Inheritance ဆိုတဲ့ features ကိုဆက်လက်လေ့လာကြတာပေါ့။
- Note:** Inheritance features ဟာ C++ programmer တွေ

အတွက်ကတော့ ရင်းနှီးပြီးသားနေမှုပါပဲ။ ဘာကြောင့်လို့တော့ C++ ရဲ့ object - oriented - concepts မှာ Inheritance နဲ့ ကင်းကွာလို့ မရလိုပါဘူး။

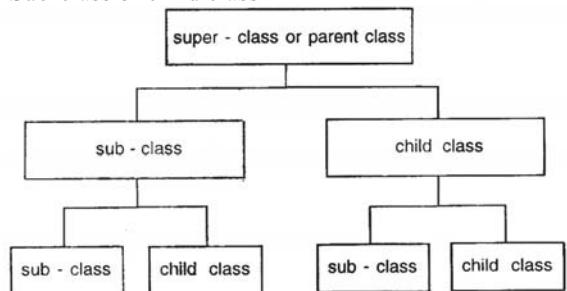
### Inheritance features in Java programming

Inheritance features ဟာ မူလကရှိပြီးသား object (class) ကို မို့ပြစ်ပြီး တစ်ဆင့်ခဲ့ ဒေါ်ယူအသုံးပြုတဲ့အတွက် မို့ပြစ်းခြင်းကိုခံရတဲ့ ပင်မ မူလ class ကိုလည်း သွားရောက်ထိနိုက်ခြင်းမရှိပါဘူး။ ကိုယ်က တစ်ဆင့်ခံပြီး ဒေါ်ယူတဲ့အခါမှုလည်း လိုချင်တဲ့ features တွေကို လည်း ထပ်မံဖြည့်စွက်ခွင့် ရှိပါသေးတယ်။ ဒီလို ပြန်လည် အသုံးချင့် စွင့် reusability ဖြစ်ခြင်းဟာ Inheritance ရဲ့ features ပါပဲ။

Inheritance ဟာ [ super - class or parent class] လို့ခေါ်တဲ့ Base class တွေရဲ့ ရေးပြီးသား မှန်နေတဲ့ class တွေကနေတစ်ဆင့် create လုပ်ထားတဲ့ features ဖြစ်ပါတယ်။ ဒီအခါမှု [ sub- class or child class] လို့ခေါ်တဲ့ Derived class ကိုဖန်တီးလိုက်တာပါပဲ။ ဒီ Derived class မှာ မူလ base class ရဲ့ features တွေလည်း အကုန်ပါတယ်။ ဒါအပြင် ကိုယ်ထပ်မံဖြည့်စွက်ချင်တဲ့ features တွေကိုလည်း ထပ်မံဖြည့်စွက်လို့ ရပါသေးတယ်။ ပင်မမူလ Base class မှတ်လာတဲ့ features တွေထဲက ကိုယ်မလိုချင်တဲ့ feature တွေ / method တွေကိုလည်း ဖယ်ရှားထားခဲ့လို့ ရပါတယ်။ ဒီလိုဖယ်ရှားပြင်ဆင် တာတွေ၊ ဖြုတ်ပယ်ထားခဲ့တာတွေ၊ ထပ်မံဖြည့်စွက်တာတွေစွတဲ့ ပြပိုင်ရေးသားမျှတွေလုပ်တဲ့အခါမှု programmer ဟာ မူလက [super - class or parent class] လို့ခေါ်တဲ့ Base class ကိုသွားထိစရာမလို့ တစ်ဆင့်ခဲေါ်ယူခြင်းပြင့် အလုပ်လုပ်နိုင်ပါတယ်။ ဒီလို ပြန်လည်အသုံးချခွင့်ရှိတဲ့ reusability စွမ်းရည်ဟာ inheritance ရဲ့ စွမ်းရည်ပါပဲ။ အခြေခံကစားတော့ နားလည် အောင်လို့ flow diagram နဲ့ အရင် ရှင်းပြထားပါတယ်။ သော် ချာချာ လေ့လာကြည့်ပါ။

Figure 2.2 Step of ingeritance method in Java programming language

Super-class or parents  
Sub-class or child class



Inheritance ဆိုတာဟာ class အချင်းချင်းကို မို့ပြစ်ပြီး အသုံးပြုတဲ့ method ပါပဲ။ super - class or parent class တွေကို sub - class or child class တွေကနေတစ်ဆင့် မို့ပြစ်ပြီး အသုံးပြုရတာပါပဲ။ ဒါဟာ အခြေခံနည်းပညာပါပဲ။ ဒီထက်ပို့ပြီးနားလည်အောင်ပြောရရင်တော့ Java ရဲ့ကျော်ကြားလှတဲ့ package နည်းပညာတွေအထိ သက်ဆိုင်နေပါတယ်။ (Package တွေ အကြောင်းကို 1.5 Arrays of Java မှာ ရှင်းလင်းတင်ပြပြီးပါပြီ။) ရှေ့က programmer တွေ ဖုန်းသွားတဲ့ အတဲ့ကမ္မ လိုအပ်တာ တွေကို ဒေါ်ယူအသုံးပြုမယ်။ မလိုအပ်တာတွေကို ခေါ်သုံးစရာ မလိုပါဘူး။ Project တွေ ရေးသားရာမှု့တော့ ဒီနည်းပညာ အကြောင်းကို နားမလည်ရင် ဘာမျှ ရေးသားနိုင်မှာ မဟုတ်ပါဘူး။ ဘာကြောင့်လိုတော့ ရှေ့က programmer တွေ ဖုန်းထားတဲ့ object နဲ့ classes တွေထဲက လိုချင်တာကို ဆွဲထုတ်အသုံးပြုပြီး တော့ နောက်ထပ် features တွေကိုလည်း ထပ်မံဖြည့်စွက်လို့ ရသေးတာဆိုတော့ Inheritance method ဟာ အင်မတန်အသုံးဝင်တဲ့ နည်းပညာတစ်ရပ်ဆိုတာကို နားလည်သောပေါက်သွားလောက်ပါပြီ။

**Note:** Inheritance နည်းပညာဟာ code တွေကို reuse လုပ်ပြီး software maintenance လုပ်ရာမှုလည်း အလွန်တစ်ရာမျှ အသုံးဝင်ပါတယ်။ ဘာကြောင့်လိုတော့ inheritance features ကိုအသုံးချခြင်းအားဖြင့် minimizes redundant code ဖြစ်ပြီး time, cost စတာတွေကိုပါ လျော့ချိန်လိုပါပဲ။ software ရဲ့ မူလစွမ်းရည်ကိုလည်း ထိခိုက်စေခြင်းမရှိပါဘူး။

အပေါ်မှာ ရေးသားရှင်းပြခဲ့တဲ့ အချက်အလက်တွေနဲ့ Figure 2.2 က flow Diagram ကို လေ့လာပြီးသွားရင်တော့ inheritance ပုံစံပြရေးသားရှိတဲ့ class declaration syntax form ကို ရေးပြပြီးတော့ ရှင်းပြထားပါတယ်။ ဒီ class form ကိုအလွှာတဲ့ ရအောင်ကြည့်ပြီးတော့ မှတ်ထားမှုသာ နောက်နိုင်း program တွေ ရေးတဲ့အခါမှု အလွယ်တကူရေးသားနိုင်မှာ ဖြစ်ပါတယ်။ က Basic Syntax - form ကိုလေ့လာကြည့်ပါ။

```

Class Sub-class_name extends
super-class_name{
{ Public function_name ( )
{ //arguments and code }
}
  
```

ဒီ Basic Syntax - form မှတ်ဝင်တဲ့ sub - class - name နေရာမှာ မို့ပြစ်းမယ့် class ရဲ့အမည်ကို ရေးသားပြီးတော့ extends ရဲ့နောက်က super - class - name ဆိုတဲ့ အမည်နေရာ မှာတော့ မို့ပြစ်းခြင်းကိုခံရမဲ့ class ရဲ့ နားမည်ကို ရေးသားရမှာဖြစ်ပါတယ်။ public function name ဆိုတာကတော့ မိမိစိတ်ကြော်

ရေးသားတာပဲဖြစ်ဖြစ်၊ ပြန်လည်ပြင်ဆင်ရေးသားတာပဲ ဖြစ်ဖြစ် ရေးသားရမယ့် function name ပါပဲ။ ဒီ class form ကို နားလည်သော့ပါက်သွားရင်တော့ အောက်မှာ Example အနေနဲ့ ရေးပြထားတဲ့ listing 2.6 ကို လေ့လာကြည့်ပါ။

### Listing 2.6 Using Inheritance method in Java Programming

```
1. public class Test Inher {
2.     public static void main
3.         (String args [ ] ) {
4.             Clone c = new clone ( );
5.             System . Out. print In ("Parent
6.             value is :" + c. getvalue
7.             ( ) ); // parent value is : 0
8.         }
9.     }
10. Class Clone { // parent
11.     int value;
12.     public int getvalue ( ) {
13.         return value; }
14. class Quarter extends coin {
15.     public Quarter ( ) { value =
16.         25 } // constructor function }
```

#### Output

Parent value is: 0  
Child value is:25

#### Analysis

၁။ ဒီ program မှာ Inheritance features ကို အထွယ်ကျခဲ့် နားလည်အောင်လို့ Demonstrate လုပ်ထားပါတယ်။ သေ သေချုပ်ချုပ် ဂရာတစိုက်လေ့လာပါ။ လိုင်းနံပါတ် (၁) မှာ TestInher ဆိုတဲ့ primary class ကြီးကို ရေးဖွဲ့ထား ပါတယ်။ လိုင်းနံပါတ် (၂) မှာတော့ main () function ကို ရေးဖွဲ့ထားတာကို တွေ့ရမှာပါ။ ဒီတော့ အရေးကြီးတဲ့ပင်မ object class နဲ့ Inheritance လုပ်ပြီး တစ်ဆင့်ခံမိုင်းထားတဲ့ [ပင်မ object ကို extends လုပ်ထားတဲ့ ] class ကို အရင်သွားပြီး လေ့လာကြရအောင်။

- ၂။ လိုင်းနံပါတ် (၁၀)မှာ clone ဆိုတဲ့ class ကိုရေးဖွဲ့ထားပါ တယ်။ အဲဒီ class ထဲမှာတော့ int data type ကို အသုံးပြုပြီး value ဆိုတဲ့ variable ကို Declare လုပ်ပါတယ်။ အဲဒီ clone ဆိုတဲ့ class ရဲ့ body ထဲမှာတင်ပဲ public keyboard ကိုအသုံးပြုပြီးတော့ getvalue () ဆိုတဲ့ function ကိုရေးဖွဲ့ပါတယ်။ သတိပြုရမှာကတော့ ဒီ getvalue () function ထဲမှာ value ရဲ့ တန်ဖိုးဟာ ဘာမျှ သတ်မှတ်ထားခြင်း မရှိပါဘူး။ ဒီတော့ သူတန်ဖိုးဟာ 0-zero ပေါ့။ လိုင်းနံပါတ် (၁၃) မှာတော့ အဲဒီ object အဖြစ် ဖန်တီးမယ့် class ရဲ့ close brace ကို ထိတ်ထားတာကို တွေ့ရမှာပါ။
- ၃။ လိုင်းနံပါတ် (၁၄) မှာတော့ မူလ class ဖြစ်တဲ့ clone ဆိုတဲ့ class ကို ရေးဖွဲ့လိုက်တာကို တွေ့ရမှာပါ။

**Note:** C++ language မှာတော့ class တွေကို Inheritance လုပ်ရမှာ public ဆိုတဲ့ keyword နဲ့ပဲ Accessing လုပ်ကဲတာပါ။ Java language ကတော့ ရောထွေးမှတ် မလိုလားဘဲ ရိုးရှင်းလွယ်ကြစေခဲ့အတွက် extends ဆိုတဲ့ keyword ကို အသုံးပြုထားတာပါပဲ။ class- Declaration Syntax -form မှာ သေသေချုပ်ရေးပြုပါတယ်။ သေသေချုပ်လေ့လာကြည့်ပါ။

ဒီမိုင်းထားတဲ့ class ဖြစ်တဲ့ Quarter ဆိုတဲ့ class ကိုရေးတဲ့အခါမှာတော့ constructor function အဖြစ် Quarter () ဆိုတဲ့ function ကိုရေးဖွဲ့ခဲ့ပါတယ်။ ဒီ constructor function ထဲမှာ value = 25; လို့ Declare လုပ်ခဲ့ပါတယ်။ သတိပြုရမယ့်အချက်ကတော့ ဒီမိုင်းထားတဲ့ Child class (or) Sub-class ထဲမှာ variable ရဲ့ တန်ဖိုး (value = 25;) ကိုသာ ကြည့်ထားပါတယ်။ ဒီ variable ကို data type ကို အသုံးပြုပြီးတော့ Declare လုပ်ခဲ့ခြင်းမရှိပါဘူး။ Declare လုပ်ခဲ့ခြင်းမရှိဘူး။ ဘာကြောင့် အသုံးပြုရတာလဲဆိုတော့ ဒီ class ဟာ clone ဆိုတဲ့ class ကို မို့မို့ပြီးမှ ရေးထားတာမဟုတ်လား။ ဒီတော့ variable တွေကို ပြန်ကြည့်စရာ မလိုတော့ဘူးပေါ့နော်။ ဒီ class ကို object အနေနဲ့သာ Declare လုပ်ပြီး ခေါ်သုံးမယ်ဆိုရင်တော့ value ရဲ့ တန်ဖိုးဟာ 25 အနေနဲ့လာပြီး calculate လုပ်ရမှာပေါ့နော်။

- ၄။ လိုင်းနံပါတ် (၂) မှာတော့ Clone ဆိုတဲ့ လိုင်းနံပါတ် (၁၀) မှာ ဖန်တီးခဲ့တဲ့ class ကို object အနေနဲ့ ဖန်တီးခေါ်ယူလိုက်ပါတယ်။ ပြီးတော့ ဒီ class ရဲ့ member function ဖြစ်တဲ့ getvalue () ကို ခေါ်ယူအသုံးပြုလိုက်ပါပြီ။ အရေးကြီးတဲ့ အချက်ကတော့ ဒီ clone ဆိုတဲ့ class မှာ constructor function မပါဘူးဆိုတဲ့ အချက်ပဲနော်။ ဒီတော့ ဒီ class ရဲ့ member data ဟာ ဘာမျှ Declare လုပ်ထားခြင်းမရှိတာကြောင့် default အနေနဲ့ 0 - zero

လိုပဲထွက်လာမှာဖော်။ ဒါကြောင့် Output ဟာ Parent Value : 0 လို့ ထွက်လာတာပါပဲ။ ကဲ print statement နဲ့ output ထုတ်အပြီးမှာ လိုင်းနံပါတ် (၅) မှာ မြားထားလိုက်ပါတယ်။ လိုင်းနံပါတ် (၅) ဟာ Blank line feed အနေနဲ့ပါ။ မြားလည်း ရုပါတယ်။ နောက်လိုင်းနံပါတ် (၆) မှာ တော့ တစ်ဆင့်ခဲ ရေးဖွဲ့ထားတဲ့ Quater ဆိတ် ကို ခေါ်ယူလိုက်ပါတယ်။ ဒီအခါ ဒီ class (object) ထဲမှာ constructor function ရေးဖွဲ့ထားခဲ့တယ် ဆိတ်ကို သတိပြုရပါမယ်။ ဒါဟာ အလွန်အရေးကြီးတဲ့ အချက်ပါပဲ။ ၅။ ဒီအခါမှာ constructor function ထဲမှာ value အနေနဲ့ ရေးထားတဲ့ 25 ဟာ အလိုအပျောက် ဝင်လာနဲ့ပါပြီ။ ပြီးတော့ လိုင်းနံပါတ် (၇) မှာ print statement နဲ့ ခေါ်ယူပြီး Output ထုတ်လိုက်ပါတယ်။ ဒီအခါ getvalue() function ဟာ တစ်ဆင့်ခဲ Inheritance Technique နဲ့ခေါ်ယူတာဖြစ်တဲ့အတွက် ဝင်လာပြီး အဖြော်ပြုပေးပါတယ်။ ဒီအခါ child value is : 25 လို့ output ထွက်လာခဲ့ပါတယ်။ လိုင်းနံပါတ် (၈) မှာ main () function ရဲ့ close brace } နဲ့ လိုင်းနံပါတ် (၉) မှာ primary class ရဲ့ close brace} ကို လာပိတ်ထားပါတယ်။

ကဲ ဒီ program ကို လေ့လာကြည့်ရင် Inheritance Technique ကို demonstrate လုပ်ထားတာကို အလွယ်တကူနားလည်း သဘောပေါက်သွားပါလိမ့်မယ်။ ကဲ ဆက်လက်ပြီးတော့ Inheritance အကြောင်းကို နားလည်း သဘောပေါက်သွားအောင် Listing 2.7 နဲ့ ထပ်မံပြီး program ရေးပြထားပါတယ်။ ထူးမြားတာကတော့ ဒီ program ကိုရေးသားတဲ့အခါမှာ [Inheritance ကို အသုံးပြုတဲ့ နည်းပညာမှာ] Super -class (or) parent class ရဲ့ constructor ကို လုပ်ပြီး ခေါ်ယူထားတာကို တွေ့ရမှာပါ။ ဒါဟာ အလွန်အသုံးဝင်တဲ့ လုပ်ငန်းခွင်မှာ programmer တွေအနိုင် ပြုထောက်သာဖော့တဲ့ နည်းပညာတွေခဲ့ပါပဲ။ ကဲ သေသေချေချာ လေ့လာကြည့်ပါ။ Listing 2.7 အနေနဲ့ ရှင်းပြ ပို့ချေပေးထားပါတယ်။

#### Listing 2.7 using super to call the super - class constructor

```
1. Class Super Class {
2.     protected int Supaer Value;
3.     public Super Class (int P) {
4.         Super Value = P;
5.     }
6.     Class SubClass extends Super
```

```
Class {
7.     protected int Subvalue;
8.     public SubClass (int subV,
9.         int p) {
10.        Super Class (*);
11.        SubValue = SubV;
12.    }
13.    public Class TestSuper {
14.        public static void main
15.            (String args [ ] ) {
16.                SubClass ref = new SubClass (4,3);
17.                System.out.printf ("Super
18.                    Value is :" + ref.get Super
19.                    Value ( ) );
18.    }
19.}
```

#### Output

Super Value is : 3

Sub Value is: 4

#### Analysis

၁။ ဒီ program ဟာ Inheritance အကြောင်းကို ကျမ်းကျင် စားလည်အောင် ရေးပြထားတဲ့ program တစ်ခုပဲ ဖြစ်ပါတယ်။ လိုင်းနံပါတ် (၁) မှာ object အဖြစ်ဖန်တီးမယ့် class တစ်ခုကို Super Class ဆိတ်အမည်နဲ့ Declar & write လုပ်ပါတယ်။ အဲဒီ class ရဲ့ member function ကိုတော့ လိုင်းနံပါတ် (၂) မှာ ကြညားပါတယ်။ အဲဒီအခါမှာ သတိပြုစရာတော့ protected ဆိတ် keyword ကို အသုံးပြုထားတာပါပဲ။ ဒါဟာ ဒီ data member ကို ခေါ်ယူတဲ့အခါမှာ အကုန်အသတ်နဲ့ပဲ ခေါ်ယူစေခဲ့လိုပါပဲ။ Note: protected keyword အကြောင်းကိုတော့ လာမယ့် အခန်းကဏ္ဍမှာ သိုံးသန့် တင်ပြပေးပါမယ်။ Protected keyword ကို "Only allowed within the fragments of a class and its sub class" လို့ အမိုးပိုးဖွင့်ဆိုထားတာဆိုတော့ ဒီ keyword တွေအကြောင်းကို သိုံးသန့်ရှင်းပြရမှာပါ။ လိုင်းနံပါတ် (၃) မှာ အဲဒီ class ရဲ့ constructor function ကိုရေးဖွဲ့ပါတယ်။ အဲဒီ constructor function မှာ parameter value (or) arguments value ကိုထည့်ထားတယ်နော်။ int data type နဲ့ p လို့ Declare လုပ်ထားတာဖော်။ ဒီ constructor function ထဲမှာတော့

- ၂။ လိုင်းနံပါတ် (၄) မှတော့ public keywordကို အသုံးပြု၍  
တော့ int getSuperValue () ဆိုတဲ့ function ကို  
ရေးပါတယ်။ အမှန်ကတော့ main () function ထဲကို  
return ပြန်ပေးပို့ပါပဲ။ လိုင်းနံပါတ် (၅) မှတော့ အခါး  
Super Class ရဲ့ close brace } ကို လာပိတ်ထားပါတယ်။  
လိုင်းနံပါတ်(၆) မှာ Sub Class ဆိုတဲ့ Name နဲ့ Super  
Class [အပေါ်မှတော်ခဲ့တဲ့ class] ကို extends လုပ်ထား  
တာနော်။ ဒီ extends လုပ်ထားတဲ့ class ထဲမှာ int data  
type ကို အသုံးပြု၍တော့ SubValue ကို Declare  
လုပ်ပါတယ်။ ဒီအခါး constructor function ကိုရေးဖွဲ့  
ထားတာကိုလည်း တွေ့ရမှာပါ။ ထူးခြားချက်ကတော့ ဒီ  
constructor function ကို pass -by -object  
လုပ်တဲ့အခါးမှာ parameter arguments ဟာ int Subv  
နဲ့ int P ဆိုပြီး နှစ်ခု တွေ့ရမှာပါ။ ဒါဟာ Inheritance ရဲ့  
features ပါပဲ။ အပေါ် class မှတော်ဝင်တဲ့ စွမ်းရည်ကို  
၏သုတေသနလိုက် ကိုယ်လိုချင်တဲ့ feature ကိုလည်း ထည့်မံဖြည့်  
စွက်ထားပြီလေ။ ဒီတော့ ဒီ Sub Class ဟာအပေါ်က  
Super Class မှတော်ဝင်တဲ့ စွမ်းရည်တွေ အကုန်ပါဝင်နေ  
သလိုပဲ Super Class ထက်ပိုတဲ့ [ထည့်မံဖြည့်စွက်ထားတဲ့  
features ] တွေလည်း ပါဝင်နေပါတယ်။

၃။ လိုင်းနံပါတ် (၁၃) မှာ TetSuper ဆိုတဲ့ Class ကိုရေးဖွဲ့  
ထားပါတယ်။ ဒီ class ထဲမှာမှ main () function ကို  
ရေးဖွဲ့ပါတယ်။ ဒီ main () function ထဲမှာ Sub Class  
ကို obj တစ်ခုအနေနဲ့ ဖန်တီးပါတယ်။ ဒီအခါး Sub Class  
ဟာ obj အဖြစ် ဖန်တီးခဲ့ရတဲ့အခါးမှာ သူရဲ့ constructor  
function ဟာ ဝင်ရောက်ပြီးတော့ အလုံအလျောက် invoke ဖြစ်လာပါတယ်။ ဒါအပြင် Sub Class ဟာ  
SuperClass ကို extends လုပ်ထားတာဆိုတော့ အပေါ်က  
Super Class ရဲ့ စွမ်းရည်တွေလည်း ပါဝင်နေပါပြီ။ ဒီ  
SubClass ရဲ့ constructor function ထဲကို parameter arguments ရဲ့ တန်ဖိုးအဖြစ် ၄နဲ့ ၃ ကို ထည့်သွေး  
လိုက်ပါတယ်။ ပြီးတဲ့အခါးမှာ SubClass ကို obj အဖြစ်  
ဖန်တီးထားရာကနေပဲ getSuperValue () function  
ကို၏ယူပြုး output ထုတ်ပါတယ်။

**Note:** ဒီနေရာမှာ စာဖတ်သူရဲ့စိတ်လုပ်တော့ SubClass ကနေ SuperClass ကိုလုပ်:ခေါ်ယူတာလို့ ထင်နေမိမှာပါ။ ဒါဟာ အမှန်တကယ်တော့ Inheritance စွဲ:ရှုပ်ကြောင် SuperClass

၄။ features တွေကို အကုန်လှမ်းခေါ်ယူပြီးတော့ အသုံးချလို့ ရနေတာပါပဲ။ ဒါဆို ရှင်းသွားမှာပါ။

လိုင်းနံပါတ် (၁၇) မှာတော့ SubValue ရဲတစ်နံပါတ် Out-  
put ထုတ်ပေးပါတယ်။ main () function ရဲ close brace }  
ကို လိုင်းနံပါတ် (၁၈) မှာ လာပိတ်ထားပြီးတော့ လိုင်းနံပါတ်  
(၁၉) မှာတော့ TestSuper Class ရဲ close brace } ကို  
လာပိတ်ထားပါတယ်။ output နဲ့ နှင့်ယဉ်ဥကြီး ဒီ program ကို  
လေ့လာကြည့်ပါ။ နားလည်သဘောပေါက်သွားပါတယ်မယ်။

## Method Overloading

JAVA Language-ରୁ Inheritance features ପ୍ରିସେଟୋ ଫେରିଗ୍  
ତାନ୍ତରିକ ଲାଭମାନ୍ୟ କାହାରେ କାମ କରିବାକୁ ପାଇଁ କାମ କରିବାକୁ  
କାମ କରିବାକୁ କାମ କରିବାକୁ କାମ କରିବାକୁ କାମ କରିବାକୁ

**Note:** constructor function ഒന്തേ Class Name ഫലിക്കുന്നതാണ് function Name എന്നതാണ്। Overloading Method ഒന്തേ function അവും ഏറ്റ്: Name ട്രാക്കേഡിൽ ഉൾപ്പെടെയാണ്।

အခါကတော့ arguments အနေနဲ့ ပါဝင်တဲ့ parameter တွေ ကွားမြားတာဖြစ်လိုပါပဲ။ အမိဘယ်ကတော့ function တစ်ခု ကို double type နဲ့ရေးထားမယ်။ အဲဒီ function Name နဲ့ပဲ နောက်ထပ် function တစ်ခုကို int data type နဲ့ဖွံ့ဖည်းထားမယ်။ ဒီအခါ Run လုပ်တဲ့အခါမှာ compact ဖြစ်တဲ့ data type နဲ့ သင့်လျော်တဲ့ function ထဲကိုဝင်ပြီ။ အလုပ်လုပ်ရှုပါပဲ။ ဒီလို အလုပ်လုပ်တာကို Overload လုပ်တယ်လို့ ခေါ်ပါတယ်။ Overloading Method ကို တစ်နည်းအားဖြင့် polymorphism လိုလဲ ခေါ်ခေါ်ပါသေးတယ်။

**Note:** Polymorphism ဆိတ်ဘာ C++ ရဲ့ နာမည်ကြီး အခေါ်အဝေါနဲ့ features ဖြစ်ပါတယ်။ Poly ဆိတ်ဘာကတော့ many လို့ အဓိပ္ပာယ်ရှုံးမြို့ mophism ဆိတ်ဘာကတော့ form လို့ အဓိပ္ပာယ် ထွက်ပါတယ်။ ဒါကြောင့် သင့်တော်ရာ data type နဲ့ ပုံစံအဖျိုး ပျိုး ရေးဖွဲ့ထားတဲ့ function ကို polymorphic function (or) overloading function လို့ ခေါ်ပေါ်နိုင်ပါတယ်။ ကဲ ဒီတော့ လက်တွေ နားလည်သော်ပေါက်နိုင်အောင် Sample code ကို အသုံးပြုပြီး ရှင်းပြပါးမယ်။ သေသေချာချာလေ့လာကြည့်ပါ။

### Sample

```

1. public date (String Str) // parameter using string data type
2. { System.out.print In (Str);}
3. // write another function with same Nam
4. public date (int mm, int dd, int yy) // parameter using int data type
5. { System.out.print In (mm + dd + yy);

```

ဒီ function နှစ်ခုစလုံးမှာ function Name ဟာ date ဆိုပြီးဖြစ်နေတာကို တွေ့ရမှာပါ။ ဒီ function တွေ၏ body ထဲမှာလည်းအဖြစ် print ထုတ်ပေးဖို့အတွက် System.out.print In () Method ကိုပဲ အသုံးပြထားတာကို တွေ့ရမှာပါ။ ဒါပေမဲ့လိုင်းနှင့် (၁) က data () function ရဲ့ arguments ဟာ String data type ဖြစ်ပြီး လိုင်းနှင့်ပါတ် (၃) က date () function ရဲ့ arguments ကတော့ int data type တွေဖြစ်နေပါတယ်။ ဒါကို ခေါ်ယူအသုံးပြုမယ်ဆိုရင် Complier ကနေ compact ဖြစ်တဲ့ data type နဲ့ စိစိုးပေးမှာပါ။ ရွှေ့ထွေးကုန်မှာကို စိုးရိမ်စရာမလိုပါဘူး။ ဒါဟာ Method overloading ရဲ့ အခြေခံသဘောတရားပါပဲ။ က ဒါဆိုရင် Method Overloading အကြောင်းကို အခြေခံကျကျ နားလည်သဘောပေါက်သွားအောင် program ရေးပြီးလေ့လာကြတာဖေါ့။ Listing 2.8 နဲ့ Method overloading အကြောင်းကို ရှင်းလင်းပြထားပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

### Listing 2.8 Overloading function demonstrate

```

1. public Class Testol {
2. public static void main
   (String args [ ] ) {
3. DATE D = new DATE ( );
4. D.date (" 9/9/2005 ") ; / /
   String data type used
5. D.date (9,9, 2005 ) ; // int
   data type used
6. }
7. }
8. public class DATE {
9. / / overloading function ( )
   in two ways
10. public String date (String
   Str)
11. { System.out.print In (
   "Date:" + Str) ; }
12. public int date (int mm, int
   dd, int yy)
13. { System.out.print In ("Date:" +

```

```

   + mm + "/" + dd + "/" + yy ) ;}
14. }

```

### output

Date: 9/9/2005  
Date: 9/9/2005

### Analysis

၁။ ဒီ program ကို လေ့လာကြည့်မယ်ဆိုရင် လိုင်းနှင့်ပါတ် (၁) မှာ Testol ဆိုတဲ့ primary class ကိုဖွံ့ဖည်းပါတယ်။ ဒီ Class ထဲမှာပဲ main () function ကိုရေးဖွဲ့ပါတယ်။ ပြီးတဲ့အခါ ဒီ main () function ထဲမှာပဲ လိုင်းနှင့်ပါတ် (၂) မှာ ရေးသားတည်ဆောက်ထားတဲ့ DATE ဆိုတဲ့ Class ကို object အနေနဲ့ဖန်တီးပြီးတော့ ခေါ်ယူပါတယ်။ ဒီအခါ မှာ main () function ထဲကို object အနေနဲ့ ဝင်ရောက်လာတဲ့ DATE class ရဲ့ member function တွေကို obj. (dot) ခံပြီး ခေါ်ယူအသုံးချပါတော့တယ်။ လိုင်းနှင့်ပါတ် (၃) မှာ D.date ("9/9/2005"); ဆိုပြီး data input ထည့်သွင်းထားတာဟာ String data type နဲ့ရေးသားခဲ့တဲ့ function နဲ့သွားရောက် ချိတ်ဆက် အလုပ်လုပ်မှာ ဖြစ်တာကြောင့် comments အနေနဲ့ // double backslash ခံပြီး ရေါ်ပြောပါတယ်။ နောက် လိုင်းနှင့်ပါတ် (၄) မှာတော့ D.date (9,9, 2005); ဆိုပြီး date input လုပ်ထားတာကတော့ int data type နဲ့ ရေးထားခဲ့တဲ့ date () function နဲ့သွားပြီး compact ဖြစ်နေပါတယ်။

၂။ က ဒါဆိုရင်တော့ ဒီ object အနေနဲ့ ခေါ်ယူအသုံးနေတဲ့ DATE class ကို သွားရောက်လေ့လာကြည့်ကြရအောင်။ ဒီ DATE Class ကို လိုင်းနှင့်ပါတ် (၈) မှာ တည်ဆောက်ထားပါတယ်။ ဒီ Class ကို သီးသန့် ဖွဲ့စည်းတည်ဆောက်ထားတာပါ။ လိုင်းနှင့်ပါတ် (၉) မှာ ရေးသားထားတာဟာ comments အနေနဲ့ ရေးသားထားတာပါ။ မရေးရင်လည်းရုပါတယ်။ က ဒီတော့ လိုင်းနှင့်ပါတ် (၁၀) က ရေးထားတဲ့ date () function / method ကို စတင်လေ့လာကြည့်ရအောင်။ ဒီ function ကို String data type နဲ့ အသုံးပြုရေးသားထားပါတယ်။ ဒီအခါရှာ arguments အနေနဲ့ data တွေကို input ထည့်သွင်းတဲ့အခါမှာ function () ရဲ့ arguments နေရာမှာ data ကို input ထည့်သွင်းလိုက်တာဘဲ object ကနေ လက်ခံထားတာဟာ အတူတူပဲ ဖြစ်သွားပါပြီ။ String data type ကို အသုံးပြထားတာဆိုတော့ data input လုပ်တဲ့အခါမှာ Double quote " "နှစ်ခုကြားမှာ စာသား Text အနေနဲ့ ထည့်သွင်းရမှာနော်။ ဒါဟာ String data type ကို အသုံးပြထားလိုပါ ဆိုတာကို အထူးသတိပြုပါ။

၃။ လိုင်းနှင့်ပါတ် (၁၂)မှာ ရေးသားထားတဲ့ data function /

method ကတော့ int data ကို အသုံးပြုထားပါတယ်။ ဒီတော့ function / method ရဲ့ arguments နေရမှာလည်း month, day, year ဆိုပြီး ကဲပြားခြားမားအောင် int mm, int dd, int yy ဆိုပြီးတော့ parameter arguments တွေကို နဲ့ခြားပြုသို့ ရေးသားခဲ့ပါတယ်။ ဒီအခါမှာ arguments ကနေတစ်ဆင့် function ထဲကို pass by value လုပ်တဲ့ အခါမှာ data type တွေဟာ int data type ဖြစ်တဲ့အတွက် month, day, year စတဲ့ data type တွေကြားမှာ (I) Symbol လေးအောင်လို့ " / " Double quote ကို အသုံးပြုပြီး ထည့်သွင်းရေးသားထားပါတယ်။ ဒါကြောင့် လိုင်းနဲ့ပါတ် (၁၃) မှာ " " sign လေးတွေကို + assign လိုပ်ထားတာပါ။ ကဲဒီတော့ function ( ) အနေနဲ့တော့ date ( ) ဆိုပြီး နှစ်မျိုးကဲပြားခြားမားတာကိုလည်း သဘောပေါက်ပြီးနော်။ ဒါပေမဲ့ ဒီ program ကို Run လိုက်တဲ့အခါမှာ compact ဖြစ်တဲ့ data type နဲ့ အလုပ်လုပ်မှုဖြစ်တဲ့အတွက် output အနေနဲ့ –

Date: 9/9/2005

Date: 9/9/2005

ဆိုပြီး output ထွက်လာခဲ့တာပါပဲ။ ဒီလိုမျိုး ပုံစံအမျိုးမျိုး ကဲပြားခြားမားတဲ့ Date type တွေနဲ့ function / Method form တွေ ရေးသားတာကို Method Overloading လုပ်တာလို့ ခေါ်သလို Polymorphism လုပ်တယ်လို့ ခေါ်ဝါးနှင့်ပါတယ်။

Note: အလွယ်ဆုံး နားလည်းအောင် ဥပမာပေးရရှင်တော့ ဆေးရုံ တစ်နှစ်မှာ နှလုံးရောဂါအထူးကုသရာဝန်ကြီး၊ အဆတ်ရောဂါအထူးကုသရာဝန်ကြီး၊ အထွေထွေရောဂါကု ဆရာဝန်ကြီး ခွံစိတ်အထူးကု ဆရာဝန်ကြီး၊ အခုံကြေားဆိုင်ရာ အထူးကု ဆရာဝန်ကြီး စသဖြင့် ရှိရှိလိုပေါ့။ ဒီအခါ ရောဂါကုသရှင်လုပ်တဲ့အခါမှာ သက်ဆိုင်ရာဆရာဝန်ကြီးက လက်ခံတွေဆိုပြီး ကုသပေးသလိုပေါ့။ နှလုံးရောဂါကုသရန် လူနာလာတယ် ဆိုကြပါစို့။ နှလုံးရောဂါကု ဆရာဝန်ကြီးက လက်ခံကုသပေးမှာပါ။ ရောဂါနှင့် မသက်ဆိုင်တဲ့ အဆတ်ရောဂါကုသရာဝန်ကြီးက လက်ခံတွေဆုံးမှာ မဟုတ်ပါဘူး။ ဒါဟာ Method Overloading ကို ကျော်သားမားလည်းအောင် ရေးသား ရှင်းလုပ်းပေးတာပါပဲ။ ကဲ ဒါဆိုရင်တော့ နားလည်းလွယ်သွားမှာပါ။

Chapter [1] က Listing 1.21 က function () တွေ့ရေး စဉ်က အသုံးပြုရေးသာခဲ့တဲ့ program ထိုလေးမှာရေးတဲ့ print () function type နှစ်မျိုးကဲဖော်ပြုးဟာလည်း Method Overloading type ပုံစံပါပဲ။ ပုံစံတူဖြစ်တဲ့အတွက် အလျဉ်းသင့်သလို ပြန်လှန်လေ့လာသင့်ပါတယ်။ ပြန်လှန်လေ့လာထားပါ။ Over-

loading Method နဲ့ပတ်သက်ပြီး နားလည်းလွယ်အောင် ရှင်းပြုးပါပြီ။ ဒီအခါ overloading ပဲ့ သဘောတရားကို နားလည်းလွယ်အောင် ပြောရရှင်တော့ function / Method တွေဟာ common name တစ်ခုတည်း ပေးထားပါတယ်။ ဒီအခါ အသုံးပြုရေးသားခဲ့တဲ့ function / Method တွေဟာ မတူညီတဲ့ data type တွေနဲ့ ဖွစ်ည်းတည်ဆောက်ရပါတယ်။ ဒါအဖြင့် constructor function ကိုရေးသားပြီးတော့ default values ကိုလည်းအသုံးပြုထည့်သွင်းခဲ့လို့ ရှုပါတယ်။ အကယ်၍သာ New-soft value ကို input မလုပ်ခဲ့ရင် မူလ default value ဟာ ဝင်လာပြီး အလုပ်လာလုပ်မှာပါ။ ဒါကို လက်တွေနားလည်းသဘောပေါက်အောင် ရှုမှာ Sample ရေးပြဲခဲ့တဲ့ date () function ကို overloading type ပုံစံဝင်အောင် ရေးပြုပါမယ်။ လေ့လာကြည့်ပါ။

Sample

```
void date (String Str);
void date(int mm, int dd, int yy);
void date (double mm, int dd, double yy);
void date (int mm, int dd, Str yy);
void date (int mdy);
```

ဒီအခါ အပေါ်ကရေးပြဲခဲ့တဲ့ sample code တွေဟာ အသုံးချမယ့် နယ်ပယ်နဲ့ အနေည်းငယ်စွဲလွှာ ဖြစ်ချင်ဖြစ်နေပါတယ့်မယ်။ ဥပမာ အနေနဲ့ ပြောရရှင်တော့ double ဆိုတဲ့ data type ကို variable က year လို့ ရော်ညာထားတာကြောင့်ပါပဲ။ အဆင်ပြေသင့် တော်မယ့် data type ဟာ int type ပဲ ဆိုတာကို စာဖတ်သူအနေနဲ့ တွေ့ခေါ်မှာပါ။ ဒါပေမဲ့ ဒါဟာ Sample အနေနဲ့ဆိုတာကို သတိပြုထားလိုက်ပါး။ Overloading Method ကိုအသုံးပြုပြင်းအားပြင့် –

parameter arguments အနေနဲ့ compact ဖြစ်သင့်လျဉ်းမှု ပုံစံအားပြင့်။ အလုပ်လုပ်မှာပါ။

မူလ default - value ကြောင့် Class / object ပဲ မူလစွမ်း ရည်ကို မထိခိုက်စော့ ထပ်မဖြည့်စွဲဖြုံး Class ပဲ စွမ်းဆောင်ရည်တွေထဲက အရေးပါတဲ့ constructor function & Overloading Method ကို ပေါင်းစပ်အသုံးချမိုင်မဲ့ စွမ်းရည်တွေပဲ ဖြစ်ပါတယ်။ ကဲဒီတော့ ဆက်လက်ပြီးတော့ ကျွော်တော်တို့အနေနဲ့ Java Language ပဲ စွမ်းဆောင်ရည်တွေကို ဆက်လက်လေ့လာ နိုင်ပဲ့။ Sub- title အနေနဲ့ရေးသားလိုအပူးတဲ့ 2.2.1 Basic Input and Output features In Java ဆိုတဲ့ ခေါင်စဉ်ကျော်လေး။ ကို ဆက်လက်လေ့လာကြပါစို့။ Java Language ပဲ Input and Output features တွေအကြောင်းကို အခြေခံကျကျ Introduce လုပ်ပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

Kyaw Zayar Lay | IT Engineering ]



## . . 1 i p p f JAVA

ဒီအခန်းကဏ္ဍမှာ အလွန်အရေးပါတဲ့ **Input and Output** typeတွေကို JAVA Language ဘာ သယ်လိုအသုံးချဖတယ်ဆိတာ ကို သင်ကြားပိုချမှာ ဖြစ်ပါတယ်။ ရွှေမှာ ကျွန်ုတ်တော်သင်ကြားပိုချ ခဲ့တဲ့ listing တွေမှာ sample code ဘွား Data Input ပုံတဲ့ type ဟာ ပါဝင်ခြင်း ရရှိသေးပါဘူး။ ကျွန်ုတ်တော်အနေနဲ့ JAVA Language မျှ အခြေခံ features တွေကို ကျမ်းကျမ်းကျင်ကျင် တတ် ဖြောက်နားလည်တဲ့ အချိန်ကို ရောက်မှ ဒီအခန်းကဏ္ဍကို သင်ကြားပိုချမယ်လို့ ရည်ရွယ်ခဲ့လိုပါပဲ။ JAVA ဟာ အဆင့်မြင့် High-Level-Language ဖြစ်တာကြောင့် Standard Input and Output type ပတ်သက်ပြီး အလွန်အဆင့်မြင့်တဲ့ features ကို ရိုင်ဆိုပါတယ်။ ဥပမာအနေနဲ့ ပြောရရင်တော့ Input Streams အပိုင်းမှာတင် Byte Streams နဲ့ Character Streams ဆိုတဲ့ Group နှစ်ခုကွဲသွားပြီနောက်။ ဒီ Group နှစ်ခုဟာလည်း အသုံးပြုတဲ့ package အလိုက် ကွဲပြားမြားမှာတွေ ရှိနေပါတယ်။ ဥပမာအားဖြင့် ပြောရမယ်ဆိုရင် Byte Streams အပိုင်းကိုပါ အသုံးပြုမယ်ဆိုရင် java .io package တို့ ခေါ်ယူပြီး Declare လုပ်ခဲ့ရမှာပါ။ ဒီအခါမှာ java.io package တို့ import လုပ်ပြီးတော့ -

- FileInputStream Class
- BufferedInputStream Class
- DataInputStream Class
- ByteArrayInputStream Class

စတာတွေဟာ invoked and compute type လာပါတယ်။ ဘာကြောင့်လဲဆိုတော့ ဒီ class တွေဟာ java.io package အောက်မှာ ဖန်တီးရေးသားထားတဲ့ object တွေအဖြစ် ရိုတည်နေ ရတာဖြစ်လိုပါပဲ။ ဒါဟာ ကျွန်ုတ်အနေနဲ့ အကြမ်းဖျမ်းပဲ ရှင်းပြတာနော်။ ဒီအခန်းဟာ Input type အကြောင်းကို Introduce လုပ်တာပဲဖြစ်လိုပါ။ ဒီအခါမှာ စာဖတ်သွားအနေနဲ့ ရှုပ်ထွေးသွားဖွှေ ရှိတဲ့ FilterInputStream အကြောင်းနဲ့ ObjectInputStream အကြောင်းကို ဖြော်ပယ်ထားခဲ့ပါတယ်။ FilterInputStream ပိုင်ဆိုင်တဲ့ BufferedInputStream နဲ့ DataInputStream အကြောင်းကိုတော့ အပေါ်မှာရေးပြခဲ့တယ်ဆိုတော့ ခေါင်းစဉ် အနေနဲ့ပဲ FilterInputStream အကြောင်းကို ဖြော်ပယ်ခဲ့တာပါ။ ဒါဟာ Byte Stream ပတ်သက်တဲ့ အခန်းကဏ္ဍပါပဲ။

ဆက်လက်ပြီးတော့ Character Streams အပိုင်းကို လေ့လာ ကြည့်မယ်ဆိုရင်လည်း java.io package တို့ပဲ use လုပ်ပြီး တော့ File Reader Class  
Buffered Reader Class  
InputStreamReader Class

ဆိုပြီး input type အပိုင်းမှာတင် features တွေ အချိုးချိုး ကွဲပြားမြားနေပါတယ်။ ဒါဟာ Input အပိုင်းကိုပါ အပ်စွဲ ပြီး ရှင်းလင်းပြနေတာနော်။ Output အပိုင်းမှာဆိုရင်လဲ features တွေ အမြောက်အမြောပါဝင်နေပါတယ်။ ဒါဟာ အခြေခံ လောက်အနေနဲ့ပဲ ဒီအခန်းမှာ ရှင်းလင်းပြေားတာပါ။

**Note:** ဒီအခန်းမှာ Byte Streams and Character Streams အပိုင်းနဲ့ ပတ်သက်ပြီး ရှင်းလင်းပိုချမှာ မဟုတ်ပါဘူး။ ရွှေလာမယ့် Chapter တွေမှာ သက်ဆိုင်ရာအလိုက် ထည့်သွင်းပိုချပေးသွားမှာ ဖြစ်ပါတယ်။ သိုးသနဲ့ ခေါင်းစဉ်အနေနဲ့လဲ ရှင်းလင်းပိုချပေးသွားမှာပါ။

C & C ++ မှာ ကျွန်ုတ်တော်တို့ ခက်ခက်ခဲ့ လုပ်ယူရတဲ့ အကြောင်းအရာ အလုပ်ကိစ္စတွေကို JAVA Language တို့ အသုံးပြုပြီးတော့ အလွယ်တကူ ဖြေရှင်းလို့ရပြီဖြစ်ပါတယ်။ က ဒီတော့ JAVA Language တို့ အဆင့်မြင့် Input and Output features တွေကို မလေ့လာခင်မှာ လက်ရှိရေးသားနေတဲ့ program တွေမှာ သင်ယူလေ့လာစဉ် အသုံးပြန်မယ့် standard Input and Output အပိုင်းကို လေ့လာကြရအောင်လား။

## p p vi

Java Language ဘာ C & C ++ ဆန်တဲ့ အတွက် Standard Output device တို့ အသုံးပြုရာမှာတော့ သိပ်ပြီးတော့ ခက်ခဲ့မှာ မရှိပါဘူး။ ကျွန်ုတ်တို့ရဲ့ family ဖြစ်နေပြီဖြစ်တဲ့ print Statement ပြီးတဲ့ System.out.print( ) statement <OR> System.out.println( ) statement method တို့ ခေါ်ယူ အသုံးပြုတာပါပဲ။ ဒါဟာ calculate Data <OR> Streams တွေကို Output လုပ်ပေးပိုအတွက်ပါပဲ။

က ဒါဆိုရင်တော့ ကျွန်ုတ် တစ်ခါမျှ အသုံးပြု ပိုချမြင်းမရှိ သေးတဲ့ Standard Input device method ဘာကို အသုံးပြုရမှာလဲ? ကျွန်ုတ်အနေနဲ့ ရွှေကသင်ယူခဲ့တဲ့ Fix on set data program တွေလို မဟုတ်ဘဲနဲ့ Input data <OR> streams

တွေကို သွင်းယူပြီးမှ calculateလုပ်မယ်။ ပြီးမှ Outputအတွက် တွက်ထုတ်ဖို့ဆိုရင် Standard Input deviceကို သိရှိထားရမှာ ဖြစ်ပါတယ်။

ဒီအခါမှာ Standard Input deviceတော့ C & C++ နဲ့ ဆင်တူမှု မရပါဘူး။ ဘူး၏ input typeဟာ Dialog Box တစ်ခုကနေပြီးတော့ Dataကို သွင်းယူရတာပါပဲ။ ကဲ အဲဒီ InputDialog Boxကို ဖန်တီးခေါ်ယူ အသံပြုရတာဟာ အလွန် လွယ်ကူပါတယ်။ ကဲ ဒီတော့ Sampleအနေနဲ့ ရေးပြပါမယ်။ ဒီ methodဟာ javax.swing packageက methodမဖြစ်ပါတယ်။ လေလာကြည့်ပါ။

#### Sample

```
JOptionPane. ShowInputDialog("promptString")
```

Dialog Box တွင် ပါဝင်စေလိုသောစာသား (text) က ဒါဆိုရင် ကျွန်တော် sample code lineတော်ကြောင်းရေးပြမယ်။ အဲဒါကတော့ အမည်ကို ဖတ်ယူစေဖို့အတွက် String data typeကို အသံပြုပြီးတော့ nameဆိုတဲ့ variableအို့ Declare လုပ်ခဲ့မယ်။ ဒီအခါ ကျွန်တော်တို့က program Runနေစဉ် InputDialog Boxပေါ်လာခဲ့တဲ့အခါမှာ ကိုယ်ထည့်သွင်းချင်တဲ့ Nameအို့ထည့်သွင်းယူမယ်လေ။ ကဲလေလာလိုက်ပါဦး။ အလွယ် ကူခံးဖြစ်အောင် ရှင်းလင်းပို့ချေပေးထားပါတယ်။

#### Sample

String ↑ data type	name = JOptionPane. ShowInputDialog ("what is your name?") ↑ variable	JavaX.swing package နဲ့ Declare လိုအပ်နေရန်နှင့်	Dialog box တွင်သိမ်း text အောက်သွေးပဲ Double code နေရာအောင်ရှင်းလင်း
--------------------------	--	---	---

ဒီအပေါ်မှာ ရေးပြခဲ့တဲ့ Sample code lineလား ရေးပြီး Run လိုက်ပြ ဆိုပါစို့။ အောက်မှာဖော်ပြထားတဲ့ InputDialog Box ကို တွေ့ရမှာပါ။ Figure 2.2.1အနေနဲ့ ဖော်ပြပေးထားပါတယ်။

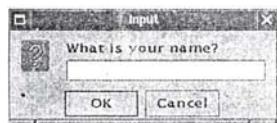


Figure 2.2.1 Show InputDialog Box in Java programming language

JAVA Language၊ standard Input deviceအား languageအတွက် အသံပြုရလွယ်ကြပြီးတော့ ကျွန်တော်တို့ လိုချင်တဲ့ windows featuresအို့ အလွယ်ကူခံးဖန်တီးနိုင်တာကို တွေ့ရမှာပါ။

**NOTE:** Microsoft visual Basic Languageအသံပြန်သူမှာ သူအဖို့ကတော့ ဒီစွမ်းရည်မျိုးကို သိပ်ပြီး ထူးဆန်းတယ်လို့ ထင်မြင်မှုမှ မဟုတ်ပါဘူး။ VB မှာကတော့ လိုချင်တဲ့အရာကို Libraryအဲ အကူအညီနဲ့ ဖန်တီးယူလို့ရတာပဲ မဟုတ်လား။ ဒါပေမဲ့

C & C++ပတ်ဝန်းကျင်မှာ ကြီးပြင်းခဲ့တဲ့သူတွေအဖို့ကတော့ အလွန်တရာကျော်ဖုန်းဖြစ်တဲ့ featuresအို့ရရှိတာပဲ ဖြစ်မှာပါ။

ဒါအပြင် Microsoft Access (DataBase Language)

ကို အသံပြုမှုနေတဲ့ programmerအတွက်လည်း သိပ်ပြီး က်ခက်ခဲ့ခဲ့ လောကျင့်ရောသားစာလိမ်းလို့ မထင်မိပါဘူး။ ဒီလို လွယ်လွယ်ကူကူ Input Dialog Boxကို ဖန်တီးနိုင်တဲ့ featuresဟာ MS Accessအား Design viewအသွားရောက်ဖန်တီးရေးသားလိုက်ရှိလို့ အလုပ်မျိုးစွမ်းရည်ဖြင့် Input Dialog Box ကို အလွယ်တကူဖန်တီးရယူနိုင်ပါတယ်။ သဘောတရားခြင်း နှင့် ယုံးမှုလွယ်ဆိုရင် VB Languageအဲ သဘောတရားဂွဲပြားခြားနား ပြီးတော့ MS Accessအဲ သဘောတရားနဲ့ ဆင်တူနေပါတယ်။ ဒါပေမဲ့ စွမ်းရည်မြင့်ဖန်တီးဆိုရှိကတော့ JAVA Languageမှာပဲ ရှိပြီးတော့ အခုန့်ငါးယုံးပြခဲ့တဲ့ VB တို့၊ Accessအို့က JAVA Languageမေးတဲ့ ယုံးပြုပြင်နိုင်မှု မဟုတ်ပါဘူး။ သဘောတရားခြင်း ဆင်တူတယ်ဆိုတဲ့အချက်ကို နှင့်ယုံးပြတာပါ။

ဒါဆိုရင်ပေါ်လာတဲ့ ဒါ Input Dialog Boxအို့တွေ့ရတဲ့အခါ မှာ ပါဝင်တဲ့ Text Boxကိုလည်းတွေ့ရမှာပါ။ အဲဒီ Text Box ထဲမှာ ကိုယ်ဖြည့်သွင်းချင်တဲ့ Data <Streamsအဲ Inputလုပ်ဖြည့်သွင်းပြီး OK Buttonကို Clickလုပ်ရပါပဲ။ ဒါမှမဟုတ်လဲ keyboardအား Enter keyကို ဖြစ်ဖြစ်နိုင်လိုက်ရှုပေါ့။ ဒါဆိုရင် အရှေ့မှာရေးပြခဲ့တဲ့ Sample codeအဲ Input Dialog Boxကို ယုံးပြုလေလာပြီးရင်တော့ Listing 2.9အနေနဲ့ ရေးသားပို့ချေပေးထားတဲ့ programလေးကို လောက်ကြည့်ပြုရအောင်။ ဒီအခါ ပို့ချေမယ့် programလေးကိုသာ အမှန်တကယ် နားလည်တတ်မြောက်အောင် လောက်ပြီးသွားရင်တော့ ရှေ့သင်ခန်းစာတွေမှာ လေးလာသင်ယူခဲ့တဲ့ listingတွေကို ပြန်လုပ်ပြီး upgradeလုပ်ပြီးတော့ ထပ်မံဖြည့်စွက်ပြပြင်နိုင်မှာပါ။ သာကြောင့်လဲဆိုတော့ ရှေ့မှာသင်ကြားခဲ့တဲ့ programတွေဟာ dataတွေကို set orပြီးမှ တွက်ချက်ထားတဲ့ programထုပါ။ Input Dialog Boxတွေ ဖန်တီးမယ်။ ပြီးတော့မှ Dataကို Inputလုပ်ပြီးတော့ compile calculate fieldပုံပြီး output dataအုတ်ပေးတဲ့ Structureကို လောကျင့်ပါ။ ဒါဟာ JAVA programmerကောင်းတစ်ယောက်ဖြစ်စေမယ့် လမ်းစတုပဲ။ ဒီလိုချို့ listing တွေကို ပြန်လုပ်ဖြည့်စွက်ပြီးတော့ လောကျင့်ခြင်းကို Tuforialပုံစံ ကိုယ်လုပ်မယ်ဆိုရင် ထက်မြောက်တဲ့ JAVA Developerကောင်းတစ်ယောက်ဖြစ်လာမှာပါ။ ကဲ အောက်မှာရေးထားတဲ့ listing 2.9အနေနဲ့ရေးပြထားတဲ့ programအို့လောက်လုပ်ပြုတဲ့ပါ။ သာမန် input & output typeအို့ပဲ ရှင်းလင်းပြတာပါ။

Listing 2.9 Input Dialog Box demonstration in JAVA Language

```
1. imp j v x. wi g. *; // D i
   j v x. wi g.*; p k g
```

```

1. p i p D { i g [] g }
2. p i i v i m i ( i g [ ] g ) {
3. // i p i g fi
4. i g m = J p i . h w p Di
   g ("Wh i y m ?");
5.
6.
7. // i p i g
8. i g = J p i . h w p Di g
   ("H w y ?");
9. // v i g i g v
10. i g = g .p ( );
11. y m. .p i ("m :" + m );
12. y m. .p i ("Ag :" + g + "y ");
13. y m. .p i ("H !" + m + "
   y , y ' " + ( g + 1 ) + "y ");
14.
15. y m. xi (0);
16. }
17. }
```

### A y i

၁။ လိုင်းနံပါတ် (၁) မှာ javax.swing.\*;package ခေါ်ယူပြီးတော့ Declareလုပ်ခဲ့ပါတယ်။ ဒါမှသာ Input Dialog Box ကို ဖန်တီးယူလို့ရမှာ ဖြစ်ပါတယ်။ စာဖတ်သူ အနေနဲ့ အလွယ်တကူနားလည်နိုင်အောင်လို့ Double backslash မျှော်း comments အနေနဲ့ ရေးသားပေးထားပါတယ်။ လိုင်းနံပါတ် (၂) မှာတော့ Input Date လို့ပါ၊ primary class ရှိရေးသားပြီး open brace ကို ဖွင့်ထားပါတယ်။ ဒါကျေဟာ သာမန်လုပ်ရှိုးလုပ်စဉ် အတိုင်း programဘစ်ခုကို ဖန်တီးတာပါပဲ။

၂။ လိုင်းနံပါတ် ၃ မှာတော့ main () function ရှေ့ဖွဲ့ထားပါတယ်။ အဖွင့် brace { ကိုလေတွေ့ရမှာပါ။ လိုင်းနံပါတ် (၄) ကတော့ Double backslashမျှော်း comments ရှိထားတာပါ။ ဒီ line ကို ထည့်မရေးရင်လည်း ရတယ်နော်။ ပြီးတဲ့အော့ String data type အော့ပြုပြီးတော့ name ဆိုတဲ့ variable ကို Declareလုပ်ပါတယ်။ အလုပ်လုပ်ရင်း စာတို့တဲ့ variable ကို Declareလုပ်တာပါ။ တကယ်လို့ စာဖတ်သူဟာ ရှေ့မှာရေးပြုတဲ့ programming sample type ရွှေ့ကိုပဲ ရှုံးနှုံးနေရင် ဒီပုံးပါဘာ အနည်းငယ် complexဖြစ်နေနိုင်ပါတယ်။ ဒါအော့မှာ ပုံစံကို အခုလုပ်ပြောင်းရေးဖွဲ့ရင်လည်း ရပါတယ်။

```

String name; // String data type use in variable
declare
    name = JOptionPane.showInputDialog("what is
your name?");

က ဒီပုံးဆိုရင် ရေးနေကျပုံးဖြစ်ပါတယ်။ အတွက် အလွယ်တကူ
နားလည်နိုင်တာပေါ့။ စာဖတ်သူအနေနဲ့ နှစ်သက်သလို ပြောင်းလ
ပြီး ရေးဖွဲ့နိုင်ပါတယ်။ ဒီ programမှာတော့ Dialog Box ကို
တစ်ဆက်တည်း ခေါ်ယူလိုက်တာပါ။ အဲဒါကြောင့် လိုင်းနံပါတ်
```

(၅) မှာ String name = JOptionPane .ShowDialog
("What is your name?")လို့ ရေးသားထားတာပါ။ ကျွန်တော်
အရှေ့မှာရေးပြုခဲ့တဲ့ Input Dialog Box လိုပါပဲ ပေါ်လာမှာပါ။
ဒါအော့ ကျွန်တော် တို့ ထည့်သွင်းချင်တဲ့ Data လို့ Input Dia
log Box က text Box ထဲမှာ ဖြည့်သွင်းပြီး OK Button ကို
နှိပ်ရမှာပါ။ ဒါမှမဟုတ် keyboard က Enterကို pressလုပ်ရင်
လည်း အတူတူပါပဲ။

၃။ လိုင်းနံပါတ် (၆) ကတော့ code line တွေကို ရေးသားတဲ့
အော့မှာ ရှင်းလင်းအောင်လို့ free line space လုပ်ခဲ့တာပါ။
မချိန်ရင်လည်းရပါတယ်။ အဲဒီနောက် လိုင်းနံပါတ် (၇) မှာတော့
comments ရှိ ရေးသားပေးထားပါတယ်။ ပြီးတဲ့အော့မှာမှာ
String data type အော့ပြုပြီးတော့ old ဆိုတဲ့ variable
ကို Declareလုပ်ပြီး Input Dialog Box ကို ခေါ်ယူလိုက်ပါ
တယ်။ အဲဒါကြောင့် How old are youဆိုတဲ့ Input Dialog
Box ပေါ်လာတာဖြစ်ပါတယ်။ ဒါအော့ ကျွန်တော်တို့ ဖြည့်သွင်းချင်
တဲ့ Data ကို ထည့်သွင်းပြီးရင် keyboard က Enter key ကို
ဖြစ်စေ၊ Input Dialog Box မှာပါဝင်တဲ့ OK Button ဖြစ်စေ
pressလုပ်ရပါတယ်။ ဒါအော့ ကျွန်တော်တို့ သတိပြုရမှာကတော့
How old are youဆိုတဲ့ Input Dialog Box မှာ data
Input လုပ်စဉ်က ထည့်သွင်းလိုက်တဲ့အသက်ကို String data
type နဲ့ အသုံးပြုပြီး လက်ခံတော့နော်။ ဒါအတိုင်း Output ပြန်
ထွေတိရင်လည်းရပါတယ်။ ဒါပေမဲ့ JAVA Language မှာ ဂိုဏ်း
ရင်းနှီးကျမ်းဝင်သွားအောင် လက်ခံထားတဲ့ String data type
ကို int data type အဖြစ် convertလုပ်လိုက်ကြရအောင်။ ဒါအော့
int data type ကို အသုံးပြုပြီးတော့ age ဆိုတဲ့ variable ကို
Declareလုပ်ရမှာပါ။ ဒါကြောင့် လိုင်းနံပါတ် (၁၀) မှာ int
age = Integer.parseInt (old) ပြီး အပေါ်က လက်ခံခဲ့တဲ့
Data ကို int data type ဖြစ်အောင် convertလုပ်ပါတယ်။
ဒါအော့ data ကို ထည့်သွင်းစဉ်က string data type အနေနဲ့
ပေမယ့် အခုအချိန်မှာ int data type အဖြစ် ပြောင်းလဲသွားပါပြီ။

၄။ လိုင်းနံပါတ် (၁၁) မှာတော့ လက်ခံထားတဲ့ Name ဆိုတဲ့
data ကို print statement အော့ပြုပါတယ်။ Output ထွေတိပါတယ်။
ဒါအော့ Name: Input\_Name ကော့ လာပြပေးမှာပါ။ ဒါအော့မှာ
လာပြီး output ထွေတိပေးနေကျအတိုင်းပဲ console window
မှာ လာပြပေးမှာပါပဲ။ Dialog ပုံးနှင့် မဟုတ်ပါဘူး။ ဒါကိုတော့
ပို့ပြီးအတွက်တဲ့ ရှင်းပြနေစရာမလိုပါဘူး။ အရင် listing
တွေမှာ သင်ကြေးမျှ ပေးထားတဲ့ ပုံစံဖြစ်နေလိုပါပဲ။ အဲဒီနောက်
လိုင်းနံပါတ် (၁၂) မှာတော့ လက်ခံထားအသက်ဖြစ်တဲ့ data ကို
Age: Input\_age အဖြစ် output လာပြပေးမှာပါပဲ။ သတိပြုရ
မှာကတော့ ထည့်သွင်းစဉ်ကအသက်ဟာ String data နဲ့ old
ဆိုတဲ့ variable ဖြစ်ပေမဲ့ output ထွေတိပေးတဲ့အချိန်မှာတော့

`integer data typeအဖြစ် convertလုပ်ပြီး ageဆိုပြီး  
လာပြေးတာပါ။ ဒါကြောင့် လိုင်းနံပါတ် (၁၂) မှာ -`

```
y m. .p i ("Ag :" + g + "y ");  
လို့ရေးသားထားတာပါပဲ။
```

၅။ အဲဒီနောက် လိုင်းနံပါတ်(၁၃) မှာတော့ ရှုံးလာမယ့်နစ်မှာ

သင်ဟာ အသက်တစ်နှစ်ကြီးလာမယ်ဆိုတာကို ယခု ရိုက်  
သွင်းလိုက်တဲ့ `data`ဘို့ `age + 1`ပြုပြီးတော့ တစ်နှစ်တိုးလိုက်ပါ

တယ်။ တကယ်လို့ သင်ဟာ `Increment operator`ခန်းကဲ့  
က `postfix`နဲ့ `prefix`ကိုသာ နားလည်တတ်ကျမ်းခဲ့ရင် (`age+1`)  
လို့ မရေးပဲနဲ့ (`+ + age`)လို့ ရေးလိုက်ရင်လည်း ရမှာလို့ နားလည်  
သဘောပေါ်ကိုမိမှုပါ။ ဒါမှမဟုတ်ဘဲနဲ့ အပေါ်တစ်ကြောင်းမှာ  
`age ++` လို့ ရော့ပြီးမှ အောက်တစ်ကြောင်းမှာမှ `age;`လို့  
ခေါ်ယူလိုက်ရင်လည်း အသက်တစ်နှစ်တိုးပြီးသားဖြစ်မှုပါ။ အခု  
ရှင်းပြတာကို နားမလည်ခဲ့ရင် အရင်ပို့ချခဲ့တဲ့ 1.2 Decision,  
control structure and looping statements of JAVA  
အခန်းကဲ့ကို ပြန်ပြီးလေ့လာကြည့်ပါ။

အဲဒီနောက် လိုင်းနံပါတ် (၁၅) မှာတော့ အရင်က ကျွန်တော်  
တို့ အသုံးမပြုဖူးသေးတဲ့ `System.exit (0)`နဲ့တာကို တွေ့ရမှာ  
ပါ။ ဒါ `code line`လေးဟာ ကျွန်တော်တို့ဟာ `JOptionPane.  
ShowInputDialog Box`နဲ့ ခေါ်ယူအသုံးပြုတိုင်း `program  
code`အဆုံးမှာ `System.exit (0)`နဲ့တဲ့ `code`နဲ့ လာပိတ်ပေး  
ရပါတယ်။

ဒါဟာ အခန်းတစ်ခုကို ဖွံ့ဖြိုးဝင်ရောက်ခဲ့မယ်ဆိုရင် အလုပ်  
ပြီးလို့ ပြန်ထွက်တဲ့အခါမှာ အခန်းတံခါးကို ပြန်ပိတ်ပေးရသလို  
ပါပဲ။ ဒါ `code line`နဲ့ ပတ်သက်ပြီး အောက်မှာ သေသေချာချာ  
ရှင်းပြေးသေးပါတယ်။ အခုတော့ `Input Dialog Box`နဲ့  
ခေါ်ယူအသုံးပြုရင် `System.exit (0)`နဲ့တဲ့ `code`နဲ့ လာပိတ်  
ပေးရတယ်လို့ပဲ အလွယ်မှတ်ပါ။ လိုင်းနံပါတ် (၁၆) မှာ `main  
( ) function`၊ `close brace` ဘို့ လာပိတ်ပေးသေးပါတယ်။  
လိုင်းနံပါတ် (၁၇) ကတော့ `primary class`၊ `close brace }`  
ကို လာပိတ်ထားပါတယ်။ ဒါအခါ ဒါ `program`နဲ့ `Run& com-  
pile`လုပ်ပြီးလေ့လာမယ်ဆိုရင် `data`ဘို့ `Input`အဖြစ်လက်ခံစဉ်  
က `Dialog Box`နဲ့ လက်ခံပြီးတော့ `output`ကိုတော့ `con-  
sole window`နဲ့ လာပြေးမှုပါ။ `Dialog Box`နဲ့ မဟုတ်ပါဘူး။  
သေသေချာချာလေ့လာကြည့်ပါ။ အလွယ်ကူးအုံးဖြစ်အောင်ရေး  
ပြထားပါတယ်။

ဒါ `program`လေးဟာ JAVA S DK 1.4အထိ standard versionအိုင်းမှာ Demonstrateလုပ်ပြတဲ့ `program  
type`ပုံစံပါပဲ။ windows featuresတွေဖြစ်တဲ့ GUI Envi-  
ronmentတွေ ပါဝင်မှု ဘာမျှမရှိသေးပေမဲ့ C & C + + Lan-  
guageနဲ့ နယ်ပယ်မှာကြီးပြင်းလာတဲ့ ကျွန်တော်တို့အတွက်တော့

ကျေန့်ဖွယ်ရာ `features`တစ်ခုအဖြစ် လက်ခံနိုင်ကြပါ။ ဒါ  
အပေါ်က `program`လေးကို လေ့လာပြီးတဲ့အခါမှာ အောက်  
မှာ သီးသန့်ရှင်းပြထားတဲ့ JAVA API methodလေးကို လေ့လာ  
ကြည့်ပါ။ ဆက်နွယ်တဲ့ သဘောတရားကို အတတ်နိုင်ခဲ့းနားလည်  
အောင် ရှင်းပြေးသေးတာပါပဲ။ `code`တစ်ကြောင်းကိုရေးတော့  
မယ်ဆိုရင် ဘာကြောင့်ရေးရတာလဲ။ ဘယ်လိုစွမ်းရည်လဲ၊ ဘာ  
ကြောင့်လဲဆိုတာကို အမှန်တကယ် သိနားလည်စေချင်လိုပါပဲ။

`i i g h w p Di g ( j m g )`  
ဒါ JAVA API methodနဲ့ အသုံးပြုတဲ့အခါမှာ Input text  
တွေကို လက်ခံနိုင် Text Boxတစ်ခု OK & cancel Button  
နှစ်ခုပါဝင်တဲ့ `InputDialog Box`တစ်ခုလာပြေးမှုပါ။ ဒီစနစ်  
ကို အဓမ္မ စတင်အသုံးပြုပေမဲ့ ကျွန်တော်တို့အဖို့ အလွယ်တာကူ  
ကွဲ့ဝင်သွားမှုပါ။ ဒါပေမဲ့ ကျွန်တော်တို့ အထူးသတိပြုရမယ့်  
အချက်ကတော့ `System.exit(0)`နဲ့တဲ့ code line API function  
လေးနဲ့ `program`ကို အဆုံးသတ်ပေးရပါတယ်။ ဒါဟာ  
`program`တစ်ခုကို ရေးသားတဲ့အခါမှာ importလုပ် အသုံးပြု  
ထားတဲ့ import javax.swing.\* ; packageနဲ့ အဆုံးသတ်ဖို့  
ထိန်းချုပ်ထားတဲ့ Multi-threaded featuresနဲ့ exit (0)လုပ်  
ခိုင်းတာပါပဲ။ ဒါအခါမှာ system.exit (0) ထဲမှာပါဝင်တဲ့  
parameter valueနဲ့ int status valueတစ်ခုအနေနဲ့လည်း  
ထည့်သွင်းအသုံးပြုနိုင်ပါတယ်။

ကဲ့ဒီအခန်းကဲ့မှာကတော့ Basic Input and Output  
featuresတွေကို ဘယ်လိုအသုံးပြုမယ်ဆိုတာကို အတတ်နိုင်ခဲ့း  
ရှင်းလင်းပို့ချာပေးသေးပါတယ်။ ကျွန်တော်အရေးမှာ ပြောခဲ့သလို  
ပါပဲ။ တကယ်လို့ အချိန်ပေးပြီး အရင် listing တွေကို ပြန်လည်  
ပြီး Input featuresတွေကိုသာ ထပ်ဖြည့်ရေးသားမယ်။  
Flowchartတွေ ရေးဆွဲကြည့်မယ်။ Algorithm ရေးသားစီစဉ်  
ကြည့်မယ်ဆိုရင် JAVA အတွက် ပြည့်စုံတဲ့ အလေ့အကျင့်တွေ  
ရရှိမယ်ဆိုတာ မမှားပါဘူး။

ကျွန်တော်အနေနဲ့လည်း Chapter [2] အပြီးမှာ Tutorial  
အနေနဲ့ Testလုပ်စံဆေးချင်ပါတယ်။ အဲဒီအခါမှာ ဒီသင်္ကြား  
ပို့ချာထားတဲ့ listing တွေကိုလည်း ရောထွေးပြီး combine & test  
လုပ်ချင်ပါတယ်။ သီးသန့်ခက်ခဲတဲ့ program typeတွေကို မေးမှာ  
မဟုတ်ပါဘူး။ အဲဒီကြောင့် အချိန်ရတဲ့အချိက်မှာ တတ်နိုင်  
သလောက် Alogrithm & Flowchartအပိုင်းတွေကိုရေးပြန်လှန်  
လေ့လာထားစေချင်ပါတယ်။

ရှုံးလမှာတော့ chapter [2] နဲ့ index, sub-titleတစ်ခု  
ဖြစ်တဲ့ 2.3 Access Methods and Passing Arguments  
In JAVA ဆိုတဲ့ အခန်းကဲ့ကဲ့ဆက်လက်လေ့လာကြပါစို့။  
ကျော်ဇူးလေး(ကသာ)



### 2.3 Access Methods and Passing Arguments In JAVA

ဒီခေါင်းစဉ်၏ယောက်လေးဟာ JAVA Language ရဲ့ OOP အခန်းကဏ္ဍကို လေ့လာနေတဲ့ ကျွန်တော်တိအတွက် အရေးကြီးတဲ့ ခေါင်းစဉ်၏ယောက်လေးတစ်ခုထိရင် မမှားပါဘူး။ ဘာကြောင့်လဲဆိုတော့ programming လုပ်တယ်ဆိုတာ အမှန်တကယ်ကတော့ အလွန် တရာ့မှ စနစ်ကျွေတဲ့ အစီအစဉ်တစ်ခုကို ကြိုးတင်တွက်ချက်ရေးသား ရခြင်းပဲ ဖြစ်ပါတယ်။ ဒီတော့ ဒီနေရာမှာ ကျွန်တော်တိအတွက် လိုအပ်တဲ့ Access Methods တွေအကြောင်းရယ် passing Arguments (ရွှေ့မှာ အနည်းငယ်ပုံးချပ်ပြီးပါပြီ။) အကြောင်း တွေရယ်ကိုမှ နားမလည်ရင် အစီအစဉ်ကျွေနေတဲ့ program တစ်ခု ကို ဖန်တီးရေးသားနိုင်မှာ မဟုတ်ပါဘူး။ ကဲ့ဒေါ် အောက် Access Methods တွေကို အရင်လေ့လာကြရအောင်။ Access Method ကိုလေ့လာမယ်ဆိုရင် အောက်မှာရေးပြထားတဲ့ အခြေခံ Methods အုပ်စု (၃) ခုကို တွေ့ရမှာပါ။ အဲဒါတွေကို အောက်မှာရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။

Static Methods

Final Methods

Abstract Methods

ကဲ့ဒေါ် Methods အုပ်စုတွေကို တွေ့ရပြီဆိုရင် Methods တစ်ခု ချင်းစီရဲ့ အသုံးဝင်ပုံ usage နဲ့ ထည့်သွင်းရေးသားအသုံးပြုပုံ တွေအကြောင်းကို လေ့လာကြရအောင်။ ဒီတော့ Static Methods အကြောင်းကနေစပြီး လေ့လာကြတာပေါ့။ လေ့လာကြည့်ပါ။

#### Static Methods

Static Methods ဆိုတာဟာ ကျွန်တော်တိ ဖန်တီးရေးသား ထားတဲ့ class ကို အခြားတစ်နေရာမှနေ၍ လုမ်းပြီးခေါ်ယူအသုံးပြုတဲ့အဓိက အလွန်အသုံးဝင်တဲ့ Methods တစ်ခုဖြစ်ပါတယ်။ Static Methods ကို အသုံးပြုခြင်းအားဖြင့် ဖန်တီးရေးသားထားတဲ့ class ကို object အနေနဲ့ ဖန်တီးစရာမလိုဘဲ main () function / Method ထဲမှ လိုက်ရိုက်လုမ်းပြီး ခေါ်ယူအသုံးပြုလို့ရပါတယ်။ ဒီအချက်ဟာ C ++ ထက်သာတဲ့ JAVA ရဲ့ အားသာချက် အချို့ပါပဲ။ Static Methods ရဲ့အမို့ယွင့်ဆို ချက်ကို JAVA Language နဲ့ ပတ်သက်တဲ့ ရည်ညွှန်းစာအုပ်

တွေမှာ အောက်ပါအတိုင်း အမို့ယွင့်ဆိုထားပါတယ်။ လေ့လာကြည့်ပါပြီး။

Static Method is assigned to a class and not to an object of that class.

The syntax for calling a static method is the method name plus whatever arguments the method requires.

ဒီ Static Method ကို အသုံးပြုခြင်းအားဖြင့် C ++ Language မှာလို object အဖြစ် ဖန်တီးရာကန် မလိုလားအပ်တဲ့ မှားယွင်းမှုတွေကို ရောင်စွဲလို့ ရနိုင်ပါပြီ။ ဒါဟာ JAVA Language ရဲ့ အားသာချက်ဆိုတာကို သဘောပေါက်ပါ။ Listing 2.10 ကနေစပြီးတော့ Sample program ရေးထားတာကို စတင် လေ့လာကြည့်ရအောင်။ အလွယ်ကူရုံးဖြစ်အောင် ရေးသားပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

#### Listing 2.10 Demonstrate Static Method In JAVA

```
1. public class DemoStatic {
2.     public static void main
3.         (String [ ] args) {
4.             String str = Another Class.
5.             getDate ( );
6.             / * -----
----- *
7.             call the class to use but
8.             without create
9.             on object of the class
10.            ----- * /
11.             System .Out. print In (Str);
12.         }
13.         Class Another Class {
14.             public static String getDate
15.             ( )
16.             { return "Hello! Kyaw Zayar
17.               Lay. Int Please ,give me cof-
18.               fee."; }
19.         }
20.     }
```

### Output

Hello! Kyaw Zayar Lay.

Please, give me coffee.

**Note:** ဒီ Listing 2.10 မှာရေးသားထားတဲ့ Program မှာသတိပြုရမှာကတော့ လိုင်းနံပါတ် (၄) ကနေ လိုင်းနံပါတ် (၇) အထိရေးသားထားတဲ့ဟာကတော့ comments တွေပဲ ဖြစ်ပါတယ်။ ဒီအခါမှာ comments လို့ခေါ်လိုက်လို့ စာဖတ်သူအနေနဲ့ // Double Backslash ခံပြီး ရေးသားတာမျိုးကိုသာ တွေ့မြင်ဖူးတဲ့သူအဖို့ကတော့ complex ဖြစ်သွားနိုင်ပါတယ်။ အခု လိုင်းနံပါတ် (၄) မှာ ရေးသားထားတဲ့ comments type ကတော့ C style လို့ ပြောရင်တော့ မမှားပါဘူး။ C Language ကို အသုံးပြုဖူးတဲ့ programmer တွေအဖို့ များပြားလှတဲ့ comments တွေကို ရေးသားရတော့မယ်ဆိုရင် Backslash တစ်ခုခံပြီး Indirection operator နဲ့တဲ့ဖောက်ပြီး / \* ပုံစံအတိုင်း အဖွဲ့ရေးသားရပါမယ်။ ပြီးတော့ နောက်ထပ်လို့အပ်သလောက် comments [program code မဟုတ်တဲ့ text တွေ] တို့ ရေးသားပြီး comments အဆုံးမှာ လဲ \*/ သင်္ကာတနဲ့ ပြန်ပိတ်ပေးရပါတယ်။ ဒါဟာ comments အမြောက်အမြားရေးချင်ရင်တော့ ဒီ program မှာရေးတဲ့ type ပုံစံဟာ အဆင်ပြေစေနိုင်မှာပါ။

### Analysis

၁။ ဒီ listing 2.10 မှာရေးသားထားတာဟာ Static Method ကို သရုပ်ပြ ရှင်းလင်းပြတာဖြစ်တာမျို့ အတတ်နိုင်ဆုံး ရှိုးရှင်းအောင် ရေးဖွံ့ဖြိုးထားပါတယ်။ လိုင်းနံပါတ် (၁) မှာ primary Class ကို ရေးဖွံ့ထားပါတယ်။ အဲဒီ primary class ရဲ့ Name ကတော့ DemoStatic ပါ။ အဲဒီနောက် လိုင်းနံပါတ် (၂) မှာမှ မှာမှ main ( ) function ကိုရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၃) မှာမှ String data type ကိုကြည့်လိုပြီးတော့ လိုင်းနံပါတ် (၁၁) ကနေ လိုင်းနံပါတ် (၁၄) အထိအေးဖွံ့ထားတဲ့ Another Class ကို Static Method ကိုအသုံးပြုပြီး တိုက်ရှိလှမ်းခေါ်ယူပြီးတော့ အဲဒီ class ထဲက getdata () function ကို ခေါ်ယူ အသုံးချုပ်လိုပါတယ်။ ဒါအဲရင် ဒီ main () function ကို အရင်လေ့လာမယ်ဆိုရင် အဲဒီ Another Class ကို အရင်လေ့လာမှုရမှာနော်။ ဒီတော့ Another Class ကို အရင် လေ့လာကြည့်ကြရအောင်။

၂။ လိုင်းနံပါတ် (၁၂) က getdata () function ဟာ Another Class ဆိုတဲ့ Class ထဲမှာ ဖန်တီးတည်ဆောက်ခြင်း ခံထားရတာနော်။ အဲဒီ getdate () function ထဲမှာ ဘယ် data type ကိုမျှ ထည့်သွင်းအသုံးပြုထားခြင်းမရှိဘဲ စာသား String ကို Double quote " " နှစ်ခုကြားမှာရေးပြီး main () function ထဲကို return ပြန်ပေးထားပါတယ်။ ဒီအခါမှာ

return ပြန်ပေးထားတဲ့ စာသားကို လေ့လာကြည့်ပါဦး။ "Hello! Kyaw Zayar Lay. InIt Please , give me coffee."

ဒီအခါမှာ Double quote နှစ်ခုကြားမှာရေးထားတဲ့ စာသားမှာ Hello ! Kyaw Zayar Lay ဆိုတဲ့ စာသားအပြီးမှာ In နဲ့ It ဆိုတဲ့ sepecial character [escape character] ကို ထည့်သွင်း ရေးသားထားပါတယ်။ ဒါကြောင့် please, give me coffee,ဆိုတဲ့ string ဟာ အောက်တစ်ကြောင်းကို ဆင်းပြီး output လာပြပေးတာပါ။ ဒါဟာ Another Class ထဲက getData () function ထဲမှာရေးသားထားတဲ့ String ပါပဲ။ return ပြန်ပေးထားတာ ကြောင့် main () function ထဲကို ပြန်သွားရမှာနော်။ ဒါဟာ လိုင်းနံပါတ် (၁၁) ကနေ လိုင်းနံပါတ် (၁၄) ကြားမှာ ရေးသားထားတဲ့ Another Class ကို လေ့လာကြည့်တာပါပဲ။ ဒီအထိ နားလည် သဘောပေါက် ပြနော်။

၃။ ဂဲဒါဆိုရင် main () function ထဲကို ပြန်ကြည့်ရအောင်နော်။ လိုင်းနံပါတ် (၃) မှာ ရေးသားထားတာကတော့ Static Method ကို အသုံးပြုပြီးတော့ getData () function ကို လှမ်းခေါ်ယူ လိုက်ပါတယ်။ ခေါ်ယူတဲ့အခါမှာ object အဖြစ်ဖန်တီးစရာမလိုဘဲ Class ^ Name ကို (. ) dot ခံပြီး ရေးသားခေါ်ယူယံပါပဲ။ ဒါဆိုရင် အဲဒီ getDate () function ထဲက String တွေဟာ main ထဲကို ရောက်လာခဲ့ပြီပဲ။ နော်။ လိုင်းနံပါတ် (၄) ကနေ လိုင်းနံပါတ် (၇) အထိ ရေးသားထားတာကတော့ comments ရေးသားထားတာပါ။ သိုးသနဲ့ Note: အနေနဲ့ အပေါ်မှာ ရှင်ပြပေးခဲ့ပြီးပါပြီ။

၄။ လိုင်းနံပါတ် (၈) မှာတော့ System.out.print In (Str); ဆိုပြီး print statement နဲ့ output ထုတ်ပါတယ်။ ဒီအခါရေးသားထားတဲ့ Str ဆိုတာဟာ getData () function ထဲမှာရေးသားထားတဲ့ String ကို return ပြန်ပေးထားပါ။ return ပြန်ပေးထားတဲ့အတွက် main () function ထဲ သေချာပေါက် ဝင်လာပြီပေါ်နော်။ ဒီအခါ လိုင်းနံပါတ် (၁၁) မှာရေးသားထားတဲ့ Another Class ကို တိုက်ရှိက်လှမ်းခေါ်ယူသုံးလိုက်တာနော်။ ဒါဟာ Static Method ပါပဲ။

**Note:** Static Method ဆိုတာဟာ ရေးသားထားတဲ့ class ကို object အဖြစ် ဖန်တီးစရာမလိုဘဲ Class ^Name ကို . (dot) ခံပြီးလိုချင်တဲ့ function ကို လှမ်းခေါ်ယူလိုက်တာပါ။ ဥပမာ အနေနဲ့ ပြောရင်တော့ အခုရေးခဲ့တဲ့ class - Name က Another Class ဖြစ်ပြီး function ^ Name က getdata () function ပါ။ ဒီတော့ ခေါ်ယူတဲ့အခါမှာ – Another Class. getData () ;

ဆိုပြီးခေါ်ယူရမှာပါ။ စာဖတ်သူအနေနဲ့ complex ဖြစ်သွားနိုင်တာ

ကတော့ လိုင်းနံပါတ် (၃) မှာ ရေးသားထားတဲ့

```
String Str = Another Class . getData ( );
```

လိုဂျေားခဲ့တဲ့အခါမှာ String data type ကိုအသုံးပြုပြီး String variable နဲ့ညီတေားတာကိုပါပဲ။ အမှန်ကတော့ `getData()` function ထဲမှာ စာသားတွေရေားသားခဲ့ခြင်းက Variable Name ကိုပေးပြီး ရေားသားခဲ့တာ မဟုတ်ပါဘူး။ ဒီအခါ Output ထုတ် မယ်ဆိုရင် ဒီစာသားတွေကို ခေါ်ယူတဲ့အခါမှာ ကိုယ်စားလှယ် ကိန်း Variable ရှိထားရမှာပါ။ ဒါကြောင့် Static Method နဲ့ အသုံးပြုစဉ်မှာပဲ String variable ကို Declare လုပ်ခဲ့တာပါ။ ဒါဆိုရင် ဒီအချက်ကို နားလည်သဘောပေါက်ပြီနော်။

၅။ လိုင်းနံပါတ် (၉) မှာ ရေားသားထားတာကတော့ `main()` function ရဲ့ close brace } ပါ။ လိုင်းနံပါတ် (၁၀) မှာမှ primary Class ရဲ့ close brace } ကို လာပိတ်ထားတာပါ။ ဒီလိုအဆင့်ဆင့် ခေါ်ယူရေားသားလိုက်တဲ့အခါမှာတော့ Output ဟာ -

Hello! Kyaw Zayar Lay.

Please, give me coffee.

ဒီ Static Method ကို ကျမ်းကျင့်ပိုင်နိုင်မှသာလျှင် ရှေ့ဆက်လက်အသုံးချရမဲ့ OOP နယ်ပယ်မှာ JAVA Language ဆိတဲ့ tools ကို အသုံးချဖိုင်နှာဖြစ်ပါတယ်။ ဆက်လက်ပြီးတော့ ကျွန်တော်တိအနေနဲ့ final Method ကိုဆက်လက်လေ့လာကြရအောင်လား။

## Final Methods

& object ଟେଗ୍ କି Final Keyword କି ଅଛି ପ୍ରିଣ୍ଟିଂରେ  
Protected ଫୁଲ୍ ଲାଇସ୍ ରୁହିତାଯି। Final Methods କୁ ପରିଚାଳନା କରିବା  
ଅଛିବୁଏବୁ ଫୁଲ୍ ଲାଇସ୍ ରୁହିତାଯି।

\* A Method that is declared as final can't be overridden in a sub- class.

\* The final keyword is written after the access type and before the return type.

ဒါဆိုရင် Final Methods ဟာအချို့သော OOP concepts တွေအတွက် အသုံးဝင်တဲ့ Method တစ်ခုဖြစ်နိုင်တယ်ပဲ။ Note: JAVA Language ရဲ့ အခန်းကဏ္ဍတစ်ခုဖြစ်တဲ့ Access Methods ဆိုတဲ့ အခန်းကဏ္ဍဟာ object-oriented paradigm တွေအတွက် ထိအပ်သလိုဖန်တီးနိုင်မဲ့ features တွေကို ပေါင်းထည့်ထားတဲ့ JAVA ရဲ့ အဆင့်မြင့် OOP စွမ်းရည်ပါလို့ ပြောရင် မမေးပါဘူး။ ဒါ features တွေကို ကျွန်တော်တို့အနေနဲ့ စနစ်တကျ ဂရုစိုက်ပြီးတော့ လေ့လာရမှာ ဖြစ်ပါတယ်။ ဒါဆိုရင် Final Keyword ကို အသုံးပြုပြီးတော့ လက်တွေ program တစ်ပုဒ်ရေးကြည့်ကြရအောင်။ ဒါမှသာ Final Keyword ရဲ့ သဘောတရားကို နားလည်သဘောပေါက်သွားမှပါ။ အထူးသတိပြုရမယ့်အချက်ကတော့ class တစ်ခုတည်းမှာ final keyword နဲ့ Declarations လုပ်ခဲ့တဲ့ class ရဲ့ member function ( ) ကို အခြား class တစ်ခုမှဖော်၍ Inheritance Method ဖြင့် extends လုပ်ဖို့မြင်း၍ မရနိုင်ဘူးဆိုတဲ့ အချက်ကိုတော့ အထူးလုပ်ပေါ်သွားပါ။ class ကိုယ်၏ member ^ function () တဲ့

## **Listing 2.11 Using the final Methods in JAVA programming**

```
Methods in Java programming
1. public class testFinal {
2.     public static void main
(string args [ ]) {
3.     SuperClass s = new superClass
( );
4.     System .Out. print In ("Mes-
sage is" + S. getMessage ( ) );
5.
6.     SubClass t = new subclass (
);
7.     System .Out. print In ("Mes-
sage is " + t.getMessage ( ));//-
getMessage ( ) can't be overridden
8.   .}
9. }
```

```

10. class superclass {
11.     public final String getMessage() // Declare in final Method
12.     {return "SuperMessage";}
13. }
14. class SubClass extends Super Class {
15.     public String getMessage() // error: getMessage() can't
be overridden
16.     {return " Sub Message";}
17. }

```

### Analysis

၁။ ဒီ program ဟာ final Method ကို အသုံးပြုပြီးတော့ ဘယ်လိုအသုံးချလိုရတယ်။ ဘယ်လို protected လုပ်လို့ရတယ်ဆိုတာကို ရှင်းပြေားပါတယ်။ လိုင်းနံပါတ် (၁၀) က ရေးသားတဲ့ SuperClass ကို စတင်ပြီး အရင်လေ့လာကြည့်ရအောင်နော်။ လိုင်းနံပါတ် (၁၁) မှာ getMessage() ဆိုတဲ့ function ကို ရေးသားထားပါတယ်။ ဒီအခါ function ရဲ့ရှေ့မှာ final ဆိုတဲ့ keyword ကို Declare လုပ်ထားတာကို တွေ့ရမှာပါ။ ဒီ keyword ကြောင့် ဒီ function ဟာ Inheritance Method နဲ့ extends လုပ်လိုမရနိုင်တော့ပါဘူး။ အဲဒီ function ထဲမှာ ရေးသားတာကတော့ SuperMessage ဆိုတဲ့ ဘာသား String ကို main() function ထဲကို return ပြန်ပေးဖို့ပါပဲ။ လိုင်းနံပါတ် (၁၂) မှာ ရေးသားထားပါတယ်။ အဲဒီ class ရဲ့ close brace} ကိုအောက်တစ်ကြောင်းမှာ လာပိတ်ထားပါတယ်။ ဒီ class ထဲမှာ final keyword ကို Declare လုပ်ထားတာကြောင့် တစ်နည်းပြာရရင်တော့ final Methods ကို အသုံးပြု ရေးသားထားတာကြောင့် နောက် class တစ်ခုကနေ extends လုပ်လို မရနိုင်တော့ပါဘူး။ Note: Final Methods အကြောင်းကို ရှေ့မှာ သေသေချာချာရှင်းပြုခြော့ပါပြီ။ နားလည်သောပေါက်အောင် အထပ်ထပ် အခါခါ ဖတ်ပြီးတော့မှ ဒီ program ကို လေ့လာအချင်ပါတယ်။ programming လုပ်တယ်ဆိုတာဟာ Theorem ကို သောာပေါက် နားလည်မှသာလျှင် ကိုယ့်အတွက် အကျိုးရှိရှိအသုံးချရေးသားနိုင်မှာပါ။

၂။ လိုင်းနံပါတ် (၁၃) မှာ အပေါ်မှာရေ့ခဲ့တဲ့ SuperClass ကို မိုင်မြေး၍ extends လုပ်ပြီး SubClass ကို ဖန်တီးရေးသားပါတယ်။ ဒီတော့ လိုင်းနံပါတ် (၁၄) မှာ getMessage() ကို ရေးသားပါတယ်။ ဒီအခါမှာ function () - Name ဟာ တူဖော့အတွက် Overloading Method နဲ့ ဆင်တူဖော့ပါတယ်။ ထူးခြားတာက data type ဟာလည်း တူဖော်ပြန်တယ်နော်။ ဒါပေမဲ့ ဒီ function () ဟာ အသုံးချလိုမရနိုင်ဘူး။

Error ဖြစ်တယ်ဆိုတာ ကို comments ရေးသားထားပါတယ်။ ဒီ class ဟာ အသုံးချလို မရနိုင်ဘူးဆိုတာကို စာဖတ်သူအနေနဲ့ နားလည်သောပေါက်အချင်ပါတယ်။ ဒီ class ရဲ့ close brace } ကို လိုင်းနံပါတ် (၁၇) မှာ လာပိတ်ထားတာကို တွေ့ရမှာပါ။

၃။ ကဲခါဆိုရင်တော့ main() function ရေးသားပါဝင်တဲ့ primary class ကို လေ့လာကြရအောင်။ ဒါဆိုရင် လိုင်းနံပါတ် (၁) ကနေစပြီး လေ့လာကြရအောင်။ testfinal ဆိုတဲ့ primary class ကိုတွေ့ပြုနော်။ အဲဒီ class ရဲ့ လိုင်းနံပါတ် (၂) မှာ main() function ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၃) မှာတော့ SuperClass ကို object အဖြစ် ဖန်တီးပါတယ်။ ပြီးတော့ လိုင်းနံပါတ် (၄) မှာ main() function ထဲကို return ပြန်ပြီး ရောက်လာတဲ့ String ကို Output ထုတ်ပေးပါတယ်။ ကဲ ဒီအထိတော့ ဘာမျှ ပြဿနာမရှိသေးပါဘူး။ စာဖတ်သူလည်း အကုန်နားလည်း သောာပေါက်မှာပါ။ အဲဒီအောက်မှာ Line - space ချုန်ပြီးတော့ လိုင်းနံပါတ် (၅) မှာ SubClass ကို object အဖြစ်ဖန်တီးလိုက် ပြန်ပါတယ်။ ပြီးတော့ အဲဒီ Sub Class ထဲက getMessage ကို လုပ်စော်ပေးပါတယ်။ ဒေါ်ယူလို မရနိုင်တော့ ပါဘူး။ ဒါကြောင့် comments ရေးပြထားတာပါ။ ဒါဟာ Final Method ကြောင့် လုပ်စော်ပေးပါယူလို မရနိုင်တာပါ။ ဒါဆိုရင်တော့ Final Method ရဲ့အသုံးဝင်ပုံကို နားလည်သောပေါက်သွားလောက်ပါပြီ။ သေသေချာချာ လေ့လာထားပါ။

ဒီ program ကို လေ့လာကြည့်ပြီးတော့ Final Methods protected ပြုလုပ်နိုင်တဲ့ စွမ်းရည်အကြောင်းနဲ့ Declaration ပြုလုပ်နဲ့ ဒေါ်ယူအသုံးချတဲ့ type မှာ clearified ဖြစ်ပုံတို့ကို လေ့လာတွေ့ရှုရမှာပါ။ ဒါဆိုရင် static.Method, Final Method အကြောင်းတို့ကို နားလည်သောပေါက်သွားပြီဆိုရင်တော့ ဆက်လက်ပြီးတော့ Abstract Method အကြောင်းကို လေ့လာကြတာပေါ့။ ဒီတော့ Abstract Method အကြောင်းကို ဆက်လက်ပြီး လေ့လာကြပါစို့။

### Abstract Method

ရှေ့မှာရှင်းပြုခဲ့တဲ့ static Method, Final Method တွေ ကို နားလည်သောပေါက်ပြီဆိုရင် Abstract Method ဖန်တီးရေးသားပုံကို လေ့လာရမှာပေါ့။ Abstract Method ကိုဖန်တီးရေးသားတော့မယ်ဆိုရင် Abstract Keyword ကို အသုံးပြုပြီးတော့ Abstract class ကိုဖန်တီးရပါတယ်။ အဲဒီ Abstract class ထဲမှာ Abstract- function ကို ရေးဖွံ့ဖြိုးပါတယ်။ ထူးခြားတာကတော့ abstract function ကိုရေးဖွံ့ဖြိုးပါတယ်။ ဒါမူးခြား အပေါ်မှာ အသုံးချလိုမရနိုင်ဘူး။

stract function မှာ function - body ကို ရေးသားပေးစရာ မလိုပါဘူး။ ဒါဟာ abstract - function ကို design ချွဲတဲ့ ဖြစ်ပါတယ်။ ဒါအခါ ဘယ်လိုအကျိုးကျေးဇူးမျိုးတွေ ရရှိနိုင်မလဲ ဆိုတော့ abstract Method ကို အသုံးပြုခြင်းအားဖြင့် ရေးသား ခဲ့တဲ့ အဲဒီ abstract - function ကို အခြား sub - class တွေက တစ်ဆင့်မြို့ပြီးတော့ အသုံးချလိုရနိုင်သလို ဆက်နှံယ်ချက်တွေ နဲ့ ထပ်မံ မျှဝေပြီးတော့ အသုံးပြုနိုင်ပါတယ်။ Abstract Method နဲ့ ပတ်သက်တဲ့ အဓိပ္ပာယ်ဖွင့်ဆိုချက်ကို လေ့လာကြည့်ပါ။

\* Abstract Methods are designed by the abstract keyword and no body.

\* Abstract methods are used in abstract classes to impose a certain functionality on sub- classes of the abstract class.

ကဲဒါဆိုရင်တော့ အခုရှင်းပြခဲ့တဲ့အချက်တွေကို လေ့လာပြီးတော့ abstract Method နဲ့ ပတ်သက်တဲ့ Listing 2.12 ကို လေ့လာကြရအောင်။

**NOTE:** Abstract Method ကို ပို့ချုပ်အခါမှာ JAVA Language ကိုလေ့လာပြီး ယနှစ် programming နယ်ပယ်ထက် စတင်ဝင်ရောက်လာတဲ့ သူတွေကတော့

#### Listing 2.12 Using the Abstract Methods in JAVA

```

1. public class testABs {
2 public static void main (String
args [ ] ) {
3. Rectangle r = new Rectangle
(5, 4);
4. Triangle t = new Triangle
(4, 3);
5. System .Out. print In ("Rect-
angle area is :" + r. getArea (
) );
6. System . Out. print In (""
Triangle area is "+
t.getArea ( ) );
7. }
8. }
9.
10. abstract class Figure { // 
abstract class
11. double idim, jdim;
12. Figure (double a, double b)
13. {idim = a; jdim = b}
14. abstract double getArea ( )
;
15. }
16.
17. class Rectangle extends Fig-
ure {
```

```

18. Rectangle (double a, double
b)
19. {Super (a,b);} / / the class
figure constructor is called
20. double getArea ( ) {return
idim* jdim;}
21. }
22.
23. class Triangle extends Fig-
ure {
24. Triangle (double a, double
b)
25. {Super (a,b);} // the class
Figure constructor is called
26. double getArea ( ) {return
0.5* idim * jdim;}
27. }
```

#### Output

Rectangle area is: 20.0

Triangle area is: 6.0

#### Analysis

၁။ ဒါ program ဟာ အဆင့်မြင့် Abstract Method ကို Demonstrate လုပ်ပြီးရှင်းလင်းပြထားတာပါ။ ဒီတော့ အရင်ဆုံး လိုင်းနံပါတ် (၁၀) ကရေးသားထားတဲ့ Figure ဆိုတဲ့ class ကိုအရင်ပြီး လေ့လာကြရအောင်။ အဲဒီ Class မှာ abstract ဆိုတဲ့ keyword ကို Declare လုပ်ထားတာမှာ အဲဒီ class ဟာ abstract class ဖြစ်သွားပြီး။ abstract class ထဲမှာ double data type ကို အသုံးပြုပြီးတော့ idim နဲ့ jdim ဆိုတဲ့ variable နှစ်ခုကို Declare လုပ်ပါတယ်။ ပြီးတော့ abstract class ရဲ့ constructor function ဖြစ်တဲ့ Figure () function ကို လိုင်းနံပါတ် (၁၂) မှာ ရေးသားထား တာကို တွေ့ရမှာပါ။ အဲဒီ Figure () function ဟာ constructor function ဖြစ်ပါတယ်။ အဲဒီ function မှာ parameter value နှစ်ခုကို arguments အနေနဲ့ Declare လုပ်ပေးထားပါတယ်။ အဲဒီကြောင့်

Figure (double a, double b)

ဆိုတာကိုတွေ့ရတာပါ။ အဲဒီ parameter value နှစ်ခုကို function ရဲ့ body ထဲမှာ အပေါ်က class ရဲ့ member data တွေဖြစ်တဲ့ idim နဲ့ jdim ဆိုတဲ့ variable နှစ်ခုနဲ့ဖြစ်ပြီး ညီယူပါတယ်။ အဲဒီကြောင့် လိုင်းနံပါတ် (၁၃)မှာ - {idim=a,jdim=b;}

အနေနဲ့ရေးပြီး ညီယူထားပါတယ်။ ဒီလိုမျိုး parameter arguments နဲ့ variables ကို ညီယူတာမျိုးတို့ အရင်ရေးမှာ ပို့ချုပ်တဲ့ Listing တွေမှာ တွေ့ဖူးမှာပါ။ ကဲလိုင်းနံပါတ် (၁၄)

မှာ abstract keyword နဲ့ အသုံးပြုရေးသားထားတဲ့ getArea( ) function ကိုတွေ့ရမှာပါ။ အဲဒီ getArea( ) function ဟာ abstract keyword ကို အသုံးပြုထားတာ ကြောင့် အဲဒီ function / Method ဟာ abstract Method ပါပဲ။ အဲဒီ abstract Method ဟာ လုပ်ဆောင်ရမဲ့ လုပ်ဆောင်ချက် Statement & Expression တွေပါဝင်တဲ့ function - body ကို ရေးသားထား ခြင်းမရှိပါဘူး။

၂။ လိုင်းနံပါတ်(၁၇) မှာ အဲဒီ abstract class ကိုလုပ်ဖို့မြှင့်းပြီး တော့ Rectangle ဆိုတဲ့ class ကို ရေးသားဖွံ့ဖြိုးပါတယ်။ ဒီအခါ အဲဒီ class ရဲ့ constructor function ကို ရေးသားပါတယ်။ အဲဒီ constructor function ထဲမှာတော့ abstract class ရဲ့ constructor function ကို လုမ်းခေါ်ယူထားတာကို တွေ့ရမှာပါ။ ဒါကြောင့် လိုင်းနံပါတ် (၁၉) မှာလည်း comments အနေနဲ့ ရေးပြထားတာဖော့။ လိုင်းနံပါတ် (၂၀) မှာတော့ ပင်မ Figure class က abstract function ကို overloading Method နဲ့ ရေးသားပြီး body - function - code တွေကို ရေးသားပါတယ်။ abstract function အဖြစ် ရေးသားစဉ်က code ကို ရေးသားခဲ့ခြင်း မရှိပါဘူး။ ဒါကြောင့် Inheritance လုပ်ထားတဲ့ ဒီ class မှာ လာပြီးတော့ function- body ကို ရေးသားတာဖြစ်ပါတယ်။ ဒီအချက်ဟာ အရေးကြီးတာင့် သေသေချာချာ နားလည်းအောင် ဂရတုဂိုက် လေ့လာခေါ်ပါတယ်။ ဒီ extends လုပ်ထားတဲ့ class ရဲ့ close brace ကို လိုင်းနံပါတ် (၂၁) မှာ ရေးသားထားတာကို တွေ့ရမှာပါ။ အခုရေးပြတဲ့အထိ နားလည်းပြီနော်။

**NOTE:** ဒီ program မှာ Inheritance Method, abstract Method, Overloading Method, Parameter arguments & pass by value လုပ်ထားတဲ့ နည်းစနစ်တွေဖြစ်တာမျိုး အထပ်ထပ်အခါခါ နားလည်းအောင် လေ့လာခေါ်ပါတယ်။ ကဲ ဆက်လက်ပြီးတော့ လိုင်းနံပါတ် (၂၃) မှာ ရေးထားတဲ့ [extends လုပ်ထားတဲ့ သီးသန်း class တို့ခြုံဖြစ်တဲ့] Triangle class ကို ဆက်လက် လေ့လာကြည့်ကြရအောင်။

၃။ ဒီ လိုင်းနံပါတ် (၂၃) မှာရေးထားတဲ့ Triangle class ဟာ လည်း လိုင်းနံပါတ် (၂၀) မှာ ရေးသားထားတဲ့ Figure class ကို extends လုပ်ထားတာပါပဲ။ ဒီ Triangle class မှာလည်း constructor function ကို ရေးဖွံ့ဖြိုးပါတယ်။ အဲဒီ constructor function မှာလည်း parameter value နှစ်ခုကို arguments အနေနဲ့ Declare လုပ်ထားပါတယ်။ လိုင်းနံပါတ် (၂၅) မှာကတော့ အပေါ်က extends လုပ်ခဲ့တဲ့ Figure class က constructor function ကို လုမ်းပြီးခေါ်ယူလိုက်ပါတယ်။ ဒါဟာ အပေါ်က Rectangle class မှာလည်း တွေ့မြင်ခဲ့ရပြီ ဆိုတော့ Inheritance Methods ကို အသုံးပြုနေတယ်

ဆိုတာကို စာဖတ်သူအနေနဲ့ နားလည်းသောပေါက်မိမှာပါ။ comments လည်း ရေးပြထားတယ်နော်။ အဲဒီနောက် abstract Method ကို အသုံးပြုပြီး ရေးသားထားတဲ့ getArea( ) function အတွက် တွက်ချက်ရမဲ့ function - code တွေကို ရေးသားပါတယ်။ လိုင်းနံပါတ် (၂၆) မှာ inline statement အနေနဲ့ ရေးသားထားတဲ့ -

```
double getArea () {return 0.5 * idm * jdim ;}
```

ဆိုတာကို တွေ့ရမှာပါ။ ဒါဟာ မူလ getArea( ) function မှာ မူလက ဘာမျှ မရေးခဲ့ပေမဲ့ အခုခြုံ Triangle class မှာကတော့ တွက်ချက်ရမဲ့ calculus လုပ်ရမဲ့ formula ကို လာရောက် ရေးသား ထည့်သွင်းထားတာဟာ abstract Method နဲ့ ဖန်တီးရေးသားတာပါ။ ဒီနေရာမှာ Inheritance Method ရဲ့ parent to child Method ကိုလည်း အသုံးပြထားတယ်ဆိုတာကိုလည်း တွေ့ရမှာပါ။ ဒါဆို စာဖတ်သူအနေနဲ့ ရှင်းသွားပြီနော်။

၄။ လိုင်းနံပါတ် (၁) ကစြိုးတော့ သွားလေ့လာကြည့်ကြရအောင် နော်။ ဒီတော့ main( ) function ပါဝင်တဲ့ primary class ကို testAbs ဆိုတဲ့ class ကိုတွေ့ရပြီနော်။ အဲဒီ class ထဲမှာ main( ) function ကိုတွေ့ရပါပြီ။ အဲဒီ main( ) function ထဲက လိုင်းနံပါတ် (၃) မှာ -

```
Rectangler = new Rectangle(5,4);
```

ဆိုတဲ့ စာကြောင်းဟာ လိုင်းနံပါတ် (၁၇) ကနေ လိုင်းနံပါတ် (၂၁) အထိရေးသားထားတဲ့ Triangle class ကိုလမ်းခေါ်ယူလိုက်ပြန်ပါတယ်။ ဒီ class ဟာလည်း Figure class ကို extends လုပ်ထားတာပေါ်နော်။ ဒီလိုပဲ Inheritance Method ကို အသုံးပြုပြီးတော့ ထပ်ဆင့် နှိမ်ပြုးထားတာကိုလည်း သတိပြုထားပါ။

၅။ လိုင်းနံပါတ် (၅) မှာတော့ object အဖြစ် ဖန်တီးခြင်းခံထားရတဲ့ Rectangle class ရဲ့ getArea( ) function ကိုလမ်းပြီး ခေါ်ယူလိုက်ပါတယ်။ အဲဒါကြောင့် လိုင်းနံပါတ် (၅) မှာ print statement နဲ့ ခေါ်ယူလိုက်တာနော်။ System.out.print In ("Rectangle area is :" + r.getArea()); ဒီလိမ့်မျိုး print statement နဲ့ အဖြေထုတ်လိုက်တဲ့အခါ Out-put အနေနဲ့ -

Rectangle area is :20.0

လို့ Output ထွက်လာပါတယ်။ လိုင်းနံပါတ် (၆) ကတော့ Triangle class ရဲ့ calculate area ကို တွက်ချက်ရန် အောက်ပါအတိုင်း ပေါ်ယူလိုက်တာပါ။

```
System.out.print("Triangle area is :" + t.getArea());
```

ဒီပုံစံအတိုင်းလိုင်းနံပါတ် (၆) မှာ ပေါ်ယူလိုက်တဲ့အခါကျတော့ output ဟာ

Triangle area is : 6.0

ဆိုပြီးထွက်လာတာပါ။ ဒါဆိုရင် abstract class ကို ဖန်တီးပုံ abstract Method နဲ့ရေးသားပုံကို စာဖတ်သူ အနေနဲ့ တတ်မြောက်နားလည် သဘောပါက်သွားလောက် ပါပြီဖော်။ အဲဒီ abstract class ကိုပေါ်ယူအသုံးချထားတဲ့ main () function ရဲ့ close brace }ကို လိုင်းနံပါတ် (၇) မှာ လာပိတ်ထားတာပါ။ လိုင်းနံပါတ် (၈) မှာတော့ primary class ရဲ့ close brace }ကို လာပိတ်ထားတာပါ။ Class တွေ ဖန်တီးထားတာများတဲ့ အတွက် အဲဒီ brace } တွေကအစ သတိပြုပို့ချပေးနေတာပါ။ ဒါဆိုရင်တော့ abstract Method အကြောင်းကို တတ်မြောက်နားလည်ပြီး ကျမ်းဝင်သွားလောက် ပါပြီ။

ကဲ အခုံလို့ abstract Method အကြောင်းကို နားလည်ရှင်းလင်း တတ်မြောက်သွားပြီးဆိုရင် Method Declarations လုပ်တဲ့ type တွေကို ရှင်းလင်းတင်ပြထားတဲ့ ယေားဖြစ်တဲ့ Table 2.2 ကို လေ့လာပါ။ အဲဒီ Table 2.2 မှာ Method Declarations တွေရဲ့ သဘောတရား၊ Keyword တွေရဲ့ သဘောတရားနဲ့ ရှင်းလင်း ချက် visibility တွေကို မူလ Java Standard Edition ရဲ့ Guide Book တွေအတိုင်း ရေးသားဖော်ပြပေးလိုက်ပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

### Method Declarations Forms

Keyword	visibility
Static final	This method as a class method A final method cannot be overridden by sub-class.
Abstract	It has no implementation and must be a member of an abstract class.
Native	Methods implemented in a language other than JAVA are called Native Methods.
Synchronized	Concurrently running threads often invoke methods that operate on the same data.

Table 2.2 Type of Methods Declarations Forms in JAVA

ဒီအခါ Table 2.2 ကို လေ့လာတဲ့အခါမှာ Methods Declarations Form ဟာ ပုံစံ (၅) ပျီးရှိနေတာကို စာဖတ်သူဟာ တွေ ဖြင့်ရှုံးပါ။ ဒါ Table 2. မှာပါဝင်တဲ့ Methods Declarations Type တွေထဲက ကျွန်ုတ်သင်ကြားပို့ချာတဲ့ Static Methods, final Methods, Abstract Methods စုတဲ့ Methods (၃) ချုဟာ လက်ရှိ J2 SE version 1.4 မှာ အမိက ပြဋ္ဌာန်းတွက် ချက်ထားတာပါ။ Native Methods နဲ့ Synchronized Methods တို့ကိုတော့ သီးသန့် သင်ကြားပို့ချေပေးမှာပါ။ ဒါတွေဟာ J2SE Standard Version 1.4 မှာသာမက J2 EE Enterprise Edition မှာပါ အလွန်ထင်ရှားတဲ့ Methods Declarations form တွေပါပဲ။ ဒီပုံစံတွေကို နားလည်သဘောပါက်မှသာလျှင် programmer တစ်ယောက်အနေနဲ့ ရှေ့လျှောက်တက်လှမ်းရမှာဖြစ်တဲ့အတွက် သီးသန့် ဆက်လက်ပို့ချေပေးပါမယ်။

Table 2.2 မှာရေးသားပြထားတဲ့ Methods Declarations Forms တွေကို သေသေချာချာ နားလည်သဘောပါက်သွားရင် Class တွေတို့ object အဖြစ် ဖန်တီးပဲအခါမှာ အလွန် အသုံးဝင်တဲ့ Access Specifiers Methods တွေကို လေ့လာကြည့်ရအောင်။

### Access Specifiers Methods

စာဖတ်သူအနေနဲ့ သာမန်အခြေအနေပျီးမှာ ဒါ Access Specifiers Methods တွေအကြောင်းကို မသိရင်လ အရေးမကြီးပါဘူး။ ဒါပေမဲ့ အခုံလို့ object -concepts တွေနဲ့ program တွေကိုရေးသားတဲ့အခါမှာတော့ ဒါ Access Specifiers Methods တွေကို မသိလို့မရတော့ပါဘူး။ ဘာကြောင့်လ ? ဒါ Object -concepts တွေဆုံးရမှာ -

- Inheritance , Derived Methods (OR) Parent to Child Methods
- Overloading Methods,
- Constructor Methods, Destructor Methods

ဒါ Methods တွေနဲ့ ရေးသားပြီးဆိုတာနဲ့ class တစ်ခုနဲ့တစ်ခုဖြစ်စေ၊ object အချင်းချင်းဖြစ်စေ၊ package တွေမှဖြစ်စေ၊ မို့ပြစ်းရတာ မဟုတ်လား၊ ဒီအခါ class တစ်ခုနဲ့တစ်ခုဖြစ်စေ၊ object အချင်းချင်းဖြစ်စေ၊ မို့ပြစ်းအသုံးချခွင့်ရှိဖို့ရှိရမှာ အခုံလို့ပို့ချေနေတဲ့ Access Specifiers Methods တွေကို နားလည်တတ်မြောက်ထားရမှာပါ။

**NOTE:** ဒါ Access Specifiers Method တွေနဲ့ပတ်သက်ပြီး keyword တွေနဲ့ သက်ဆိုင်တာကို အကြမ်းဖျင်းအားဖြင့်တော့ 2.1 Introduction to classes ဆိုပြီး ရှင်းဖြေချို့ပါပြီ။ ဒါတော့ အဲဒီမှာ အကြမ်းဖျင်းအားဖြင့် public keyword အကြောင်းကို ရှင်းပြခဲ့တာပါ။

## ကွန်ပူးတာ သင်ခန်းစာ

ဒီတော့ Access Specifiers Methods တွေအကြောင်းကို Table 2.3 အနေဖြင့် ရေးသားပို့ချုပ်မယ်။ ဒီတော့ Access Specifiers Method နဲ့ ပတ်သက်တဲ့ Table 2.3 ကို အရင်လေ့ လာလိုက်ပါဉိုး။ ဆက်နှစ်ချက်တွေကိုတော့ သီးသန့်ရှင်းပြပါမယ်။

### **Access Specifiers Methods**

Keyword	Visibility
Private	Access to a private variable or method is only allowed within the code fragments of a class.
Protected	Access to protected variable or method is only allowed within the code fragments of a class and its sub class.
(friendly)	Access to a friendly variable or method is only allowed within the code fragments of a class and any other class in the same package.
Public	Access to a public variable or method is unrestricted. It may be accessed from the code fragments of any class.

**Table 2.3 Access Specifiers Methods in JAVA programming Language**

စာဖတ်သူအနေဖြင့် Table 2.3 က ယေားကို လေ့လာကြည့်ပါ။ ဒီအခါမှာ ကျွန်တော်တို့ ရင်းနှီးကျမ်းဝင်ပြီးသားဖြစ်တဲ့ Public keyword, Private keyword စာတော့အပြင် Protected keyword ကိုပါတွေရမှတ်ပါ။ ဒါဟာ ကျွန်တော်တို့ အနေဖြင့် တတ်မြောက်ကျမ်းဝင်ပြီးသား keyword တွေပါပဲ။ public keyword နဲ့ ပတ်သက်တဲ့ ဆက်နှစ်ချက်ကို 2.1 Introduction to Class ဆိုတဲ့ အခန်းကဏ္ဍမှာ သေသေချာချာရှင်းလင်းပြီးပါ၍။ ဒီအခါ private keyword နဲ့ protected keyword တွေအကြောင်းကို

တော့ ကျွန်တော်အနေဖြင့် ဆက်လက်သင်ကြားပို့ချေပေးပါမယ်။ ဒါဟာ Access Specifiers Methods နဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကို သဘောတရားနဲ့ နားလည်တတ်မြောက်အောင် သေသေချာချာ ရှင်းလင်းပြောပါမယ်။

ဒီနေရာမှာ ရှင်းလင်းပြတဲ့နေရာမှာ public keyword, private keyword, protected keyword စာတွဲရဲ့ အကြောင်းအရာတွေဟာ Class တွေရဲ့ အရေးအကြီးတဲ့ အခန်းကဏ္ဍပဲဆိုတာကို မမေပါနဲ့။ class နဲ့ object တွေရဲ့ ဆက်နှစ်ချက်ဖြစ်တဲ့ အတွက်ပါပဲ။ ဒီတော့ အရင်ဆုံး Table 2.3 ကို လေ့လာအပြီးမှာ ဆက်လက်ပြီးတော့ Table 2.4 ကို ဆက်လက်လေ့လာကြည့်ပါ။ Table 2.3 နဲ့ Table 2.4 ဟာ ဆက်စပ်နဲ့ အတွက်ဖြစ်ပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။ Table 2.4 ကို လေ့လာတဲ့အခါမှာ Table 2.4 ရဲ့ ခေါင်းစဉ်အခန်းကဏ္ဍတွေကိုပါ သေသေချာချာ ဖတ်ရှုလေ့လာပါ။

1. Access Specifies	2. Accessible from Base <Parent> class	3. Accessible from derived <child> class	4. Accessible from objects outside class
public	Yes	Yes	Yes
private	Yes	No	No
protected	Yes	Yes	No

**Table 2.4 Connected to Access Specifiers Methods in JAVA Language**

ဒီ Table 2.4 ဟာ keyword တစ်ခုချင်းစီရဲ့ သဘောတရားတွေကို အသေးစိတ်ရှင်းလင်းပြထားတဲ့ ယေားပါပဲ။ ဒါကြောင့် Table 2.4 က ယေားကွက်မှာ အကွက်နံပါတ်လေးတွေ တပ်ပေးထားပါတယ်။ အကွက်နံပါတ် (၁) က Access Specifiers ဆိုတာဟာ keyword တွေပါ။ ဒီတော့ public keyword ကစပြီး လေ့လာကြည့်ကြရအောင်။ ကဲ Table 2.4. ၁ မှာ ရှင်းလင်းပြထားပါတယ်။

public keyword ကိုအသုံးပြုခြင်းအားဖြင့် data type variables ဖြစ်စေ၊ function () / Method ဖြစ်စေ၊ Class &

1. Access Specifies	2. Accessible From Base <parent> class	3. Accessible From Derived <child> class	4. Accessible From Objects outside class
Public	Yes	Yes	Yes
ဒီအကွက်ဟာ public keyword ဖြစ်ခြင်းကို ရေးပြထားတာပါ။ ဒီ public keyword ကို အသုံးပြုထားခြင်းအားဖြင့် ဝင်မ(မိခင်) class ထဲမှ စိတ်တိုင်းကျ ၁၅၂ အသုံးပြုခြင့်ရှိတယ်။ အချက်ကို ရှင်းလင်းပြထားတာပါ။	ဒီအကွက်ဟာ public keyword ကို အသုံးပြုခြင်းအားဖြင့် အခြား class <sub-class> ကနေ ၏ ယူအသုံးချုပ်မှုများ ပေးပါတယ်ဆိုတဲ့ အချက်ကို ရှင်းလင်းထားတာပါ။	ဒီအကွက်ကတော့ အခြား object တစ်ခုကနေ ဖန်တီးပြီးတော့ လုမ်းချွေ ယူမယ်ဆိုရင်လဲ Public keyword ရဲ့ သဘောတရားကြောင့် အသုံးချုပ်ပေးပါတယ်။ ဒါဟာ public keyword ရဲ့ ဇမ်းဆောင်ရည်ပါပဲ။	

**Table 2.4.1 Complete Illustrate of public keyword (Access Specifies)**

ကုန်ပူတာ သင်ခန်းစာ

object တစ်ခုလုံးကိုဖြစ်စေ ပင်မမိခင် class ထဲမှ လိုအပ်သလို ၌။ အသုံးချိန်ယုံသာမကပဲ အခြား Derived (child) class များကပါ နှိမ်မြေး၏ ရရှိပါတယ်။ ဒါ၊ အပြင် အခြား class & object များ အတွင်းကပါ ဆွဲ၌ အသုံးချိန်ပါတယ်။ အဲဒီ စွမ်းဆောင်ရည်

**Table 2.4.2 Complete Illustrate of the private key-word (Access Specifiers)**

1. Access Species	2. Accessible From Base <parent> class
Private	Yes
ဒီအတွက်ဟာ private keyword ကို Declare လုပ်ဖို့အနေအထားကို ဖြတဲ့ အတွက်ပါပဲ။ ဒါပိုရင် အခု ဒီ private keyword ကို အသုံးပြုခြင်းအား ဖြင့် ရလာနိုင်မဲ့ ဆက်နှစ်ယ်ချက်တွေကို လေ့လာကြည့်ကြရအောင်လား။	Private keyword ကို Declare လုပ်ခြင်းအားဖြင့် မိမိပင်မ class လုပ်ထားပဲ။ ဆက်နှစ်ယ်ချက်အားလုံး ရရှိပါမယ်။ ဒါဟာ private keyword ကို declare လုပ်ပိုက်တာနဲ့ အလိုအလောက် ကန်သတ်ပေးလိုက်တဲ့ အချက်ပါပဲ။
3. Accessible From Derived <child> class	4. Accessible From Objects outside class
No	No
ဒီယေားဘက်ကတော့ Private keyword ကို အသုံးပြုခြင်းအားဖြင့် ရေးသာထားတဲ့ function/method ကို ဖြစ်စေ၊ class object တစ်ခုလုံးကို ဖြစ်စေ၊ အခြား derived <child> class မှတ်၍ ခေါ်ယူအသုံးချလို့ မရရှိပါဘူးဆိုတာဘို့ ရှင်းပြထားတာပါ။	Private keyword ကို အသုံးခြင်းအားဖြင့် အခြား object & class မှ လုပ်ပေါ်ယူ အသုံးချ၍ မရရှိပါ။ အဲဒီ အချက်ကို ကန်သတ်ချက်တစ်ခု အနေနဲ့ ရှင်းပြထားတာပါပဲ။

ຕົວ ລັບື້ນິດື້ນີ້ສີຕາບາ public keyword ສີຕະ Access Specifiers ກິດເວັ່ນໆປຸງມາວາ ລັບື້ນິດຕະ ດູ້ຈະເຫັນລົບປີບ॥

ဒီအချက်ကို သေသေချာချာ နားလည် သဘောက်ပေါက်အောင် Table 2.4 နဲ့ Table 2.4 .1 တို့ကိုသေသေချာချာ လေ့လာပါ။ ရှင်းလင်းချက်တွေကိုလည်း အထပ်ထပ်အခါခါ ဖတ်ရှုလေ့လာပါ။ public keyword အကြောင်းကို နားလည် သဘောပေါက်သွားပြီဆိုရင် private keyword အကြောင်းကို ဆက်လက် လေ့လာကြတာပေါ့။

ကဲ Table 2.4.1 ဆိတာဟာ Private keyword ကို ရှင်းလင်းပြထားပါတယ်။ အဲဒီ Private keyword ဟာ Declare လုပ် အသုံးချဖိုက်ဖြစ်ဆိုရင် မိခင် ပင်မ (parent) Base Class ထဲမှာ စိတ်တိုင်းကျခေါ်ယူအသုံးချနိုင်ပါတယ်။ ဒါပေမဲ့ အဲဒီ မိခင် ပင်မ (parent) class ထဲကနေတစ်ဆင့် အခြား Class မှာ အသုံးချဖို့အတွက်ကိုတော့ ခွင့်မပြပါဘူး။ နားလည်အောင်ပြောရရင်တော့ အခြား Derived (child) Class ကနေပြီးတော့ လုမ်းခေါ်ယူအသုံးချမယ်ဆိုရင်တော့ အဲဒီ private keyword ကြောင့် အသုံးချဖို့မှုနိုင်ပါဘူး။ အဲဒီလို့ ဆွဲခေါ်အသုံးချဖို့မှု မရနိုင်အောင် Private keyword ကို Declare လုပ်တယ်ဆိုရင်လည်း မမှားပါဘူး။

အခုရင်းပြတဲ့အချက်ကို သေသေချာချာ နားလည်သော  
ပေါက်သွားအောင် Table 2.4 Connected to Access Speci-  
fiers Methods in JAVA Language ဆိုတဲ့ ထေားနဲ့ Table  
2.4.1 Complete Illustrate of the private keyword (Ac-  
cess Specifiers) ဆိုတဲ့ ထေားကို နှိမ်းယဉ်ပြီးတော့ သေသေချာ  
ချာလေ့လာပါ။ ဒီအချက်တွေဟာ အဂုန်အရေးကြီးတဲ့ အချက်  
အလက်တွေပါပဲ။ အခုရင်း public keyword အကြောင်းနဲ့  
private keyword အကြောင်းကိုတော့ နားလည်သောပေါက်  
သွားလောက်ပါပြီ။ ဒါပါရင်တော့ ဆက်လက်ပြီးတော့ protected  
keyword အကြောင်းကို ဆက်လက်လေ့လာကြရအောင်လား။

**Table 2.4.3 Complete Illustrate of the protected keyword (Access Specifiers)**

1. Access Specifiers	2. Accessible From Base <parent> class	3. Accessible From Derived <child> class	4. Accessible From Objects outside class
Protected	Yes	Yes	No
ဒီယောက်တော့ protected keyword ကို declare လုပ် အသုံးဖြင့် အာမြင့် ရလာနိုင် ဆက်စွမ်းယူ၍၊ အား သာချက်တွေကို ရင်းပြုနိုင်ရန် Declare လုပ်ပြထားခြင်းပဲ ဖြစ်ပါတယ်။ ဒီတော့ ယောက်တော်ကိုချင်းစီ သေသေ ချာချာ လေ့လာကြည့်ကြရအင်။	ဒီအကွက်ကတော့ protected keyword ကို အသုံးပြုရင် သူ၏ ဆက်စွမ်း class ဖြစ်တဲ့ တစ် ဆင့်ခဲ့ derived <child> class ထဲမှာ အသုံးချလဲ ရရှိပါတယ်။ ဒီအချက် ဟာ private keyword ထောက်သာတဲ့ အားသာချက်ပါပဲ။ ဒီနေရာမှာ ဘယ် keyword ကိုပဲ အသုံးသုတေသနဖြစ်စေ၊ မိခင် class ထဲမှာတော့ အသုံးချိန်ပါတယ်။ အဲဒီ အချက်ဟာ public/private/protected keyword အားလုံးမှာ အတူတူပါပဲ။	Protected keyword ကို အသုံးပြုရင် သူ၏ ဆက်စွမ်း class ဖြစ်တဲ့ တစ် ဆင့်ခဲ့ derived <child> class ထဲမှာ အသုံးချလဲ ရရှိပါတယ်။ ဒီအချက် ဟာ private keyword ထောက်သာတဲ့ အားသာချက်ပါပဲ။ ဒီနေရာမှာတော့ public keyword ပဲ သဘောတရားနဲ့ ဆင်တူနေပါတယ်။ သတိပြုလေ့လာပါ။	Protected keyword ကို အသုံးပြုခဲ့ ပြီးနိုတာနဲ့ အဲဒီ protected variable ဖြစ်စေ၊ member function ( )/Methods ကိုဖြစ်စေ၊ class & object ဖြစ်စေ၊ အခြား objects outside class နေရာမှ ဆွဲပေါ်ယူ အသုံးချလဲ မရရှိနိုင်တော့ပါဘူး။

### Protected keyword

Protected keyword နဲ့ ပတ်သက်တဲ့ အသေးစိတ်ရှင်းလင်းချက်ကို Table 2.4.3 မှာ ရှေးသား ရှင်းလင်းပြေားထားပါတယ်။ အဲဒီ Table 2.4.3 ကို အရင်လေ့လာလိုက်ပါဉိုး။

**Note:** Protected keyword အကြောင်းကို လေ့လာကြည့်တဲ့ အခါမှာ protected keyword ဟာ public keyword မှာတုန်းကလိုမျိုး Base (child) class မှာပါ အသုံးချလို့ရနိုင်ပါတယ်။ ဒီအချက်ဟာ public keyword နဲ့ protected keyword မှာ အတူတူပါပဲ။ private keyword ကတော့ မိခင် (ပင်မ) class ကနေဂွဲဗြို့ တစ်ဆင့်ခဲ့ class ဖြစ်စေ အခြား class & object တွေမှာဖြစ်စေ အသုံးချလို့ မရနိုင်ပါဘူး။

Table 2.4.3 ကို သေသေချာချာ ဖတ်ရှုလေ့လာကြည့်ရင် protected keyword အကြောင်းကို နားလည်သောပေါက်သွားမှာပါ။ ဒီအခါမှာ ကျွန်ုင်တော့ရှေ့မှာ သင်ကြားပို့ချုခဲ့တဲ့ public keyword , private keyword, protected keyword အကြောင်းတွေကို စာဖတ်သွေးသွေးသွေးသွေးအခါမှာ ကိုယ့်အနေနဲ့ program တွေရေးသားပဲ့အခါမှာ ဘယ် keyword (Access Specifiers) အနေနဲ့ သင့်တော်မယ်ဆိုတာကို စာဖတ်သွေးအနေနဲ့ သင့်တော်သလို နားလည်ဆုံးဖြတ်တတ်စေရန် ရည်ရွယ်ပါတယ်။

### အဲဒီအတွက်ကြောင့်

- Table 2.3 Access Specifiers Methods in JAVA programming language
- Table 2.4 .Complete to Access Specifiers Methods in JAVA language
- Table 2.4.1 Complete Illustrate of public keyword

### (Access Specifiers)

- Table 2.4.2 Complete Illustrate of private keyword (Access Specifiers)
- Table 2.4.3 Complete Illustrate of protected keyword (Access Specifiers)

အစရိတ် Table (5) ခုကို ထင်ပါတယ်လဲ ဖတ်ရှုပြီးတော့ လေ့လာထားစေချင်ပါတယ်။ JAVA Language မှာတင်မကပါဘူး။ အခုရှင်းပြုခဲ့တဲ့ Access Specifiers တွေဟာ C & C++ နယ်ပယ်က ကျောင်းသားတွေအတွက်ပါ အလွန်အရေးပါတဲ့ အခန်းကဏ္ဍကာန် တိဝင်ပါတယ်။ JAVA Developer Guide ကို လေ့လာသင်ယူနေတဲ့ သူအဖို့ကတော့ JAVA Language မှာသာမက အခြားသာ Programming Language တွေနဲ့ပါ Compare လုပ် နားလည်သောပေါက်သွားပါလိမ့်မယ်။ JAVA နဲ့ ပတ်သက်တဲ့ အချက်အလက်မှန်သူ့ကို အတတ်နိုင်ဆုံး၊ အပြည့်စုံဆုံးဖြစ်အောင် ရှင်းလင်းပို့ချေပေးထားပါတယ်။ JAVA Developer Guide ကို ပို့ချုပဲတဲ့အခါမှာ ပို့ချုပဲတဲ့သူ နေရာကသာမက သင်ယူတဲ့သူနေရာကနေ ရှုထောင့်အာက်ဘက်မှ တွေးခေါ်ဆင်ခြင်သွားသပ်ပြထားတဲ့အတွက် တစ်ခါတစ်ရုံ ကျယ်ပြန်သွားတယ်လို့ ထင်ရှာဖူးတွေ ရှိနိုင်ပေမဲ့ လို့အပ်မယ်ထင်တဲ့ အတွက် အပြည့်စုံဆုံး တတ်နိုင်သလောက် ရေးသားပို့ချေပေးတာပါ။ စာဖတ်ပျင်းမယ်။ Self - Study လုပ်ရာမှာ Sample program တွေ Listing တွေကို ကူးယူနိုက်သွင်းပြီး Run & Test လုပ်ရနဲ့တော့ တူးခွဲနိုင်တက်မြောက်တဲ့ပါရင် Programmer / Developer ကောင်းတစ်ယောက် ဖြစ်မလာနိုင်ပါဘူး။ ဒါကြောင့် သေသေချာချာ ဖတ်ရှုလေ့လာပါ။

ကျော်သွေးသွေးသွေးသွေး (ကထာ)

## ကွန်ပူးတာကာကွန်ယန်လုပ်ကြောန်းကို ၂၀၀၅ ခုနှစ် နိုင်ဘာလမှတ်၌ အောက်ပါအတိုင်း ပြောင်းလဲသတ်မှတ်ထားပါသည်။

တစ်အုပ်ချင်း ဝယ်ယူမိုက်

ဥရောပ

ကိုယ်တိုင်လာထုတ်ပျော်

ဥရောပ

တတိက်မှုပို့လျှင်

တစ်နှစ်စာ

ဥရောပ

ကွန်ပူးတာရာန်ယန်လုပ်ကြောန်းကို ၂၀၀၇ (ခ)၊ ပထမထပ်၊ ၃၃ လမ်း(အထက်)၊ ကျောက်တံတားမြို့နယ်။  
ဖုန်း - ၉၈၂၂၄၈



### 2.3.1 Passing Arguments in JAVA

အရှေ့မှာသင်ကြားပို့ချခဲ့တဲ့ 2.3 Access Methods and Passing Arguments in JAVA ဆိုတဲ့ အခန်းကဏ္ဍမှာကျယ်ပြန့်လာတဲ့အတွက် သီးသန် sub-title အနေနဲ့ခြဲ့ပြီး ရေးသားပို့ချရတာပါ။ ဒီနေရာမှာ Passing Arguments ဆိုတဲ့ အခန်းကဏ္ဍကို ရှုံးသင်ခန်းစာတွေမှာ Introduceလုပ်ခြဲ့ပြီးသားပါ။ arguments ဆိုတာဟာဘာလဲ? Pass by Valueလုပ်တယ်ဆိုတာဟာဘာလဲ? ဆိုတာကို အရှေ့မှာ သင်ကြားပို့ချပေးခဲ့ပါတယ်။ ဒီအခါမှာ အချို့သောလေးလာသူ programmer တွေအနဲ့ အခက်အခဲအနည်းငယ်တွေတဲ့အခါမှာ မေးဖြန်းချက်လေးတွေကို သီးသန်မေးမြန်းလာကြပါတယ်။ Projectတွေ ဖန်တီးတဲ့အခါမှာလည်း တွေလာရတဲ့ အခက်အခဲလေးတွေကိုလည်း အပါအဝင်ပေါ့။

ကျွန်တော့အနဲ့နဲ့ အခုလက်ရှိပို့ချနေတဲ့ chapter [2] အပြီး chapter [3] မတိုင်ခင်မှာ သင်ကြားပို့ချပြီးခဲ့တဲ့ chapter [1] & chapter[2] ကို Summary အနဲ့နဲ့ ဧည့်နဲ့ကြမ်းပြီးတဲ့အခါမှာ Exercises အနဲ့နဲ့ လေ့ကျင့်စရာ program လေးတွေကိုပို့ချပေးပြီး QWZ Answers အနဲ့နဲ့လည်း ဖြေဆိုပေးပါမယ်။ ကျွန်တော့ဘက်က ပြည့်စုံအောင်အတတ်နိုင်ဆုံး ကြိုးစားပေးထားပါတယ်။ အခုလက်ရှိသင်ကြားပို့ချပေးနေတဲ့ program လေးတွေရဲ့ Theory သဘောတရားတွေကို အတတ်နိုင်ဆုံး မှတ်သားလေ့လာထားပါ။

#### Passing arguments to methods

ဒီ Method နဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကို အရှေ့မှာ သင်ကြားပို့ချတာကြောင့် သီးသန်ထပ်မံမရှင်းပြတော့ပါဘူး။ အောက်မှာရေးပြထားတဲ့ Pass by Value ဆိုတဲ့ အကြောင်းအရာ လေးကိုပဲ လေ့လာလိုက်ပါတိုး။ ပြီးရင်တော့ listing 2.13 တို့ဆက်လက်လေ့လာကြည့်ပါ။

#### Pass by Value

In Java methods, Arguments are passed by value. The Arguments is of primitive type, pass by value means that the method cannot change its value.

```
Listing 2.13 used by pass by value Method in Java
1. class PassValue {
2.     void getPass (int a1, int b1,
3.                 int c1) {
4.         a1+ = 10;
5.         b1+ = 10;
6.         c1+ = 10;
7.         System.out.println ("First = "
+ a1+ "Second = " + b1+
"Third = " + c1);
8.     }
9.     public static void main
10.        (String args [ ] ) {
11.        int a = -1, b= -1, c = - 1;
12.        PassValue P = new P ( );
13.        P.getPass (a,b,c);
14.        System.out.println ("First = "
+ a+ "Second = "+ b+
"Third =" +c);
15.    }
16. }
```

#### Output

```
First=9 Second=9 Third=9
First=-1 Second=-1 Thrid=-1
```

#### Analysis

၁။ ဒီ program ကိုတော့ အထွေအထူး ရှင်းမပြတော့ပါဘူး။ ဘာကြောင့်လဲဆိုတော့ pass by value method နဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကို ရှေ့မှာသင်ကြားပို့ချပြီးတာ ဖြစ်လိုပါပဲ။ ကဲ အကြမ်းဖျင်းအားဖြင့် လေ့လာကြည့်ထုတိရိုရင် object အဖြစ် ဖန်တီးမယ့် PassValue ဆိုတဲ့

- class ကို ဖန်တီးပြီး object function အဖြစ် getPass ဆိုတဲ့ function c ယော်လေးကို ရေးသားပါတယ်။ အဲဒီ getPass ဆိုတဲ့ function c ယော်လေးရဲ့ Parameter - arguments နေရာမှာ int a1, int b1, int c1 ဆိုပြီးတော့ ရေးသားထား ပါတယ်။ အဲဒီနောက် function-body ထဲမှာတော့ a1,b1,c1 ရဲ့တန်ဖိုးကို 10 လို့ အသီးသီး သတ်မှတ်ပေးထားပြီးတော့ main () function ထဲကနေပြီး တော့ ထည့်သွင်း pass လုပ်လိုက်မယ့် တန်ဖိုးတွေနဲ့ Self-assigned လုပ်ပေးထားပါတယ်။ အဲဒီအထိ သေသေချာချာ ဖတ်ရှုလေ့လာ ကြည့်ပါ။ နားလည်သွားမှာပါ။ ပြီးတော့ သတ်မှတ်ပေးထားတဲ့ a1,b1,c1 တို့ရဲ့ တန်ဖိုးအသီးသီးကို output အဖြစ်ထုတ်ယူ လိုက်ပါတယ်။ က ဒါတော့ သီးခြား object အဖြစ် ဖန်တီးပြီးမှ class ကို object အဖြစ် ဖန်တီးခေါ်ပူးယူယံနေရာပါပဲ။ ဒီနေရာမှာ parameter - value တွေ သတ်မှတ်ပေးထားပြီးမှ အဲဒီ value တွေကတစ်ဆင့် pass လုပ်မှာပါ။
- J) main () function ထဲမှာတော့ int a,b,c ဆိုပြီး Declare လုပ်ပါတယ်။ ဒီအထဲမှာ ရေးသားအသုံးပြထားတာဟာ သီးသန့် variable တွေပါ။ main () function နဲ့သာ သက်ဆိုင်ပါတယ်။ က ဒါ main () function ထဲက တန်ဖိုး တွေဖြစ်တဲ့ int a,b,c ရဲ့ တန်ဖိုးတွေဟာ -1 တွေအဖြစ် သတ်မှတ်ပေးထားပါတယ်။ အဲဒီနောက် passvalue ဆိုတဲ့ class ကို object အဖြစ်လှမ်းပြီး ခေါ်ယူလိုက်ပါတယ်။ အဲဒီလို object အဖြစ် ဖန်တီးပြီးတာနဲ့ getPass () ဆိုတဲ့ ---- ကို ခေါ်ယူလိုက်ပါတယ်။ ဒီ getPass () function ရဲ့ parameter - value နေရာကို main () function ထဲမှာ Declare လုပ်ခဲ့တဲ့ a,b,c ရဲ့ တန်ဖိုးတွေကို pass လုပ်လိုက်ပါတယ်။ ဒါကြောင့် output ထွက်လေတဲ့ အခါမှာ - First = 9 Second = 9 Third = 9 လို့ output ထွက်လာတာပေါ့။ လိုင်းနံပါတ် (၁၂) မှာတော့ မူလ main () function ထဲက Declare လုပ်ခဲ့တဲ့ တန်ဖိုးတွေကိုပဲ Output ထုတ်တာမို့ Output အနေနဲ့ First = -1 Second = -1 Third = -1 ဆိုပြီးတော့မှ output ထွက်လာတာပါ။ အဲဒီအကြောင်းအရာ တွေကို စနစ်တကျလေ့လာကြည့်ပါ။ အသေးစိတ် line-by-line ရှင်းမပြတာကတော့ အရေးမှာ ဒီစွဲ program ထံလေးတွေ ကို ရေးသားပို့ချုပြီးဖြစ်လိုပါပဲ။ ဒါကြောင့် ဒီ program ထံလေး၊ program -flow ကို သေသေချာချာ လေ့လာထားပါ။ က ဒီအခါမှာ Pass by Value Method ကို လေ့လာပြီးရင် ဆက်လက်ပြီး Passing array arguments to method ကို ဆက်လက်လေ့လာကြရအောင်။ က ဒီ passing array arguments to method နဲ့ ပတ်သက်တာတွေကို အောက်မှာ ရေးသားပို့ချုပ်ထားပါတယ်။
- ### Passing array arguments to method
- \* The method is never able to modify the actual argument.
  - \* Arrays are passed to method by passing a copy of the reference to it.
  - \* The array elements can be modified by using the copy of the reference.
  - \* Primitive data types are passed to methods by value.
- ဒီအချက်တွေဟာ Java Language နဲ့ ပတ်သက်ပြီးလေ့လာ နေသူတွေအတွက် fully depended ဖြစ်နိုင်လိမ့်မယ်လို့ ယုံကြည့်တဲ့အတွက် ထည့်သွင်းပို့ချုပ်ထားတာပါ။ Java Language ဟာ OOP Language တွေထဲမှာ အထင်ရှုးဆုံးနဲ့ Tools အပြည့်စုံသုံးသော Language တစ်ခုဖြစ်တယ်ဆိုတာ မမေ့ပါနဲ့။ ဒီတော့ ကျွန်ုတ်တို့အနေနဲ့ အရင်ကျင်လည်ခဲ့တဲ့ Language တွေမှာ ခက်ခက်ခဲ့ ဖန်တီးယူခဲ့ရတဲ့ type တွေကို Java ရဲ့ နယ်ပယ်မှာ လွယ်လွယ်ကူကူလိုအပ်သလို ဖန်တီးရယူနိုင်ဖြောက်ပါတယ်။
- က ဒီတော့ ကျွန်ုတ်တို့အနေနဲ့ Java Language ရဲ့ ကျယ်ဝန်းလှတဲ့နယ်ပယ်ထဲမှာ လေ့လာရင်း Access Methods and passing Arguments တွေအကြောင်း Methods Declarations တွေ အကြောင်းတွေကို examples အသီးသီးနဲ့ Sample code line တွေ၊ Table တွေနဲ့ ပြည့်စုံအောင်လေ့လာ ခဲ့ကြပြီးပါပြီ။
- ဒီအခါမှာ C & C ++ နယ်ပယ်ကနေ ပြောင်းလဲလေ့လာ သူတွေအတွက်ကတော့ အခက်အခဲမရှိလေပေမယ့် သာမန် Application user ဘဝကနေပြီးတော့ Java Language ကို လေ့လာသူတွေအတွက်ကတော့ အချို့နေရာမှာ complex ဖြစ် ကောင်းဖြစ်နေပါလိမ့်မယ်။ ဒီတော့ ဒီအခန်းကဏ္ဍမှာ variable Declarations တွေအကြောင်းကို စနစ်တကျ လေ့လာကြရအောင်။ variable-Name ဆိုတာဟာ အမှန်ကတော့ program ရဲ့ field ထဲမှာ အသုံး ပြုမယ့် member - Name တွေပါပဲ။ တစ်ခါတစ်ရုံမှာ variable - Name တွေကို Declare လုပ်တဲ့အခါမှာ invalid variable - Name တွေကို ဖြစ်သွားတတ်ပါတယ်။ အဲဒီဟာ ဘာ ကြောင့်လိုတဲ့အချက်ကို လေ့လာရမယ်။ ပြီးတော့ Java ရဲ့ identifier keyword တွေကိုလည်း သီးခြား Table အနေနဲ့ ဖတ်ရှု လေ့လာကြရအောင်လား။ ဒါတွေဟာ Java Language ရဲ့ နယ်ပယ်က မသိမဖြစ်တဲ့ အချက်အလက်တွေပါပဲ။ ဒါမှာသာ

လိုအပ်ရင် J2EE (Java 2 Enterprise Edition) ကို တစ်ဆင့် လှမ်းတက်ပြီး Web-Based development အပိုင်းတွေကိုပါ အသုံးချလုပ်ကိုင်တဲ့အခါမှာ အခြေခံကျကျ အသုံးဝင်တဲ့ အချက် အလက်တွေဖြစ်တယ်ဆိုတာကို စာဖတ်သူ အနေနဲ့ နားလည် သဘောပေါက်စွဲချင်ပါတယ်။

### Variables Declarations in JAVA Programming Language

ဆက်လက်ပြီးတော့ variable Declarations နဲ့ ပတ်သက်တဲ့ Table 2.5 ကို ဆက်လက်လေ့လာကြရအောင်။ Table 2.5 ကို မလေ့လာခင်မှာ variable ဆိုတဲ့ စကားလုံး၊ စကားနေးကို လေ့လာကြရအောင်လား၊ ကျွန်တော်တို့ JAVA Language ကိုပဲ သုံးသုံး C & C++ language ကိုပဲ အသုံးပြုပြု program ရေးတွေမယ်ဆိုရင် Program မှာ ပါဝင်အသုံးပြု တွက်ချက်ရေးသား ရမယ့် variable constant တွေကို Declare လုပ်ပေးပါတယ်။ ကဲ မျက်စိတဲ့မှာ ရှင်းရှင်းလင်းလင်းပေါ်သွားအောင် Sample ရေးပြုပါမယ်။

Sample

```
int          a; // Declare variable use int data type
↑           ↑
data type   variable
```

ဒီပုံစိတ် လေ့လာကြည့်လိုက်ရင် data type နဲ့ variable ဆိုတာကို ခွဲခြားသိမြင်သွားမှာပါ။ အဲဒီလို့ Declare လုပ်ရမှာ Variables and constant type တွေ ပုံစွဲပြားမှုတွေရှိပါတယ်။ အဲဒီ အချက်အလက်တွေကို အခြေခံကျကျ လေ့လာကြည့်ကြရအောင်။ variable တွေကို Declare လုပ်ကြည့်တဲ့အခါမှာ case sensitive ဖြစ်တယ်ဆိုတဲ့အချက်ကို ရှုံးသင်ခန်းစာတွေမှာ ဦးချေ ပေးခြံးပါပြီ။ ဥပမာအနေနဲ့ပြောရရင် အပေါ်မှာ Sample ရေးပြုခဲ့တာဟာ a ဆိုတဲ့ variable နော်။ ဒါကို program မှာ ရေးသား အသုံးချ use လုပ်တဲ့အခါမှာ A လို့ ခေါ်ယူလိုက်တယ် ဆိုကြပါစို့။ error တက်လာမှာပါ။ နောက်ထပ် ဥပမာပေးရရင်တော့ Sum လို့ Declare လုပ်ခဲ့တယ်ဆိုကြပါစို့။ ခေါ်ယူအသုံးချတဲ့အခါမှာ sum လို့ အကွဲရာအစစာလုံးအသေးနဲ့ ခေါ်ယူလိုက်ရင် error တက်လာမှာပါ။ အဲဒါဘာကြောင့်လဲဆိုတော့ Declare လုပ်နဲ့တဲ့ variable name နဲ့ ခေါ်ယူအသုံးချတဲ့ variable ဟာ match မဖြစ်တော့လို့ပါပဲ။ အဲဒီလို့ဖြစ်တာကို case sensitive ဖြစ်တယ်လို့ ခေါ်ပါတယ်။

**NOTE:** QBASIC,BASIC,Visual Basic, Pascal စိုးတာသင်ခန်းစာ တွေက case sensitive မဖြစ်ပါဘူး။ Variable အနည်း

ငယ်ကွဲလွှဲမှုကို လက်ခံပါတယ်။ C & C++ Java စိုးတာသင်ခန်းစာ တွေကတော့ case sensitive ဖြစ်ပါတယ်။ အဲဒီအတွက် အမျိုးမတူတဲ့ Variable တွေကို ရေးပြပါမယ်။

Sum,SUM,sum,SUM, suM တို့ဟာ အမျိုးမတူတဲ့ variable တွေလို့ compiler က သတ်မှတ်ပါတယ်။ အထက်ပါ အတိုင်း Declare ပုံစွဲတစ်မျိုးမျိုးနဲ့ ကြည္လာခဲ့ရင် ကိုယ်ကြည္လာခဲ့တဲ့ type အတိုင်းပြန်ခေါ်ယူမှုသာ Match ဖြစ်ပြီး error မတက်မှာပါ။ ကဲ အဲဒီ အချက်ဟာ case sensitive ဖြစ်တဲ့ သဘောတရားပါပဲ။

ကဲ variable တွေကို Declare လုပ်တဲ့အခါမှာ compiler က လက်ခံတဲ့ valid variable name နဲ့ Invalid variable name ဆိုပြီး ပုံစွဲပြားတာကို Table 2.5 အနေနဲ့ ရေးသားပို့ချ ပေးထားပါတယ်။ သေသေချာချာလေ့လာကြည့်ပါ။

Valid Variable name allowed in JAVA Language compiler					
a	b1	findarea	Name	cal_area	input1

**Table 2.5 valid variable name allowed in JAVA Language compiler**

ဒီ Table 2.5 ဟာ JAVA Language compiler ကလောက်ခဲ့တဲ့ valid variable name တွေပါပဲ။ အကွဲရာစာလုံးတွေဖြစ်ရင် ဖြစ်ပေါ်အတွေးအသေးတွေ ဖြစ်နေရင်ဖြစ်ပေါ် cal-area ပုံစွဲမှာလို့ underscore ( \_) ကိုဖြစ်ပေါ်လေ့လာကြည့်ပါ။

Invalid Variable name no allow use in JAVA Language compiler	
int	no allow in Java Language data type
for	no allow in Java identifier keyword
4th	the first character must be a letter
"a"	illegal character double quote ("") or single quote(,)
cal-area	illegal operator (-), no allow use mathematical operator
find area	no blank space allowed in variable constant

**Table 2.6 Invalid variable name no allow use in Java Language compiler**

ဒီ Table 2.6 မှာ ရေးသားထားတဲ့ variable type တွေဟာ Java Language compiler က လက်မခံတဲ့ variable type တွေပါပဲ။ ဒီတော့ ဒီ Table 2.6 မှာပါတဲ့ variable တစ်ခုချင်းစီ အလိုက် လေ့လာကြရအောင်။ ဒါမှသာ ဘာကြောင့် compiler

ကနေ လက်မခံတာလည်း ဆိုတဲ့အချက်ကို ရှင်းရှင်းလင်းလင်းနားလည်းသောပေါက်မှာပါ။ ကဲ လေ့လာကြရအောင်။

Table 2.6 ဟာ Invalid variable Type ကိုရေးဖြထားတာ နောက်။ ဒီတော့ int ဆိုတဲ့ variable ကိုရေးပြီး Declare လုပ်လို့ မရပါဘူး။ programming လုပ်ရာမှာ data type တွေကို variable အနေနဲ့ Declare လုပ်လို့မရပါဘူး။ int မှ မဟုတ်ပါဘူး။ long, double အစရိတဲ့ data type မှန်သမျှကို variable အနေနဲ့ Declare လုပ်ခွင့်မရပါဘူး။

နောက် Table 2.6 က ဒုတိယအကွက်မှာရှိတဲ့ for ဆိုတာ ဟာလည်း Invalid variable ပါပဲ။ ဘာကြောင့်လဲဆိုတော့ switch, case, while, for, do အစရိတဲ့ <variable type မဟုတ်တဲ့ > JAVA Language ရဲ့ identifier keyword တွေ ဖြစ်လိုပါပဲ။ ဒီစကားစအမျိုးအစားတွေကိုလည်း variable အဖြစ် အသုံးပြုလို့မရပါဘူး။

နောက် Table 2.6 က တတိယအကွက် 4 th ဆိုတဲ့ စကားစ ကို လေ့လာကြရအောင်။ အဲဒီအကွက်နဲ့ တစ်တန်းတည်းမှာရှိတဲ့ စကားစလေးကို ဆက်ဖတ်ကြည့်လိုက်ပါဉိုး။ ဒီအခါမှာ the first character must be a letter လို့ရေးထားတာကို တွေ့ရမှာ ပါ။ အဓိပ္ပာယ်ကတော့ variable အဖြစ် Declare လုပ်တဲ့အခါမှာ variable ရဲ့ အစစာလုံးဟာ အကွာရာ (Alphabetically) ဖြစ်ရ မှာပါ။ အစစာလုံးဟာ အကွာရာမဟုတ်ခဲ့ရင် variable အဖြစ် လက်မခံပါဘူး။

နောက်ထပ် Table 2.6 က စတုတွေမြောက်အကွက်ကို လေ့လာကြည့်လိုက်ပါဉိုး။ "a" ဆိုပြီး variable ကို Declare လုပ် ကြညာထားပါတယ်။ variable မှာ " " Double quote , ' ' single quote နဲ့ Declare လုပ်လို့ မရပါဘူး။

နောက် cal-area ဆိုတဲ့ variable ကိုတွေ့ရမှာပါ။ variable - Name တွေကို Declare လုပ်တဲ့အခါမှာ Mathematical Operator တွေဖြစ်တဲ့ <သံဌာန Operator +, -, \*, /> တွေကို ထည့်သွင်းရေးသားလို့ မရပါဘူး။

ကဲ Table 2.6 က အောက်ဆုံးကအကွက်က find area ဆိုတဲ့ variable ကို တွေ့ရမှာပါ။ variable ကို Declare လုပ်တဲ့ အခါမှာ variable -Name လျှော့ကားရာ space ချိန်ထားလို့ မရပါဘူး။

ကဲ အပေါက် Table 2.6 ဟာ Invalid variable Name တွေပါပဲ။ အဲဒီ အချက်အလက်တွေကို စနစ်တကျဖတ်ရှုလေ့လာ ထားပါ။ ဒါမှသာ program ရေးတဲ့အခါမှာ စိတ်တိုင်းကျ variable -Name တွေကို ရေးသားဖန်တီးလို့ရမှာပါ။

ကဲစာဖတ်သူအနေနဲ့ Valid Variable -Name Table 2.5 နဲ့ Invalid Variable - Name Table 2.6 တို့ကိုလေ့လာပြီးရင် တော့ JAVA Language ရဲ့ အဆင့်မြင့် Variable Declarations တွေကို လေ့လာကြည့်ကြရအောင်။

**NOTE:** အပေါက် Table 2.5 နဲ့ Table 2.6 ဟာ JAVA Language မှာသာမက C & C ++ Language အပါအဝင် ဘယ် language မှာမဆို လေ့လာမှတ်သားထားရမယ့် အကြောင်း အရာပုံဖြစ်လိုပါပဲ။

### Variable Declarations in Java Language

C & C ++ programming language မှာရော် Visual Basic, Ada အစရိတဲ့ programming language တွေမှာပါ အနည်းငယ်လောက် ကျမ်းဝင်လေ့လာနဲ့ဖူးရင် Static Variable, final variable အစရိတဲ့ variable Declarations တွေအကြောင်း ကို လေ့လာနဲ့ဖူးမှာပါ။ ဒီတော့ အဲဒီ variable တွေအကြောင်းကို လေ့လာရအောင် Table 2.7 ကို လေ့လာလိုက်ပါဉိုး။ သော် ချာချာလေ့လာပါ။

Keyword	visibility
Static	Declarations, this is a class variable rather than an instance variable.
Final	The value of this member cannot change.
transient	Used in object serialization to mark member variables that should not be serialized.
volatile	Used to prevent the compiler from performing certain optimizations a member. (Java Advanced Feature)

**Table 2.7 Variable Declarations in JAVA programming language**

**NOTE:** အချေးပြထားတဲ့ Table 2.7 မှာတွေ့ရတဲ့ keyword တွေထဲက ကျွန်ုတ်တော်အနေနဲ့ Static Method နဲ့ Final Method တို့ရဲ့အကြောင်းကို ပြန်လည်တင်ပြေးပါမယ်။ သောာတရားကို နားလည်းသောပေါက်နိုင်တော်ပါပဲ။ သတ်ပြုရမယ့်အချက် ကတော့ အခုတင်ပြမယ့် Static keyword ဟာ Method အနေနဲ့ မဟုတ်ဘဲ program မှာ ထည့်သွင်းအသုံးပြုမယ့် variables -Name အဖြစ်အသုံးပြုပို့ကို ပို့ချုပ်ကြားပေးမှာဖြစ်ပါတယ်။

### Static Variables

Static Variables ရဲ့ သောာတရားကို နားလည်အောင် ပြောရရင်တော့ variable -Name တစ်ခုကို Static keyword ကို အသုံးပြုပြီး Declare လုပ်ခဲ့မယ်ဆိုကြပါစို့။ ဒီအခါမှာ Static keyword ရဲ့ စွမ်းရည်ကြာင့် ဒီ program ကို Run လုပ်နေစဉ်

မှာ program ကို စတင်အသုံးပြုစဉ်က ထည့်သွင်းခဲ့တဲ့ Data ရဲ့ တန်ဖိုးကို သိလောင်သိမ်းဆည်းပေးထားနိုင်ပါတယ်။ ဒါ program ကို အသုံးပြုစဉ်မှာတော့ အဲဒီ variable-Name ကို အသုံးပြု၍ ထပ်မံပြီး Data input လုပ်မယ်ဆိုကြပါစို့။ မူလထည့်သွင်းထားတဲ့ Data ရဲ့ တန်ဖိုးထဲမှာပဲ Append လုပ်ပြီး ပေါင်းထည့်သိလောင်ထားတော့မှာပါ။ အဲဒီအခါမှာ static variable - Name ကို အသုံးပြုပြီး Data ကို ထပ်မံဖြည့်သွင်းခြင်း ပေါင်းနှစ်တာ တွေလုပ်ပြီးသွားရင်တော့ Output ထဲတဲ့အခါမှာရောက်ဆဲ index -variable ရဲ့ Data တန်ဖိုးကိုပဲ output အနေနဲ့ ထုတ်ပေးမှာပါ။ ဒါဟာ static variable ရဲ့ သဘောတရားပါပဲ။ အဂျယ်တက္ကနားလည်အောင် ရှင်းလင်းပြောရရင်တော့ static variable ဟာ သာမန် variable တွေနဲ့ လုံးဝမတူပါဘူး။ သာမန် variable တွေဟာ function တစ်ခုကနေတစ်ခုပြောင်းလဲ တွက်ချက်ရောကန် ဖိစိရဲ့ သက်ဆိုင်တဲ့ function ရဲ့ ပြင်ပကိုရောက်သွားရင် အဲဒီ variable ရဲ့ တန်ဖိုးဟာ အလိုအလောက် တန်ဖိုးမဲ့ပြီးပျက်ပြယ်သွားပါပြီ။ static variable ကတော့ ဒါ static variable ကို အသုံးပြုထားတဲ့ function ထဲကို ဝင်လိုက် တွက်လိုက် တွက်ချက်ရေးနဲ့ပဲ လက်ကျော် value ထဲကို ပေါင်းထည့်ဖြည့်စွက်တွက်ချက်ပေးသွားတာပါ။ Looping တွေနဲ့ တွေ့ကြပြီး ပေါင်းစပ်အသုံးပြနိုင်ပါတယ်။ ဒါ static variable ရဲ့ သဘောတရားကို ထည့်သွင်းအသုံးပြနိုင်ကြမှာပါ။

ကဲ Static variable အကြောင်းကို လေ့လာလို့ နားလည် သဘောပေါက်သွားရင် Listing 2.14 အနေနဲ့ ရေးသားပို့ချပေးထားတဲ့ program ကို ပြောင်းလဲ လေ့လာလိုက်ပါ။

#### **Listing 2.14 Passing array arguments to method**

```
1. public class testArr {
2.     private int [ ] int Array;
3.     public testArr ( )
4.     { int Array = new int [ ]
5.         {1, 2, 3};
6.     }
7.     public void change (int [ ]a)
8.     {for (int i = 0 ;i <int
9.         Array.length;i + +)
10.        intArray [i] = 9 - intArray
11.           [i];
10.    }
11.    public int getZero ( ) {re-
```

```
        turn intArray [0] ;
12.    public static void main
13.        (String args [ ]) {
14.        testArr tp = new testArr (
15.        );
16.    }
```

#### **Analysis**

၁။ ဒါ program တည်ဆောက်ထားပဲဟာ array arguments တွေကို pass လဲပါစို့ အတွက် sample အနေနဲ့ တည်ဆောက်ပြထား တာပဲဖြစ်ပါတယ်။ လိုင်းနံပါတ် (၁) ကနေ လိုင်းနံပါတ် (၆) အထိဟာ testArr ဆိုတဲ့ class တစ်ခုကို သီးသန့်တည်ဆောက် ပါတယ်။ အဲဒီ testArr ဆိုတဲ့ class ထဲမှာ constructor function ( ) ကိုရေးသားတည်ဆောက်ပါတယ်။ လိုင်းနံပါတ် (၆) မှာတော့ change (intArray) ဆိုတဲ့ function ကို array passed လုပ်ပြီး တည်ဆောက်စို့ Declare လုပ်ခဲ့ပါတယ်။

၂။ ဒါ change () function ကို အကျယ်ချုပြီးရှင်းလင်းရမယ်ဆိုရင် လိုင်းနံပါတ် (၇) ကနေ လိုင်းနံပါတ် (၁၀) အထိ ရေးသားထားတာကို လေ့လာကြည့်ပါ။ for looping ကို အသုံးပြုပြီးတော့ array ကို တွက်ထုတ်ထားတာပါ။ ဒါပေါ့ ဒီနေရမှာ သတိပြုရမှာကတော့ API Method တစ်ခုဖြစ်တဲ့ Array - Name.length method ကို ခေါ်ယူအသုံးပြုထားတာကိုလည်း တွေ့ရမှာဖြစ်ပါတယ်။ ဒီနေရမှာ အကျယ်ချုပြီး ရှင်းမပြုတော့တာဟာ Array နဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကိုရေး API Method တွေနဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကိုပါ ရှုံးသင်ခန်းစာ chapter တွေမှာ ပို့ချုပြီးခဲ့တာကြောင့်ပါပဲ။ ဒါကြောင့် ဆက်လက်ပြီးတော့ main () function ထဲမှာရေးသားထားတာတွေကို လေ့လာပြီး output ဘယ်လိုထွက်လာသလဲဆိုတာကို စာဖတ်သူကိုယ်တိုင်ပဲ စဉ်းစားလေ့လာပြီးတော့ trace လုပ်ကြည့်ပါတော့။ ကဲစာဖတ်သူအနေနဲ့ passing Arguments နဲ့ပတ်သက်တဲ့ အခြေခံအကြောင်းအရာတွေကို သဘောပေါက်သွားရင် ဆက်လက်ပြီးတော့ 2.4 Object- Oriented and Development Concepts ဆိုတဲ့အပိုင်းကို ဆက်လက်လေ့လာကြပါစို့။

ကျော်ဇူးတော်လေး (ကသာ)



## 2.4 Object - Oriented and Development Concepts with Interface

ကျွန်တော်တို့အနေနဲ့ Class တွေကို တည်ဆောက်မယ်၊ object တွေအဖြစ် ဖန်တီးမယ်ဆိုတဲ့ အနေနဲ့ အရေးကြီးတဲ့ Java ရဲ့ features တွေကို လေ့လာရာမှာ Class Declarations လုပ်ရမှာပါ အရေးကြီးတဲ့ features တွေထိုပါ ဆက်လက်လေ့လာ ကြရအောင်။ ဒါကြောင့် Class ကို တည်ဆောက်တဲ့အခါမှာ class ရဲ့ features တွေကို လိုအပ်သလို ဆွဲဆန့် ဖန်တီးယူနိုင်တဲ့ Interface အကြောင်းကို စတင်လေ့လာကြပါမယ်။

### Implementing an interface

Interface နဲ့ ပတ်သက်ပြီး ရှင်းပြရမှာပါ အထောက်အကူဖြစ်စေနိုင်ဖို့အတွက် J2SE မှာ ပါဝင်တဲ့ အဆိုပြုချက်အတိုင်း ရေးသားတင်ပြေးထားပါတယ်။ စာဖတ်သူအနေနဲ့ Java Language နဲ့ family ဖြစ်လာစေမယ်လို့ ယုံကြည့်ပါတယ်။

- All methods in an interface are abstract methods.
- They are public methods with no implementation. They are to be implemented in the sub classes that use them.
- All data attributes are implicitly declared as static and final in an interface definition .
- All methods declared in an interface are public and abstract.
- An interface can also include constant declarations.

အခုရေးပြထားတဲ့ အချက်အလက်တွေကို လေ့လာတဲ့အခါမှာ ကျွန်တော်အနေနဲ့ လိုအပ်မယ့် features တွေနဲ့ ပတ်သက်တာ တွေကို ရှင်းလင်းပို့ချပေးတာပါ။ Interface နဲ့ ပတ်သက်တဲ့ အချက်အလက်တွေမှာလည်း အတတ်နိုင်ဆုံး ပြည့်စုံအောင် ထည့်သွင်းပေးထားပါတယ်။ ဒီတော့ Interface နဲ့ ပတ်သက်ပြီးတော့ ပို့ပြီး သဘောပေါက်အောင် စာနဲ့ ရှင်းပြနေတာထက် program ရေးပြီး လေ့လာတာပို့ပြီး သဘောပေါက်လွယ်လိမ့်မယ်လို့ ထင်ပါတယ်။ ဒါကြောင့် Listing 2.15 ကို လေ့လာကြည့်ပါ။

Listing 2.15 Illustrate the class implementation of an interface

```

1. class interface {
2.     public static void main (String args [ ] ) {
3.         developement obj = new developement ( );
4.         System.out.printIn ("Output"); // create an object 'obj'
5.         obj.show ( ); // call and use in show () function
6.         obj.get ( ); // call and use in get () function
7.     }
8. } // close brace for interface class
9. interface develop {
10.    void show ( ); // Declarations of show () method
11.    and get ( ) method
11.    void get ( );
12. }
13. Class developement implements develop {
14.     public void show ( )
15.     { System. Out. printIn ("Show ( ) method is
16.     invoked."); }
16.     public void get ( )
17.     { System. out. print In ("get () method is invoked."); }
18. }
```

output

show () method is invoked.  
get () method is invoked.

### Analysis

၁။ Listing 2.15 အနေနဲ့ ရေးသားထားတဲ့ program interface ရဲ့ သဘောတရားကို အပြည့်အဝ ရှင်းလင်း ဖော်ပြနိုင်စွမ်းရှုပါတယ်။ ဒါကြောင့် သေသေချာချာ လေ့လာကြည့်ကြရအောင်။ လိုင်းနံပါတ် (၁) ဟာ interface ဆိုတဲ့ Class - Name ပေးပြီး primary class တစ်ခုတို့ ဖန်တီးပါတယ်။ အဲဒါ ဗိုလ်ချုပ်မှာ main ( ) function ကို ရေးဖွံ့ထားပါတယ်။ လိုင်းနံပါတ် (၂) မှာ ဒါကိုတွေ့မှာပါ။ အဲဒီနောက် လိုင်းနံပါတ် (၃၃) မှာ ရေးသား

ဖန်တီးထားတဲ့ developement ဆိုတဲ့ class ကိုလမ်းခေါ်ယူပြီး  
တော့ object အဖြစ် ဖန်တီးယူလိုက်တာကို တွေ့ရမှာပါ။ အဲဒီ  
နောက် Output ဆိုတဲ့ စာသားကို ရိုက်ထုတ်ပေးနိုင်ဖို့အတွက်  
System. Out. print In ( ) statement နဲ့ ခေါ်ယူအသုံးချပါ  
တယ်။ အဲဒီနောက် output statement အနေနဲ့ဖော်ပြပါ။ Out-  
put - string ကို ရေးသားဖန်တီးတဲ့ show () function နဲ့ get  
( ) function တို့ကို လိုင်းနိုင်ပါတ် (၄) နဲ့ လိုင်းနိုင်ပါတ် (၅) မှာ  
အောက်ပါအတိုင်း ရေးသားခေါ်ယူထားပါတယ်။

obj: show ();

obj: get ();

ဒီလိုခေါ်ယူလိုက်တဲ့အတွက် အောက်မှာရေးပြထားတဲ့ out-  
put တွေ ရလာတာပေါ့။ ဒါဟာ main () function ထဲမှာ flow  
ဖြစ်သွားပုံကို ရှင်းပြထားတယ်နော်။ ဒါပေမဲ့ ဒါဟာ သိပ်တော့မထူး  
ခြားလုပ်ဘူး။ သာမန် Class Declaration လုပ်ပြီး Object  
အဖြစ် ဖန်တီးထားတာ ဒီ program flow အတိုင်းသွားမှာပါ  
ပါ။ ကဲ ဒီတော့ ဘယ်နားမှာ ကွဲပြားခြားနားမလဲဆိုတာ ဆက်လက်  
လေ့လာကြရအောင်။

J။ လိုင်းနိုင်ပါတ်(၆) မှာ develop ဆိုတဲ့ Interface Name ကို  
အသုံးပြုပြီးတော့ Interface package တစ်ခုကို Declaration  
လုပ်ထားတာကို တွေ့ရမှာပါ။ အဲဒီထဲမှာ void show () method  
နဲ့ void get ( ) method နှစ်မျိုးကို Declare လုပ်ခဲ့တာကို  
တွေ့ရမှာပါ။ ဒါပေမဲ့ ဒီ void show ( ) method နဲ့ void get  
( ) method တို့၏ statement expression အပိုင်းတွေကို ရေး  
သားထားခြင်းမရှုပါဘူး။ ဒီ method တွေကို အကျယ်ချွဲဖို့အတွက်  
ဒီ develop ဆိုတဲ့ Interface ကို ထပ်မံမြို့ငြိမ်းပြီးတော့ လိုင်းနှံ  
ပါတ် (၁၄) မှာ developement ဆိုတဲ့ class တစ်ခုကို မြို့ငြိမ်း  
လိုက်ပါတယ်။ ဒါဟာ Inheritance စနစ်ရဲ့ သောာတရားနဲ့  
ဆင်တူရှုရှုနေပါတယ်။ ကွဲပြားခြားနားမှုကို NOTE: အနေနဲ့  
သီးသန်နှင့်ပြပေးပါမယ်။ ဆက်လက်လေ့လာကြည့်ပါ။

R။ developement ဆိုတဲ့ class ဟာ develop ဆိုတဲ့ interface ကို implements လုပ်ထားတာပါ။ ဒီနောက်မှာ အပေါ်၌  
Declare လုပ်ခဲ့တဲ့ void show () method နဲ့ void get ()  
method ရဲ့ statements တွေကို ရေးသားပေးထားပါတယ်။  
အဲဒီကြောင့် လိုင်းနိုင်ပါတ် (၁၄) မှာ ရေးထားတဲ့ void show  
( ) method ကို လေ့လာကြည့်ရင်—  
public void show ()  
{System.out.println("Show () method is invoked.");}  
လို့ တွေ့ရမှာပါ။ ဒါကြောင့် Output statement အနေနဲ့ show  
( ) method is invoked လို့ Output ထွက်လေခဲ့တာပါ။ ဒါဟာ  
ဒီ implements လုပ်ခဲ့တဲ့ class ကို တစ်ဆင့်ခဲ့ အကျယ်ချွဲပြီး

သောာတရားအရ ထပ်မံ ဖြည့်စွက်လိုတဲ့ features တွေနဲ့  
ပေါင်းစပ်အသုံးပြုတာပါပဲ။ ဒါပေမဲ့ Class Declarations ရဲ့  
သောာတရားအရ object အဖြစ် ဖန်တီးရတယ်ဆိုတာပါ မမောပါ  
နဲ့မြို့ပါ။ void get ( ) method ကိုတော့ အကျယ်ချွဲပြီး မရှင်းပြ  
တွေ့ပါဘူး။ စာဖတ်သူကိုယ်တိုင်ပဲ လေ့လာကြည့်ပါ။ void show  
( ) method နဲ့ သောာတရား အတူတူပဲဖြစ်တာကြောင့် နားလည်  
သွားပါလိမ့်မယ်။

**NOTE:** စာဖတ်သူဟာ C++ & VC++ နယ်ပယ်ကလာခဲ့တဲ့  
သူအဖို့တော့ Inheritance ရဲ့ သောာတရားနဲ့ နှင့်ယဉ်ပြီး  
လေ့လာနိုင်ရင် အခက်အခဲလဲ မရှုပါဘူး။ ဒါပေမဲ့ C Programming  
နယ်ပယ်ကလာတဲ့ struct သောာတရား၊ union ရဲ့  
သောာတရားတို့နဲ့ နှင့်ယဉ်ပြီး ထပ်တူထင်မျှ တူညီနိုင်တယ်  
ဆိုတာကို တွေ့ခေါ်မှုပါ။ c programming language ရဲ့  
struct သောာတရားနဲ့ ဆင်တူတယ်။ data group တွေ ဖွဲ့စည်း  
ပေးနိုင်မှုကို လေ့လာနိုင်အောင် Listing 2.16 အနေနဲ့ လေ့လာ  
ကြည့်ပါ။ သင်ကြားပို့ချေပေးထားပါတယ်။

Listing 2.16 Declaring variables and data item

```
1. interface data variable {
2. int a = 1;
3. int b = 2;
4. class inter implements dataVariable {
5. public static void main (String args [ ] ) {
6. System.out.println ("Output");
7. System.out.println ("The value of a is :" + a); // The value of a is :1
8. System.out.println ("The value of b is :" + b); // The value of b is :2
9. }
10. }
```

Output

The value of a is :1

The value of b is :2

### Analysis

၁။ Listing 2.16 က program ဟာ သိပ်အတွေအထူး မရှုပါဘူး။ C programming language က struct ဆိုတဲ့ သောာတရားနဲ့ ဆင်တူပါတယ်ဆိုတာကို ရှင်းပြန့်အတွက် ရေးသားပို့ချေပေးထားတာပါ။ data type တွေနဲ့ အသုံးပြုပြီးတော့ variables တွေကို ကြည့်ရှုပါတယ်။ အဲဒီ variables တွေမှာ တန်ဖိုး value တွေကိုပါ ကိန်းသေထည်သွင်း ကြည့်ပေးထားပါတယ်။ ဒါတွေဟာ interface အနေနဲ့ ဖန်တီးထားခဲ့တာပါပဲ။

J။ လိုင်းနံပါတ် (၄) မှာ inter ဆိုတဲ့ class ကို ဖန်တီးယူပါတယ်။ အဲဒီ implements လုပ်ထားတဲ့ inter ဆိုတဲ့ class ဟာ အပေါ်က interface ကို implements လုပ်ထားခြင်းပါပဲ။ အဲဒီ နောက် main () function ကိုရေးစွဲတယ်။ အဲဒီ main () function ထဲမှာတောင်ပဲ Output statements တွေကို ရိုက်ထုတ်ထားတာပါ။ ဒီ program ဟာ C programming language ရဲ့ Struct ရဲ့ သဘောတရားနဲ့ တူညီနိုင်တဲ့သဘောတရားကို နားလည် သဘောပေါ်သွားနိုင်အောင် ရေးသားပို့ချေပေးထားတာပါ။ ဒါအိုရင် အခြေခံ interface ရဲ့ ဖွဲ့စည်းတည်ဆောက်ပဲ implements အနေနဲ့ ဖန်တီးရယူပဲကို နားလည်သွားရင် ရှေ့မှာ ဘယ်လိမ့်မျိုး အသုံးဝင် အသုံးပြနိုင်မယ်ဆိုတာကို လေ့လာကြရအောင်။ ဒီတော့ Class Declaration keywords တွေနဲ့ ပတ်သက်တဲ့ အချက် အလက်တွေကို Table 2.8 အနေနဲ့ ရေးသားပို့ချေပေးထားပါတယ်။ လေ့လာလိုက်ကြရအောင်။

ဒီ Table 2.8 ဟာ class Declarations အရ Declare လုပ်နိုင်တဲ့ keyword တွေပါပဲ။ ဒီတော့ ဒီ class keyword တွေကို တစ်ခုချင်းစီ သီးသန်လေ့လာကြည့်ကြရအောင်။ အောက်မှာ ရှင်းပြထားပါတယ်။ သေသေချာချာလေ့လာကြည့်ပါ။

**Table 2.8 keyword and usage of the class Declarations In JAVA language**

public keyword	By default , a class can be used by other classes in the same package. The public modifier declares that the class can be used by any class regardless of its package.
Abstract keyword	Declares that the class can not be instantiated.
Final keyword	Declares that the class can not be sub- classed.
Class keyword	The class keyword indicates to the compiler that this is a class declaration and that the name of the class is Name - Of- Class.
extends super	The extends clause identifies super as the super -class of the class, there by inserting the class within the class hierarchy.
implements interfaces	The declare that your class implements one or more interface, use the keyword implements followed by a comma - delimited List of the names of the interface implements by the class.

### public keyword

ဒီ public keyword အကြောင်းကို အရှေ့ပိုင်းမှာ သီးသန်း သင်ကြားပို့ချေပေးခဲ့ပါပြီ။ သဘောတရားဟာ သာမဏ် variable Declaration လုပ်တဲ့ သဘောတရားနဲ့ ဆင်တူပါတယ်။ class တစ်ခုမှ ပုံပြုပါတယ်။ အဲဒီ Class ကို အသုံးပြထားတဲ့ package အတွင်း မှာ လိုအပ်သလို အသုံးချိန်ခွင့်ရှိခြင်းဟာလည်း ပုံပြု ပါပဲ။ အရှေ့မှာ keyword တွေရဲ့ သဘောတရားတွေကို ရှင်းပြုပဲနဲ့ ဆင်တူတာကြောင့် အသေးစိတ် ရှင်းပြတော့တာပါပဲ။ ဒီတော့ public keyword နဲ့ ပတ်သက်ပြီး တော့ ဒီလိမ့်ထားကြရအောင်လား။ public keyword is Declare, that class is can be used in any other class in the same package and modifier that class can be usedလို့ အလွယ်မှတ်ကြတာပေါ်နော်။ ကဲ ဒီတော့ ဆက်လက်ပြီး final keyword အကြောင်းကို ဆက်လက်လေ့လာကြရအောင်။

### Final keyword

Final keyword ကို အသုံးပြု Declare လုပ်ပြီဆိုတာနဲ့ အဲဒီ Class တစ်ခုကနေပြီးတော့ can not be overridden ပေါ်နော်။ အသုံးချွမ်းမရှိဘူးလို့ဘဲ အလွယ်မှတ်ထားကြရအောင်။

### Abstract Keyword

Abstract keyword ရဲ့ သဘောတရားကိုတော့ that keywords and use, that the class can not be instantiated လိုပဲ အလွယ်မှတ်ထားကြရအောင်ပါ။

### Class keyword

Class ဆိုတဲ့ keyword နဲ့ ပတ်သက်တဲ့ form တွေ၊ သဘောတရားတွေဟာ ပြီးခဲ့တဲ့ သင်ခန်းစာမှာ အသေးစိတ်ခဲ့ပြီးပါပြီ။ အဲဒီ ကဏ္ဍကို ပြန်လုန်လေ့လာခြင်းအားဖြင့် အလွန်အရေးပါတဲ့ Class keyword အကြောင်းကို အပြည့်စုံဆုံး လေ့လာ သဘောပေါ်ပြီးသားဖြစ်သွားမှာပါ။ အကြမ်းပျိုးအားဖြင့် ဒီတော့ Class keyword ကို အသုံးပြုခြင်းအားဖြင့် အမျိုးအစားမတူတဲ့ data types တွေကို အသုံးပြု ပြီးတော့ အမျိုးမျိုးသော Name တွေနဲ့ variables တွေကို ဖန်တီးရယူနိုင်ပါတယ်။ ပြီးတော့ function () / method အမျိုးမျိုးကို တစ်နေရာတည်းမှာ စုစုပေါင်းတည်ဆောက်ထားနိုင်ပါတယ်။ အဲဒီလို့ နေရာပေး တည်ဆောက်နိုင်တာဟာ class ဆိုတဲ့ keyword ကြောင့်ပဲဆိုတာကို နားလည် သဘောပေါ်ထားရပါမယ်။ Note: ဒါမေ့သိပြုရမှာကတော့ JAVA Language ဟာ Class type တွေနဲ့ပဲ ပုံစံပြု တည်ဆောက်ထားတာ ဖြစ်ပါတယ်။ ဒီအချက်ကိုတော့ ထည့်သွင်းသတိပြု

စဉ်းစားစေလိုပါတယ်။ main () function / method ဟာလဲ class ထဲမှာ ထည့်သွင်းတည် ဆောက်ရတာပဲ ဖြစ်ပါတယ်။

### **extends super**

ဒီ keyword ဟာ Inheritance အကြောင်းကို သင်တွေးစိုးချက် တွေ့ခဲ့ရပြီးသားပါ။ Class တစ်ခုနဲ့ တစ်ခု extends လုပ်ပြီး အသုံးပြုလိုရတာပါ။ ဒါဟာ Class hierarchy စနစ်၏ ပုံစံ ပြုထားတဲ့ keyword ပါပဲ။ ဒီ keyword ဟာလဲ အခု လက်ရှိပို့ချေနေတဲ့ သင်ခန်းစာမှာ ဆက်နှံယ်ချက်တစ်ရပ်အနေနဲ့ ပါဝင်နေပါတယ်။ ဒါကို စာဖတ်သူအနေနဲ့ လေ့လာတွေ့ရှိရမှာပါ။

### **implements interfaces**

ဒီနေဂျာမှာ Interface keyword ကိုသာ Declare လုပ်အသုံးပြုခဲ့ရင် အခြားသော Class, one or more class တွေကနေ implements လုပ်နိုင်ပါတယ်။ ဒီအနေအထားတွေကို ရှုံးမှုပို့ချေပေးခဲ့တဲ့ Listing 2.15 နဲ့ Listing 2.16 တို့ကို သေသေချာချာလေ့လာကြည့်ပါ။ ဒါဟာ interface နဲ့ ပတ်သက်ပြီး အခြားကျကျ ရှင်းပြုပို့ချေပေးထားတာပါပဲ။ စာဖတ်သူအနေနဲ့ အလွယ်တကူ နားလည်သောပေါက်သွားမှာပါ။ interface နဲ့ ပတ်သက်တဲ့ သဘောတရားတွေကို သီးသနိုင်းချေနေတာထက် Listing တွေနဲ့ program ရေးသားပို့ချေပေးတာဟာ ပိုပြီး ပြည့်စုစုပေါ်လွယ်ကူစေလိမ့်မယ်လို့ ထင်ပါတယ်။ ဒါကြောင့် interface ရဲ့ implements သဘောတရားနဲ့ extends လုပ်တဲ့ သဘောတရားတွေပါဝင်တဲ့ Listing 2.17 ကို ဆက်လက်လေ့လာကြည့်ပါ။ နားလည်သောပေါက်သွားပါလိမ့်မယ်။

Listing 2.17 Extending the methods in an interface

```
1. interface inter
2. { void show () ; void get () ; }
3. interface inter2 extends inter // extending the interface "inter"
4. { void hello () ; }
5. Class inter2 implements inter {
6. public void show () { System.out.printIn ("Show method is invoked"); }
7. public void get () { System.out.printIn ("Get mehtod is invoked"); }
8. public void hello () { System.out.printIn ("hello method is invoked ") ; }
9. }
10. Class extends Class {
11. public Static void main (String args [ ] ) {
12. inter2 i2 = new inter2 ( ) ; // creating object for "inter 2"
```

13. System.out.printIn ("Output");

14. i2.show ();

15. i2.get ();

16. i2.hellow( );

17. }

18. }

output

show method is invoked

get method is invoked

hello method is invoked

### **Analysis**

၁။ ဒီ program ထိုးလေးဟာ interface ကို extending method အနေနဲ့ ချုံတွင်အသုံးချေတဲ့ သဘောတရားတွေ ပါဝင်ပါတယ်။ ပထမလိုင်းနံပါတ် (၁) မှာ interface ဆိုတဲ့ keyword ကို အသုံးပြုပြီး inter ဆိုတဲ့ interface - Name နဲ့ method Group တစ်ခုကို ရေးဖွဲ့ပါတယ်။ အဲဒီ Method Group ထဲမှာ void show () method နှစ်ခုကို Declare and line - up လုပ်ခဲ့ပါတယ်။ အဲဒီ method တွေဟာ statements and Expression တွေကို ထည့်သွင်းရေ့ဖွဲ့ထားခြင်း မရှိပါဘူး။ method ရဲ့ form ချေပေးထားခြင်းဆိုရင်လဲ မမှားပါဘူး။

၂။ အဲဒီနောက် inter ဆိုတဲ့ interface ကိုမိုးပြီးတော့ inter 1 ဆိုတဲ့ နာမည်နဲ့ implements လုပ်ပါတယ်။ ဒါဟာ အပေါက inter 1 ဆိုတာကို မိုးပြီးတော့ inter အမည်နဲ့ extends လုပ်ထားတာနော်။ အဲဒီ inter ထဲမှာတော့ hello () ဆိုတဲ့ method ကို Declare လုပ်ထားပါတယ်။ ဒါဟာ အလွန် ရှိုးရှင်းစွာနဲ့ စွမ်းရည်ဖြင့် စိုးပြီးထားခြင်းမျိုးပါပဲ။

**NOTE:** ဒီပုံစံဟာ Class Hierarchy & Inheritance Action ငုံးပြုးပါပဲ။ C & C ++ သမားအတွက်တော့ ကျမ်းဝင်ရင်းနှီးပြီးသားဖြစ်ပါလိမ့်မယ်။ C programming language သမားတွေ အတွက်ကတော့ Struct type နဲ့ ဆင်တူတာကို အခန်းရဲ့ အရွှေ့မှာ သင်တွေးပို့ချေပေးခဲ့ပြီးပါပဲ။

၃။ လိုင်းနံပါတ် (၅) မှာတော့ inter 1 ကို implements လုပ်ပြီးတော့ inter 2 ကို ရေ့ဖွဲ့ပါတယ်။ အပေါက interface အတွင်းမှာ method တွေရဲ့ statements & Expression တွေကို အဲဒီ inter 2 ဆိုတဲ့ Class ထဲမှာမ အကျယ်ချုံပြီးတော့ ရေးဖွဲ့မှာဖြစ်ပါတယ်။ အဲဒီကြောင့် လိုင်းနံပါတ် (၆),(၇),(၈) မှာ

```

public void show () {System.out.printIn ("show
method is invoked");
public void get () {System.out.printIn ("get
method is invoked");
public void hello () {System.out.printIn ("hello
method is invoked");
}
    
```

ဆိုပြီး ရေးသားပါတယ်။ လိုင်းနံပါတ် (၉) မှာတော့ Class in-
ter 2 ရဲ့ Close brace } ကိုလာပိတ်ထားပါတယ်။ ဒီအထိဟာ
object အဖြစ် ဖန်တီးရယူမယ့် class & interface တွေကို ဖန်တီး
ရယူမယ့်အပိုင်းဖြစ်ပါတယ်။ သေသေချာချာ ဂရတုံးကို လေ့လာ
သုတေသနပါတယ်။

၄။ လိုင်းနံပါတ် (၁၀) မှာ extends Class ဆိုတဲ့အမည်နဲ့ primary class ကို ဖန်တီးပါတယ်။ အဲဒီ primary class ထဲမှာမှ
main () function ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၁၂)
မှာ inter 2 ဆိုတဲ့ class ကို အခြေခြားပြီး i2 ဆိုတဲ့ class ကို
လိုက်ပါတယ်။ inter 2 class ဟာ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ class
တွေ interface တွေရဲ့ အကျယ်ဖြန့်ရောဖွံ့ဖြိုးမှုတွေ ပါဝင်နေတဲ့
အတွက်ပါပဲ။ အဲဒီနောက် System.out.printIn ("Output");
ဆိုတဲ့ statement ကို ရေးသားခြင်းအားဖြင့် output ဆိုတဲ့
String ကို ရုံက်ထုတ်ဖို့ ရေးသားလိုက်ခြင်းပါပဲ။ ဒီတော့ ဆက်
လေ့လာကြည့်မယ်ဆိုရင် လိုင်းနံပါတ် (၁၄), (၁၅), (၁၆) မှာ
in. show ();
in. get ();
in. hello ();

စသဖြင့် အပေါ်ကရေးသားဖန်တီးခဲ့တဲ့ interface ရဲ့ inner-
line - method တွေကို လုပ်းပြီးပေါ်ပေါ်လိုက်ပါတယ်။ ဒါကြောင့်
Output အနေနဲ့ ဖော်ပြထားတဲ့

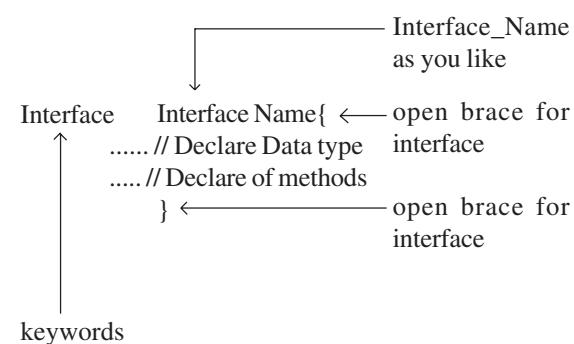
Show method is invoked  
get method is invoked  
hello method is invoked

String တွေဟာ Output အဖြစ်ထွက်လာတာပါပဲ။ သေသေချာချာ
လေ့လာကြည့်ပါ။ အလွန်ရိုးရှင်းစွာ ဖွံ့ဖြိုးတည်ဆောက်ထားတာ
ပါပဲ။ လိုင်းနံပါတ် (၁၇) မှာကတော့ main () function ရဲ့
close brace } နဲ့ လိုင်းနံပါတ် (၁၈) မှာတော့ extends Class
ဆိုတဲ့ primary Class ရဲ့ Close brace } ကို လာပိတ်ထားပါ
တယ်။

ဒါဟာ interface ရဲ့ သောာတရား၊ implements ရဲ့
သောာတရားနဲ့, extends လုပ်ခြင်းရဲ့, သောာတရားတွေကို
ဆက်နွယ်ချက်တွေနဲ့ ရှင်းပြရေးသားပေးထားတာပါ။ interface
ရဲ့ သောာတရား၊ implements ရဲ့ သောာတရားတွေကို နား
လည်ရင် Object- Oriented - programming ရဲ့ သောာ
သာာဝကိုပါ နားလည်လာလိမ့်မယ်လို့ ယုံကြည်ပါတယ်။

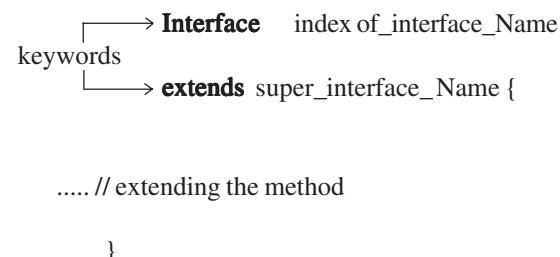
### Implementing Multiple Interfaces

Interface တွေကို ဖန်တီးပြီးတော့ implements လုပ်တယ်။
extending လုပ်တယ်။ အစရိတ္တာ သောာတရားတွေဟာ အလွန်
ရိုးရှင်းပြီးတော့ C++ programmer တွေ အဖို့တော့ လွယ်လွယ်
ကူကူ သောာပေါက်နားလည်နိုင်မှာပါ။ အခြားသော Object -
Oriented - programming (OOP) Language တွေမှာ multiple inheritance သောာတရားတွေ ပါဝင်ပေါင်းစပ် ဖွံ့ဖြိုးတည်း
ထားပါတယ်။ သောာတရားအရ ရှင်းပြရမယ်ဆိုရင် အချိုးချိုး
သော superclass အသီးသီးကနေ အဆင့်ဆင့်ဖြို့ပြုးနိုင်တဲ့
စွမ်းဆောင်ရည်မျိုးကို တွေဖြင့်ရမှာပါ။ interface တစ်ခုကို
သီးသန်တည်ဆောက်မယ်ဆိုရင် အောက်ပါ Syntax - form ပုံစံ
မျိုးနဲ့ တွေဖြင့်ရမှာပါ။



### Basic Syntax - form of Interface

၃ Basic Syntax- form ပုံစံလေးဟာ interface တစ်ခုကို
ဘယ်လိုဖန်တီးရယူမယ် တည်ဆောက်နိုင်မယ်ဆိုတာကို အခြော
ကျကျ တည်ဆောက်ပြထားတဲ့ form ပုံစံပါပဲ။ က ဒီတော့ interface
အချင်းချင်း extends လုပ်တဲ့ form ပုံစံလေးကို
ရေးသားပြပေးပါမယ်။ လေ့လာကြည့်လိုက်ပါပြီး။



### Basic Syntax form of extending the Interface

ဒီ Syntax form ကတော့ interface အချင်းချင်း extends လုပ်တဲ့ သဘောတရားကို နားလည်သောပါက်အောင် အပြော ကျကျ ရင်းဖြေပေးထားတာပါ။ ဆက်နှစ်ယူ သဘောတရားဘက် ကနေ လေးလာကြည့်မယ်ဆိုရင် index Of - interface -Name ဟာ ထင်မိပြီး တည်ဆောက်တဲ့ interface တစ်ခုပါ။ extends ဆိုတဲ့ keyword ရဲ့ နောက်က Super - interface -Name ဆိုတာကတော့ အပေါ်မှာမှုလိုပြီးသား၊ တည်ဆောက်ပြီးသား interface တစ်ခုပါပဲ။ ဒါကိုနိုင်းမြို့ extends လုပ်တာပါ။ သေသေချာချာ လေးလာကြည့်မယ်ဆိုရင် ရှေ့မှာ သင်ကြားပို့ချဲ့တဲ့ Listing တွေမှာ တွေ့ဖြင့်ခဲ့ရပြီးသားပါ။ အပြောင်းအလဲအနေနဲ့ရော family ဖြစ်သွားအောင်ပါ။ Listing တွေကို အရင်ပို့ချုပြီးမှ Syntax form တွေကို ရေးသားပို့ချုပ်ငြိုပ်ဖြစ်ပါတယ်။ ကဲ ဆက်လက် ပြီးတော့ interface ကို implements လုပ်တဲ့ သဘောတရားနဲ့ Class တစ်ခုအတွင်းမှာ ဖွဲ့စည်းတည်ဆောက်တဲ့ သဘောတရား တွေပါဝင်တဲ့ Basic Syntax- form ပုံစံကို လေးလာကြည့်ရွယ်ပေါင်း။

```
class class_Name implements interface_Name {
    .... // implements methods
    .... //
}
```

### Basic Syntax form of implements an Interface

ဒီ form ပုံစံလေးမှာ ရှေးသားတည်ဆောက်ပြုခဲ့တဲ့ Class ပုံစံမျိုးကို listing 2.17 မှာ စာဖတ်သူအနေနဲ့ တွေ့ဖြင့်လေးလာ ခဲ့ရမှုပါ။ ဒီနေရာ အရေးကြီးတဲ့အချင်းကတော့ အပေါ်မှာ တည်ဆောက်ခဲ့ကြတဲ့ interface တွေကို implements လုပ်ပြီးတော့ class တစ်ခုအဖြစ် နောက်ပြီးတော့ အလုပ်အလက်တွေပါပဲ။ ဒါကို နားလည်သောပါက်အောင် အထပ်ထပ်အောက်ဖော်ပြီး လေးလာစေလိုပါတယ်။

ကဲ ဒါဆိုရင်တော့ အရှေ့မှာ ပို့ချဲ့တဲ့ listing တွေနဲ့ပတ်သက်တဲ့ Syntax- form တွေကို အသီးသီးပို့ချေပေးပြီးပါပြီ။ အခုံဆက်လက်ပြီးတော့ Implementing Multiple Interfaces ဆိုတဲ့ အပိုင်းကို ဆက်လက်သင်ကြားပို့ချေပေးပါမယ်။ ဆက်လက်ပြီးတော့ လေးလာကြည့်လိုက်ပါပြီး။

### NOTE: Implementing Multiple Interface ဆိုတဲ့

သင်ခန်းစာကို မလေ့လာခင်မှာ Certification Practice အဖြစ် လေ့လာ သင်ယူနေကြတဲ့သူတွေအတွက် အောက်ပါ Theory သဘောတရားလေးကို အထပ်ထပ်အောက်ဖတ်ရှုလေ့လာထားစေချင်ပါတယ်။ အဲဒါကတော့-Interface, like classes, can belong to a package. Interface can also import other interfaces and classes from other packages, just as classes can.

ကဲ သဘောတရားကို နားလည်သောပါက်သွားရင် Multiple Interfaces တွေကို Implements လုပ်တဲ့အပိုင်းမှာ အရေးကြီး တဲ့ Syntax - form ကို လေးလာကြည့်ကြရအောင်။ interfaces အများအပြားကို ကိုယ်ပိုင် class ရဲ့ အတွင်းမှာ ပေါင်းစပ်ပြီးတော့ implements လုပ်တာပါပဲ။ ဒီအောက် Multiple interface တွေကို class တစ်ခုအတွင်းမှာ implement လုပ်တယ် ဆိုကြပါစိုး။ အဲဒါ interface တွေရဲ့ Name တွေကို commas တွေခံပြီးတော့ ရေးရတာပါ။ ဒီတော့ အဲဒါ Implementing Multiple Interface ရဲ့ Basic Syntax- form ကို ရေးပြပါမယ်။ လေးလာကြည့်ပါ။

```
public class AnimatedSign extends javax.swing.JApplet implements Runnable , Ovservable {
    //-----
    }
```

### Basic Syntax- form of the Implementing Multiple Interface

ကဲ ဒီ syntax- form ပုံစံကို လေးလာခြင်းအားဖြင့် interfaces အများအပြားကို implements လုပ်ပြီးတော့ Class တစ်ခုအတွင်းမှာ ပြန်လည်တည်ဆောက်နိုင်တဲ့ သဘောတရားတွေကို နားလည်သောပါက်သွားမှုပါ။

အကယ်၍ interface အတွင်းမှာ ပါဝင်တဲ့ parameter တွေဟာ different parameter list တွေဖြစ်ကြပါတယ် ဆိုကြပါစိုး။ အဲဒါ method တွေအတွက် ဖြေရှင်းပုံကတော့ အလုန်ရှိရင်းပါတယ်။ အဲဒါကတော့ ရှေ့မှာ ကျွန်ုတ်သင်ကြားပို့ချုပြီးသား Object- Oriented - programming (OOP) paradigm တွေနဲ့ ဖြေရှင်းရမှုပါပဲ။ ဒါကို နားလည်သောပါက်အောင် ရှင်းပြပါမယ်။ interface နှစ်ခုကို နိုင်းပြီးတော့ class တစ်ခုကို တည်ဆောက်မယ်၊ implements လုပ်မယ် ဆိုကြပါစိုး။ အဲဒါ interface နှစ်ခုအတွင်းမှာ အမည်တဲ့ method နှစ်ခုပါဝင်နေတယ် ဆိုကြပါစိုး။ အဲဒါ method တွေဟာလည်း different parameter of data type တွေဖြစ်တယ်ဆိုကြပါစိုး။ ကျွန်ုတ်တို့အနေနဲ့

Method Overloading ကို အသုံးပြုပြီးတော့ ဖြေရှင်းရမှာပါ။ ဒါဟာ ရှေ့ မှာ သင်ကြားပို့ချခဲ့တဲ့ နည်းပညာတွေကို အသုံးပြုပြီး ဖြေရှင်းတာပါပဲ။ အဲဒါကို နားလည်အောင် ပြန်လည်သုံးသပ် ရှင်းပြတာကို အထပ်ထပ်အခါခါ လေ့လာကြည့်ပါ။

တကယ်လိုမှား အဲဒီ Interface နှစ်ခုအတွင်းမှာပါဝင်တဲ့ method တွေဟာ parameter list တွေကလည်း တူညီကြပါတယ်။ ဒါပေမဲ့ return type တွေက ကွဲပြားခြားနားတယ် ဆိုကြပါစို့။ ဒီအခါ implements လုပ်တည်ဆောက်ထားတဲ့ class ကို compile လုပ် Run လုပ်လိုက်ရင် compilerကနေ error message ပေးလာမှာပါ။ ဒီအခါမှာတော့ ပြန်လည်ပြီးတော့ interface အပိုင်းကို ပြန်လည်စစ်ဆေးစို့ ရေးသားပြုပြင်စို့ လိုအပ်မှာပါ။ ဒါဟာ Multiple interface တွေကို implements လုပ်တယ်ဆိုတဲ့ အပိုင်းနဲ့ ပတ်သက်ပြီးတော့ ရေးသားပို့ချပေးတဲ့ အချက်အလက်တွေကို သေသေချာချာ လေ့လာဖတ်ရှုစို့ လိုအပ်တယ်ဆိုတာကို ထောက်ပြတာပါပဲ။

ကဲ ဆက်လက်ပြီးတော့ listing program အနေနဲ့ ရေးသားပို့ချုပ်မှု interface ကို abstract Class ငုံသွေ့နှစ်ဖိုးမယ်၊ implements လုပ်မယ်ပေါ့။ အဲဒီ abstract class ကို extends လုပ်ပြီးတော့ class တစ်ခုကို ထပ်မံတည်ဆောက်မှာပါ။ ပြီးတော့ main () function ထဲမှာမှ Object အဖြစ် ဖန်တီးအသုံးချုပ် ပုံစံကို လေ့လာသင်ယူရမှာပါ။

**NOTE:** Implementing Multiple interfaces အပိုင်းကို အခြေခံလောက်ပဲ သင်ကြားပို့ချပေးထားပါတယ်။ ရှေ့သင်ခန်းစာနဲ့ Listing တွေမှာ ထပ်မံဖြည့်စွက် ပို့ချပေးသွားပေးပို့မှာပါ။ Rogers Cadenhead and laura lemay ရေးသားတဲ့ Teach yourself Java 2 in 21 days ထဲမှာ ဒီ Multiple interfaces အကြောင်းကို အသေးစိတ်ရှင်းပြထားပါတယ်။ ကျွန်တော့ အနေနဲ့ မို့င်းယဉ်ပြီးယူသာ ရှင်းလင်းပို့ချပေးမှာပါ။ web site ကနေ လေ့လာချင်တယ်ဆိုရင်တော့ www.java 21 pro . com မှာ လေ့လာနိုင်ပါတယ်။

ဒီတော့ ကျွန်တော့ အနေနဲ့ Multiple Interface တွေကို အသုံးပြုရင်တွေကြံ့ရမယ့် အခက်အခဲတွေကို အကြမ်းအားဖြုံးတော့ သင်ကြားရှင်းလင်းပြပေးပြီးပါပြီ။ ဒီတော့ စာဖတ်သူအနေနဲ့ မေးကောင်းမေးနေနိုင်မယ့် overloading သဘောတရားကိုလည်း အကျယ်ရှင်းပြရမှာဖြစ်သလို Certification Practice အနေနဲ့ လေ့ကျင့်လိုသူတွေအတွက် ကျွန်တော့ပို့ချပေးမယ့် Q&A ကဏ္ဍကို မရောက်ခင်မှာ listing 2.18 ကို ကြားဖြတ်သင်ခန်းစာအဖြစ် ပို့ချပေးမှာဖြစ်ပါတယ်။

### Creating Methods with the same name Different Arguments

ဒီခေါင်းစဉ်လေးကို တွေ့ရရှင်တော့ စာဖတ်သူအနေနဲ့ ကျွန်တော်ဘာကို ဆိုရိုတာလဲဆိုတာကို သဘောပေါက်သွားမှာပါ။ Multiple interface တွေကို အသုံးပြုရင် Same Name, Same Arguments တွေ Same Name , Different Arguments တွေ အစိုးရဖြင့် တွေကြံ့ရမယ်ပါ။ ဒီအခါမှာ interface ရဲ့ စွမ်းရည်ကို အပြည့်အဝ အသုံးချိန်မယ့် အခြေခံ object- concepts တွေကို လည်း တတ်မြောက်မှ ဖြစ်နိုင်မယ့်မဟုတ်လား။

အဲဒီကြောင့် listing 2.18 မှာ Creating Same Name, Different Arguments with JAVA method လို့ အမည်လေးပြီး program code - line <58> ကြောင်းရှိတဲ့ program ငယ်လေးကို ပို့ချပေးလိုက်ပါတယ်။ ဒီ program ငယ်လေးမှာ ထူးချွေကဏ္ဍတဲ့ 2.1 Introduction to Classes ဆိုတဲ့ အနေးကဏ္ဍကို သင်ကြားပို့ချစွမ်းက လေ့လာခဲ့ရမှာဖြစ်တဲ့ this keyword အကြောင်းကို မှတ်မိမှာပါ။ listing 2.4 အနေနဲ့လည်း program ရေးသားလေ့လာဖူးမှာပါ။ အဲဒီ this keyword ကို လည်း ဒီ program မှာ ပေါင်းစပ်ထည့်သွင်းပို့ချပေးထားပါတယ်။

ဒီ program ကို လေ့လာတဲ့ အခါမှာ ကျွန်တော့ အနေနဲ့ Analysis အနေနဲ့ Line By Line ရှင်းပြပေးခြင်းမရှိပါဘူး။ ဘာကြောင့်လဲဆိုတော့ စာဖတ်သူရဲ့ အတွေ့အကြံကို ဆန်းစစ်ချင်လိုပဲ ဖြစ်ပါတယ်။ အကယ်၍များ စာဖတ်သူအနေနဲ့ နားမလည်မရှင်းလင်းတာမှားပို့ချပေးမြို့ချင်ပါတယ်။

Table 2.9 The Different levels of Access Control

Visibility	public	protected	default	private
From the same class	yes	yes	yes	yes
From any class in the same package	yes	yes	yes	no
From any class outside the package	yes	no	no	no
From a subclass in the same package	yes	yes	yes	no
From a subclass outside the same package	yes	yes	no	no

Listing 2.18 အနေနဲ့ ပို့ချမေးတဲ့ program ငယ်လေးကို စနစ်တကျရေးလာပါလို့ပဲ အကြပ်ချင်ပါတယ်။ Program ရဲ့ flow ကို Line By Line Method နဲ့ လေးလာမယ်ဆိုရင် စာဖတ်သူ အတွက် အလွန်အကျိုးရှိနိုင်ပါလိမ့်ယယ်။ ဒီတော့ listing 2.18 ကို မလေးလာခင်မှာ Table 2.9 အနေနဲ့ ရေးသားထားတဲ့ The Different levels of Access control ဆိုတဲ့ Table ကို အရင် လေးလာလိုက်ပါတီး။ ရှေ့မှာ သင်ကြားပေးခဲ့တဲ့ Table တွေနဲ့ သဘောတရားတွေ တည်ပါတယ်ဆိုပေမဲ့ အဖက်ဖက်ကပြည့်စုံ အောင် လေးလာကြည့်စေချင်လိုပါပဲ။

Listing 2.18 Creating Same Name, Different Arguments with JAVA

```

1: import java. awt. point;
2:
3: class MyRect {
4:         int x1 = 0;
5:         int y1 = 0;
6:         int x2 = 0;
7:         int y2 = 0;
8:
9: MyRect buildRect (int x1, int y1,int x2, int y2) {
10: this .x1 = x1;
11: this .y1 = y1;
12: this .x2=x2;
13: this .y2 = y2;
14: return this ;
15: }
16:
17: MyRect buildRect (Point topleft, Point bottom-
Right) {
18: x1 =topleft .x;
19: y1 =topleft.y;
20: x2 =bottomRight .x ;
21: y2 =bottomRight .y ;
22: return this ;
23: }
24:
25: MyRect buildRect (Point topleft, int w, int h){
26: x1 =topleft. x;
27: y1 =topleft.y;
28: x2 =(x1 +w);
29: y2 =(y1+w);
30: return this;
31: }
32:
33: void printRect () {
34: System.out. print ("MyRect:<" +x1 + " , " +y1);

```

```

35: System.out. print In (" , " + x2 + " , " + y2 + "> ");
36: }
37:
38: public static void main (String [ ] arguments) {
39: MyRect rect =new MyRect ();
40:
41: System. out.print In ("Calling buildRect with
coordinates 25, 25, 50, 50:");
42: rect.buildRect (25,25, 50, 50);
43: rect.printRect ();
44: System.out,Print In ( " * * *");
45:
46: System.out. print In ("Calling buildRect with points
(10, 10), (20, 20): ");
47: rect.buildRect (new Point (0, 10) ,new Point (20,20));
48: rect. printRect ();
49: System.out.print In (" ***");
50:
51: System. out. print ("Calling buildRect with 1 point
(10, 10),");
52: System.out.print In ("width (50) and height (50): ");
53:
54: rect. buildRect (new Point (10, 10), 50, 50);
55: rect.printRect ( );
56: System.out. print In ("***");
57: }
58: }
```

The following is this program's output:

Calling buildRect with coordinates 25, 25, 50,50:  
MyRect:<25, 25, 50, 50>

\* \* \*

Calling buildRect with 1 point (10,10),(20, 20):  
MyRect<10,10,20,20>

\* \* \*

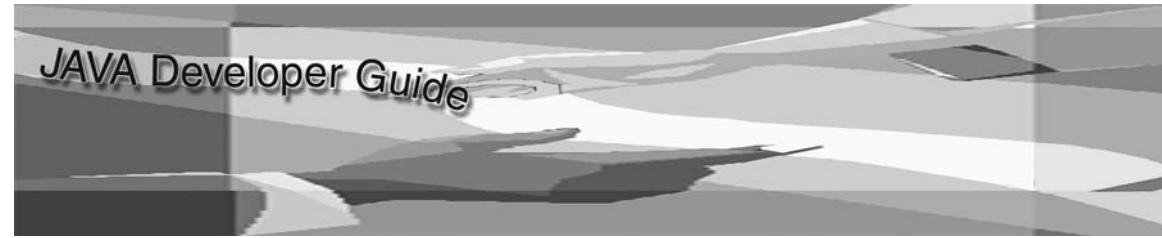
Calling buildRect with 1 point (10,10) width (50)
and height (50):

MyRent:<10,10,60,60>

ဒီprogram ငယ်လေးကို ကိုယ်တိုင်လေးလာသုံးသပ်ကြည့် ပြီးရင်တော့ ရှေ့လာမယ့် သင်ခန်းစာတွေမှာ JAVA ရဲ့ features တွေနဲ့ ဘယ်လို တည်ဆောက်အားးပြုရမယ်၊ ဘယ်နည်းနဲ့ ထင်မာ ဖြည့်စွက်နိုင်တယ် စတဲ့အချက်အလက်တွေကို ရှေ့သင်ခန်းစာမှာ လေးလာကြည့်ကြပါစို့။

ကျော်ဇူးတော်လေး (ကသာ)





## 2.5 INNER CLASSES, PACKAGES AND EXTENDING THE INTERFACE

ရှေ့က သင်ခန်းစာမှာ ကျွန်တော် သင်ကြားပို့ချော့တဲ့ သင်ခန်းစာတွေကို လေ့လာ သင်ယူတဲ့အခါမှာ Listing တစ်ချင်းစိကို သေသေချာချာ ဖော်ရှုပြီးတော့ လေ့လာစေချင်ပါတယ်။ ဘယ် Language မှာမဆို အဆင့်မြင့် နက်နဲ့လာလေလေ Theory & Logic အပိုင်းတွေမှာလဲ အဆင့်မြင့် နက်နဲ့လာမယ်။ ပြီးတော့ complex ဆန်လာမယ်။ ဒါဟာ ချိန်ခွင်တစ်ခုလို့၊ ညီမျှခြင်းတစ်ခုလို့ သဘောတရားပါပဲ။

ဘယ်နေရာ၊ ဘယ် Language မှာမဆို အားသာချက်၊ အားနည်းချက်တွေဟာ ရှိနေတာပါပဲ။ ဒီ အားသာချက်၊ အားနည်းချက်တွေကို ဘယ်လို့ ဖြင့်ဆင်ဖြည့်စွက်ပြီးတော့ အသုံးပြုရမယ် ဆိုတာကိုပဲ ကျွန်တော်တို့ စဉ်းစားရမှာပါ။ ဥပမာအနေနဲ့ ပြောရရင် JAVA Language ရဲ့ အားသာချက်၊ အားနည်းချက်တွေကို ကျွန်တော်တို့အနေနဲ့ လက်ခံရမှာပါပဲ။ အားနည်းချက်တွေကိုသာ လိုက်ရှုပြီး အပြစ်ပြောနေရင်တော့ နိုင်ငံတကာမှာ အကောင်းဆုံးလို့ လက်ခံထားတဲ့ language တစ်ခုဟာ ကျွန်တော်တို့ မြန်မာ လူမျိုးရဲ့အမြင်မှာ ကောင်းကွက်ဆိုတာတောင် ရှိနိုင်တော့မှာ မဟုတ်တော့ပါဘူး။ ဒီအချက်ဟာ ဘာကိုမျှ အကောင်းမြှင့်တတ်တဲ့၊ မမြင်နိုင်တဲ့ လူတန်းစားအချို့ကို ရည်ရွယ်တာပါပဲ။ အပြစ်ကိုသာ ရှုပြီး အဆိုးသာက်ကနေ ပြောနေမယ်ဆိုရင်တော့ Designer တွေ ကိုယ်တိုင် အများရဲ့အမြင်က မလွှတ်ခဲ့ပါဘူး။

အရှေ့က သင်ခန်းစာတွေမှာဆိုရင် Interface တွေကို extending လုပ်တဲ့ သဘောတရားကို အကျယ်တဝ် ရှင်းလင်းတင် ပြပေးပြီးပါပြီ။ အခု သင်ခန်းစာကတော့ Inner Classes & Packages တွေအကြောင်းကို အမြေခံကစားပြီးတော့ လေ့လာကြရအောင်လား။ အဲဒီကမှ နောက်ဆက်တဲ့အဖြစ် Extending Interfaces နဲ့ ပတ်သက်တာတွေကို ဆက်လက်လေ့လာကြတာပေါ့။ က ဒီတော့ Inner Classes ဆိုတဲ့အပိုင်းကစားပြီးတော့ လေ့လာလိုက်ကြရအောင်လား။

### ◆Inner .Classes In JAVA

Inner Class ဆိုတာကတော့ JAVA ရဲ့ တည်ဆောက်ထားတဲ့ ပင်မ Primary Class အတွင်းမှာဖြစ်စေ ဒါမှုမဟုတ် အခြား

တည်ဆောက်ထားတဲ့ Class ရဲ့အတွင်းမှာ ထပ်မံပြီး ဖြည့်စွက်တည်ဆောက်ထားတဲ့ Class မျိုးကို ဆိုလိုတာပါပဲ။ ဒီအခါမှာ Inner Class type တွေဟာ သဘောတရားအရနဲ့ ဖွဲ့စည်းတည်ဆောက်ထားအရ ကွဲပြားခြားနားမှုတွေ ရှိလာပါတယ်။ ဒါကြောင့် Inner Class type တွေ ကွဲပြားခြားနားတာကို သဘောပေါက်နားလည်အောင် Table 2.10 နဲ့ ရေးသားပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်လိုက်ပါ။

Static Member Class	Use allowed in all type of class
Member Classes	Defined top level class and an instance is always associated with a particular instance of the top level class.
Local Classes	Defined with a block of code.
Anonymous Classes	Classes has no name and the class definition is combined with instantiation of a object of the class.

Table 2.10 Types of Inner.Classes In JAVA

က – အပေါ်မှာ ရေးသားခဲ့တဲ့ Table 2.10 ဟာ Inner Classes တွေရဲ့ types တွေကို ပုံစံပြီးရေးစွဲရင်းလင်းပြထားတာပဲ ဖြစ်ပါတယ်။ ဒီတော့ အခြေခံအရ အပ်စွဲလိုက်ရင် JAVA ရဲ့ Inner Classes type မှာ အပ်စွဲလေးခုကွဲပြားပါတယ်။ က ဒီတော့ အပ်စွဲတစ်ချင်းစီ <Inner Classes type ပုံစံတစ်ခုစီ > ကနေ စာစ်တကျလေ့လာကြရအောင်လား။ Listing 2.19 အနေနဲ့ ရေးသားထားတဲ့ Program c ယောက်လေးကနေ စတင်ပြီးတော့ လေ့လာကြည့်ကြပါစို့။

Listing 2.19 Static Member Class In JAVA

```
1. public class Member {
2.     public static void main (String args []
3.     {
4.         Static Class SC=new static class( );
5.         SC. Print Hello ( ) ;
6.     }
7.     static class Static Class {
```

```

6. void print Hello ( )
7. {System.out.printIn ("Hello! JAVA world
     is Amazement!);}
8. }
9. }
Output
-----
Hello! JAVA World is Amazement!
-----

```

#### Analysis

၁။ ဒါ program ငယ်လေးမှာ ရေးသားထားတဲ့ static class type ပုံစံ Inner .Class အမျိုးအစားဟာ ကျွန်တော်တို့ အနေနဲ့ ပြီးခဲ့တဲ့ အရှေ့သင်ခန်းစာတွေမှာ program ပေါင်းများစွာ လက်တွေ ရေးသားပြီးခဲ့တဲ့ type ပုံစံပါပဲ။ ဒါကြောင့် static class ဆိုတာဟာ ကျွန်တော်တို့အနေနဲ့ familyဖြစ်ပြီးသား type တစ်ခုဆိုရင်လည်း မမှားပါဘူး။ အထွေအထူး သိပ်ရှင်းပြစ်ရာ မလိုတော့ဘူးပေါ့။ လိုင်းနံပါတ် (၁) မှာ Member ဆိုတဲ့ primary class တစ်ခုတို့ ဖုန်တီး ရေးသားပါတယ်။ အဲဒီ class ထဲမှာမ Static Class ဆိုတဲ့ Class Name နဲ့ Inner Class တစ်ခုကို ဖုန်တီး ရေးသားယူပါတယ်။

၂။ အဲဒီ Inner Class ဟာ လိုင်းနံပါတ် (၅) ကနေ လိုင်းနံပါတ် (၈) အထိ ရေးသားထားပါတယ်။ အဲဒီ Inner Class ဟာ အထွေအထူး ရေးသားထားခြင်း မရှိပါဘူး။ အဲဒီ function/Method ဖြစ်ပဲ့ print Hello () ဆိုတဲ့ function ကို လုမ်းခေါ်ယူလိုက်ရင် output အနေနဲ့ Hello ! ထို့ ထွက်လာအောင် Method ကို ရေးသားထားခြင်းပဲ ဖြစ်ပါတယ်။ ဒါဟာ Inner Class ထဲမှာ ပါဝင်တဲ့ function () ကို ရှုင်းပြတာနေရာ။ လေ့လာကြည့်လိုက်ပါဉိုး။

၃။ လိုင်းနံပါတ် (၂) မှာတော့ main () function ကို ရေးဖွဲ့ထားပါတယ်။ အဲဒီ main () function ရယ်၊ static class ရယ်ဟာ Member ဆိုတဲ့ primary class တစ်ခုအတွင်းမှာပဲ ရေးသားထားတာနေရာ။ ဒါဟာ Inner Class ရဲ့ သဘောတရားပါပဲ။ လိုင်းနံပါတ် (၃) မှာ Inner Class အဖြစ် ဖန်တီးယူခဲ့တဲ့ Static Class ကို new keyword ကို အသုံးပြုပြီးတော့ SC ဆိုတဲ့ Name နဲ့ ဖန်တီးရယူပါတယ်။ Object- Name အနေနဲ့ ဖန်တီးရယူတာပါပဲ။ ဒါတွေဟာ ကျွန်တော် ရှေ့က ပို့ချုခဲ့တဲ့ သင်ခန်းစာကို သင်ယူခဲ့သူအဖို့ လုပ်ရှိုးလုပ်စဉ်တစ်ခုသွေ့ ရုံးရှင်းနေမှာပါပဲ။ လိုင်းနံပါတ် (၄) မှာမ SC. Print Hello (); ဆိုပြီး Static Class အနေနဲ့ ရေးဖွဲ့ထားတဲ့ Inner Class အတွင်းက function () ကို လုမ်းပြီး ခေါ်ယူလိုက်တာပါပဲ။ အဲဒီ function ထဲမှာတော့ System.out. Print In("Hello! JAVA World

is Amazement!"); လို့ ရေးသားထားတဲ့အတွက် output အနေနဲ့ Hello! ထို့ ထွက်လာတာပါပဲ။ ဒါဟာ program code အနေနဲ့ လေ့လာစရာ program type မဟုတ်ပါဘူး။ Static Class တစ်ခုကို Inner Class တစ်ခုအနေနဲ့ ဘယ်လိုဖန်တီးရေးသားရ ဘယ်ဆိုတာကိုပဲ ရှင်းပြတာပါပဲ။ ဒါဆိုရင်တော့ Inner Class type ထဲက Static Class ရေးဖွဲ့တဲ့ type ကို နားလည်သဘောပေါက်သွားမှာပါ။ ကဲ-ဒီတော့ ဆက်လက်ပြီးတော့ Member Class ရေးဖွဲ့တဲ့ သဘောတရားကို ဆက်လက်လေ့လာကြည့်ရအောင်လား။

#### ◆ Type of Member Class

Member Class တွေကို ရေးဖွဲ့တဲ့အခါမာတော့ programmer အနေနဲ့ primary class ရဲ့ ပြင်ပမှာပဲ သီးခြား type နဲ့ ရေးဖွဲ့လို့ ရပါတယ်။ Inner Class လို့ခေါ်ပေမဲ့ သီးခြား class တစ်ခုအတွင်းမှာ member type အနေနဲ့ ရေ့ဖွဲ့တာကြည့် Member Class လို့ ခေါ်ခေါ်တာပါပဲ။ ဒါပေမဲ့ ဒါ member class ဟာလည်း အရှေ့က static class ပုံစံလိုပါပဲ။ လုပ်ရိုးလုပ်စဉ်အတိုင်း ရေးဖွဲ့နေကျုံး class ပုံစံတွေပါပဲ။ ဒါအတွက် သီးခြား အထူးတလည်း လေ့လာစရာမလိုပါဘူး။ အရှေ့မှာ သင်ကြား ပို့ချုပ်ပေးထားပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

Lisitng 2.20 Member Class type of Inner Class in JAVA

```

1. public class Member-Class{
2. public static void main (String args
   [])
3. square sq = new square (3.0,4.0);
4. sq.show volume ();
5. }
6. }
7. class square {
8. double width ,height;
9. public Square (double w, double h)
10. { width = w; height = h;}
11. public void show Volume ()
12. {Box b = new Box (2.0);
13. System.out.printIn ("Volume is" + b.get
   volume ());
14. }
15. class Box {
16. double depth;
17. public Box (double d)
18. { depth = d;}
19. public double getVolume ()
20. {return width *height* depth;}
21. }
22. }
output
-----
```

Volume is 24

---

Note : ဒီ Listing 2.20၏ program နဲ့ ပတ်သက်လိုကတော့  
 program flow အပိုင်းကိုပဲ အမိက ရှင်းလင်း တင်ပြမှာ  
 ဖြစ်ပါတယ်။ Member Class တစ်ခုကို ဘယ်လို ရေးဖွဲ့  
 သွားတယ်ဆိုတာကိုပဲ အမိက ရေးသားပြတ်းတာပါပဲ။  
 ဒီ Class နဲ့ ပတ်သက်တဲ့ ပုံစွဲ program ကို ရှုံးက  
 သင်ခန်းစာတွေမှာ ပို့ချုပြုးသွားပါပြီ။ ပြန်လှန် လေ့လာ  
 ကြည့်သွေ့ပါတယ်။

## Analysis

- ၃။ ဒီ program ကယ်လေးမှာပါဝင်တဲ့ Member Class type <Inner Class ဖုန်းရေးဖွဲ့တဲ့> ပုံစံဟာ တွေ့မြင်နေကျေ class တွေကို ရေးဖွဲ့တဲ့ ပုံစံပါပဲ။ သင်ခန်းစာအရာသာ အခန်းကလူ အလိုက် ရေးသားပို့ဆေးရခြင်းပဲ ဖြစ်ပါတယ်။ လိုင်းနံပါတ် (၁) မှာ main () function ပါဝင်တဲ့ primary class ကို Member-class လို့ အမည်ပေးပြီးတော့ ရေးသားထားပါတယ်။

၂။ လိုင်းနံပါတ် (၇) မှာတော့ Square လို့ခေါ်တဲ့ Class တစ်ခု ကို ရေးဖွဲ့ထားပါတယ်။ အဲဒီ Square ဆိုတဲ့ Class ဟာ main function ထဲမှာ object အဖြစ် လိုင်းနံပါတ် (၃)မှာ ဖန်တီးထားတဲ့ class ပါပဲ။ အဲဒီ Square ဆိုတဲ့ Class ထဲမှာဆိုရင် constructor function ကိုရေးဖွဲ့ထားပြီး အဲဒီ constructor function ထဲမှာ parameter arguments တွေ ပါဝင်တာကို သတိပြုရမှာပါ။ အဲဒီ Square Class ရဲ့ function () အဖြစ်ကတော့ public void show Volume () ဆိုတဲ့ function ကို ရေးဖွဲ့ထားတာကို တွေ့ရမှာပါ။ ဒါပေမဲ့ ထူးခြားတဲ့ အချက်ကတော့ အဲဒီ Show Volume () ဆိုတဲ့ function ထဲမှာ Class ကို object သဖြယ် ဖန်တီးထားပြန်တာကို တွေ့ရပါတယ်။

၃။ အဲဒီတော့ ကျွန်ုင်တော်တို့အနေနဲ့ Square ဆိုတဲ့ Class ကိုသာ main () function ထဲက လုမ်းပြီးခေါ်ယူအသုံးချက်ရင် အဲဒီ Square Class အတွင်းမှာ ထပ်မံ ရေးသား ဖြည့်စွက်ထားတဲ့ သူရဲ့ Member Class ကိုလည်း လုမ်းပြီးတော့ ခေါ်ယူလိုက်သလို ဖြစ်တော့မှာပေါ့။ ဒါအချက်ကို စာဖတ်သူ အနေနဲ့ သေသေချာချာကိုမှ စဉ်းစားတွေ့ခေါ် သုံးသပ်ရမှာ ဖြစ်ပါတယ်။ အဲဒီလို့ ပါဝင်တာဟာ Member Class ရဲ့ ရေးဖွဲ့ပုံ၊ အသုံးပြုပုံ သဘောတရားပဲ ဖြစ်ပါတယ်။ ဒါလို့ရင် Square ဆိုတဲ့ Class ရဲ့ အတွင်းမှာမ ထပ်မံပြီး Class တစ်ခုကို တည်ဆောက်တယ်။ အဲဒီ Class ကတော့ လိုင်းနံပါတ် (၁၅) က Box ဆိုတဲ့ Class ပါပဲ။ အဲဒီ Box ဆိုတဲ့ class ကိုပဲ object -Name b အနေဖော်ဖန်တီးပြီးမှ အဲဒီ

Square Class ထဲမှာပဲ တစ်ဖန် ပြန်လည်ပြီးတော့ အသုံးချ ထားပါတယ်။

#### ◆ Local Class In JAVA

အရှင်မှာ လေ့လာခဲ့ရတဲ့ Static Member Class နဲ့ Member Class တို့အပြီးမှာ ဆက်လက်လေ့လာရမှာကတော့ local class ပဲ ဖြစ်ပါတယ်။ Local Class ရဲ့ type ဟာ အရှင် Static Member Class & Member Class တို့နဲ့ ပုံစံဆင်တဲ့ ပါတယ်။ ဒါကြောင့် local class နဲ့ ပတ်သက်တဲ့ Listing 2.21 ကို လေ့လာကြည့်ပါ။

## Listing 2.21 Demonstrate a Local Class In JAVA

```
1. public class Demolocal {
2. public static void mian(String args[])
{
3. class local Class {
4. void print Hello ( )
5. {System.Out.Println ("Hello! JAVA World
is Amazement!");
6. }
7. LocalClass Lc = new Local Class ( ) ;
8. LC.print Hello ( );
9. }
10. }
output
-----
Hello! JAVA Word is Amazement!
```

ဒီ Local Class ဟာ အရေးမှာ သင်ယူခဲ့တဲ့ Static Member Class & Member Class တို့၏ class type တွေပါပဲ။ ဒီ Class type တွေဟာ object ရဲ့ concept တွေကို ဖန်တီးရေး သားတဲ့အခါနာ အထောက်အကူပြနိုင်တဲ့ ပုံစံတွေပါပဲ။ Analysis လုပ်ပြီး မရှုံးပြတော့တာကတော့ လွယ်ကူရှိရင်တဲ့ Sample type ဖြစ်နေလိုပါပဲ။ ကဲ-ဒီတော့ ဆက်လက်ပြီးတော့ ရှေ့မှာ ပို့ချခဲ့တဲ့ Table 2.10 မှာ ပါဝင်တဲ့ Type of Inner Class တွေ ထဲက နောက်ဆုံး အမျိုးအစားဖြစ်တဲ့ Anonymous Class တွေအကြောင်းကို လေ့လာသင်ယူကြတာပေါ့။

#### ◆ Anonymous Class In JAVA

အခါ သင်ကြားပို့ချမယ့် Anonymous Class type ဟာ JAVA ရဲ့ Inner Class type တွေထဲပါဝင်တဲ့ Method ပဲနော်။ ဒါပေမဲ့ အရောက သင်ယူခဲ့ကြရတဲ့ class တွေနဲ့ မတုတာကတော့ JAVA ရဲ့ package တွေထဲမှာပါဝင်တဲ့ ဘယ်လို့ class ဖို့ပဲဖြစ်ပါစေ။ ဥပမာပြောရရင် အရောက ဖန်တီးပြီးသား object & class တွေကနေ လိုချင်တဲ့ မို့ပြင်းချင်တဲ့ class & object ကို မို့ပြင်းခွင့်ရှိတဲ့ Super-Class to sub -class ပုံစံပြနိုင်တဲ့ Class type ဟာ Anonymous Class type ပဲဖြစ်ပါတယ်။ ဒီအခါမှာ

ရှိပြီးသား၊ ဖန်တီးပြီးသား Class တွေကို Obj အဖြစ် ဖန်တီးအသုံးချင့်သာမကပဲ package တွေထပါဝင်တဲ့ object & class တွေကိုပါ လှမ်းပြီး Inner Class type ဖြစ်တဲ့ Anonymous Class သဖွယ် ဖန်တီးရယူနိုင်ပါတယ်။ တစ်ခုကိုတော့ စာဖတ်သူ အနေနဲ့ သတိပြုရပါမယ်။ အဲဒါဟာ ဘာလဲဆိတော့ အချက်ရှိလောင်းအပိုင်းဟာ JAVA ရဲ့ features ဖြစ်တဲ့ Inner Class ဆိတဲ့ အခန်းကဏ္ဍပဲ ဖြစ်ပါတယ်။ Inner Class type ပုံစံဟာ အဓိက ပုံစံအားဖြင့် (c) မျိုးကဲ့ပြားတယ်။ အဲဒီ တွဲပြားမြားတဲ့ Inner Class type တွေဟာ သီးခြားစီ ဖန်တီးနိုင်တဲ့ စွမ်းရည်တွေ ရှိကြတယ်။ Inner Class type တွေဟာ တစ်ခုနဲ့တစ်ခု ပုံစံဆင် တူးတယ်လို့ ဆိုပေမဲ့ လုံးလုံးလျားလျား ထပ်တူထပ်မျှ မျှဆင်တူ တာမျိုး မဟုတ်ပါဘူး။ ဒီအချက်တွေကိုတော့ စာဖတ်သူရဲ့ လေးလေ နေမှုမှာ အဓိကအချက်အနေနဲ့ထားပြီးတော့ မှတ်သားရပါမယ်။

Note : C&C+++နယ်ယက် programmer တွေအနိုက်လည်း  
Inner Class types တွေကို တွေ့ရှိ လေးလေမှာ ပါတယ်။  
ဒါပေမဲ့ ဒီ Inner Class types တွေကို C&C+++  
features တွေမှာ သိပ်ပြီးတော့ ထင်ပေါ်ခြင်း မရှိပါဘူး။  
မြုပ်ကွက်သဖွယ် ရေးသားကြတဲ့အတွက်ကြောင့်ပဲ ဖြစ်ပါ  
တယ်။ ဒါပေမဲ့ visual C++ & visual J+++ features  
တွေမှာပါ Inner Class types အကြောင်းကို  
JAVA Language မှာလိမ့်း အကျယ်ချုပ်ရှင်းပြ အသုံးချ  
တာမျိုးကို တွေ့ရှိပါတယ်။ ကဲ Inner Class types  
တွေထဲမှာ ပါဝင်တဲ့ Anonymous Class အကြောင်းကို  
listing 2.22 အနေနဲ့ ဆက်လက် လေးလေသင်ယူကြတာ  
ပေါ်။

#### Listing 2.22 Using Anonymous Class types In JAVA

```

1. public Class Anonymous Class {
2. public static void main (String arguments [ ] ) {
3. object obj = new object ( );
4. /* Local Class which extends object
and
5. Overrides its to String ( ) methods to
return
6. the string "Hello! JAVA World is Amaze-
ment!"
7. */
8. {
9. public string to String ( )
10.{ return "Hello! JAVA world is Amaze-
ment! ";
}
11. };
12. System .Out. printIn (obj);
13. }
14. }
output
-----
```

Hello! JAVA world is Amazement!

Note : ObjectM class is the super class of every other class & package In JAVA. Reference of Methods in API (Application Programming Interface), Index 1.5 Arrays of JAVA, In Chapter [1] Analysis

c) ဒီ listing 2.22 မှာ ရေးသားထားတဲ့ program ဟာ Anonymous Class type အကြောင်းကို ရှင်းပြပြီး ရေးဖွဲ့ထားတဲ့ program ငယ်လေးပါဝဲ။ အရှေ့က listing တွေမှာလိုပါပဲ။ Output ဟာ Hello! JAVA world is Amazement! ဆိတဲ့ output လေးပဲ ထွက်လာမှုပါပဲနော်။ ဒါပေမဲ့ Anonymous class type တွေကို ရေးဖွဲ့တဲ့ သဘောတရားနဲ့ ဖန်တီးပြီး object & class တွေ၊ extends လုပ်ယူတဲ့ သဘောတရားတွေဟာ အခါ program ငယ်ဖြစ်တဲ့ listing 2.22က program ငယ်လေးမှာ ပါဝင် ပါတယ်။ လိုင်းနံပါတ် (c) မှာ ဖန်တီးလိုက်တဲ့ obj ဟာ ဘယ်ကနေ extends လုပ်တာလဲ။ ဘယ် class ကို override လုပ်ထားတာလဲဆိတာ စာဖတ်သူ စဉ်းစားသုံးသပ်ကြည့် နို့ လိုလာပါပြီ။ ဒီအဲမှာ ဒီ object ကို new obj; အဖြစ် ဖန်တီးတာဟာ main ( ) function ထဲမှာ ဖန်တီးထားပါတယ်။

j) အဲဒီ main ( ) function ထဲမှာ လိုင်းနံပါတ် (c) ကနေ လိုင်းနံပါတ် (j) အထိ ရေးသားထားတာကတော့ comments တွေပါပဲ။ အဲဒီ comments တွေကိုသာ လေးလေ ကြည့်မယ်ဆိုရင် Anonymous class type ရဲ့ သဘော တရားကို အကြမ်းဖျင့်အားဖြင့်တော့ သဘောပေါ်ကြသွားမှာပါ။ ဒီ class type ရဲ့ အသွင်သွောန်က ပြင်ပက object & class တွေကို မိမ်းလိုက်တယ် ဆိုတော့ package တွေ ထဲမှာ ပါဝင်တဲ့ Method တွေလဲ အပါအဝင်ပဲ ဖြစ်ပါတယ်။ ဒါကို နားလည်သဘောပေါ်အောင် သေသေချာချာ လေးလေကြည့်ပါ။

ဒီ program ငယ်လေးဟာ Anonymous class type ကို အခြေခံအနေနဲ့ ရှင်းပြတာဖြစ်တဲ့အတွက် Line-By - Line အနေနဲ့ ရှင်းပြမနေတော့ပါဘူး။ စာဖတ်သူကိုယ်တိုင်ပဲ Trace လုပ် လေးလေကြည့်ပါ။ ကျွန်ုတ်အနေနဲ့ဆက်လက်ပြီး Concepts တွေ အကြောင်းကို ဆက်လက် မပို့ချေခင်မှာ Sub-tittle အနေနဲ့ JAVA ရဲ့ Object & Class features တွေအကြောင်းကို ကြားဖြတ် အနေနဲ့ ပို့ချေပေးထားပါတယ်။ လေးလေလိုက်ပါပြီး။ ဒီ features တွေထဲမှာ JAVA ရဲ့ own packages တွေ ဖန်တီးနည်းလည်း အပါအဝင် ဖြစ်ပါတယ်။

KYAW ZAYAR LAY [ IT ENGINEERING ]

## JAVA DEVELOPER GUIDE

### 2.5.1 CREATE YOUR OWN'S PACKAGE IN JAVA

အရို့ချမယ့် သင်ခန်းစာဟာ JAVA Programming Language ကို လေ့လာ သင်ယူနေကြတဲ့ Programmer/ Developer တွေအတို့ အလွန်အရေးပါတဲ့ သင်ခန်းစာဖြစ်တယ်ဆိုရင် မမှားပါဘူး။ ဘာကြောင့်လဲဆိုတော့ ကျွန်ုတ် ရှုံးမှာလေ့လာခဲ့တဲ့ Java Language မှာ ကိုယ်ပိုင်ဖန်တီး ရေးသားတည်ဆောက်နိုင်တဲ့ API method / Package file တွေအကြောင်းကို အခု သင်ခန်းစာ မှာ စတင် လေ့လာကြရမှာပါ။ ဒီတော့ Package ဆိုတာဘာလဲ? ဒါကို အရင်လေ့လာကြရအောင်။

Packages Statement is put at the first statement in a JAVA Source File.

Any Classes contained in the file will be associated with the package , Package Name.

ဒါဟာ package file ကို ရည်ညွှန်းတဲ့ အချက်အလက်တွေ ပါပဲ။ ဒါပေမဲ့ ဒီအချက်တွေဟာ package file တွေရဲ့ အမိုးယ် ကို အကုန်အစင် ပြည့်စုံအောင် ဖော်ပြုမပေးနိုင်ပါဘူး။ ကျွန်ုတ် တို့ ရှုံးသင်ခန်းစာတွေမှာ API (Application Programming Interface) method တွေကို ခေါ်ယူ အသုံးချခဲ့တာကို စာဖတ်သူ မှတ်မိမှာပါ။ ဥပမာဏေးရရှင်တော့-

Math.Square (double x);

ဆိုပြီး ခေါ်ယူ အသုံးချရင် Math Class ထဲက Square ( ) method ကို ခေါ်ယူ အသုံးချတာကို စုတ်မိမှာပါ။ ဒါဟာ API method ခေါ်ယူအသုံးချတာပါ။ ဒီအခါ objectအဖြစ် ရည်ညွှန်း ခြင်းခံရတဲ့ Maths Class ဟာ java.util package ထဲမှာ ပါဝင် ပါတယ်။

ဒီတော့ package ဆိုတာဟာ class တစ်ခု၊ တစ်ခုထက်မက ပါဝင်နိုင်တဲ့ object (or) Data & Method ( ) container အဖော်အစီအစဉ် ပါဝင်နိုင်တဲ့ Method ( ) တွေကို ဆွဲထုတ် အသုံးချတာပါပဲ။ ဒါဟာ အရှေ့က သင်ခန်းစာတွေမှာ ကျွန်ုတ်တို့ လက်တွေ လေ့လာ အသုံးချခဲ့ပြီးပါပြီ။

အဲဒီတိုန်းက ကျွန်ုတ်ပြောခဲ့တဲ့စကား ရှိပါတယ်။ JAVA က package file ကို တည်ဆောက်ပြီးတော့ API method တွေကို ဆွဲထုတ်အသုံးချတာဟာ C & C + + Language Header file တွေရဲ့ သဘောတရားနဲ့ အတူတူပါပဲလို့ ပြောလို့ရပါတယ်။

ဒီအခါ C & C + + Language မှာလည်း ကိုယ်ပိုင် Header file တွေကို ဖန်တီး တည်ဆောက်ပြီးရင်လည်း ပြန်လည်ခေါ်ယူ အသုံးချလို့ ရပါတယ်။ ဒီ C & C + + Language ၊ Own Header file ရဲ့ သဘောတရားနဲ့ JAVA Language ၊ Own package file တွေရဲ့ သဘောတရားဟာ C & C + +နယ်ပယ်က Programmer တွေအဖွဲ့ကတော့ JAVA Language ကို ပြောင်းလဲ လေ့လာတဲ့အခါမှာ အထောက်အကြု အများကြီး ရဖော်နိုင်မှာပါ။ ဒါကို C & C + +နယ်ပယ်က programmer တွေ လေ့လာခဲ့တဲ့ Sample Type နဲ့ အကြမ်းဖျင်း လေ့လာကြည့်ရအောင်။ Sample

- // Create cal.hfile: Base Class header file in C ++
 programming language
- class calculate {
- protected:
- double length, width, area;
- public:
- void rectangle (double L, double w)
- {Length = L width = w;}
- 
- void calArea ( )
- { return area = length \* width; }
- }
- // this class is can be overridden in another derived class

အကယ်၍ ဒီ class ကို save လုပ်တဲ့အခါမှာ Calculate.h အမည်နဲ့ သိမ်းဆည်းပြီးတော့ save လုပ်တယ် ဆိုကြပါစို့။ ဒီ Own Header file ကို အသုံးချပြုတော့ program တည်ဆောက် အသုံးပြုမယ်ဆိုရင်တော့ Header file တစ်ခုအနေနဲ့ ခေါ်ယူ အသုံးပြုလို့ ရပြီပေါ့။ ဒါပေမဲ့ C + + Language က built -in (build-on) အနေနဲ့ ပါလာတဲ့ Header file တွေကို ခေါ်ယူတဲ့အခါမှာတော့ အောက်ပါအတိုင်း ခေါ်ယူရပါတယ်။

```
# include <iostream.h>
# include <stdio.h>
# include <math.h>
# include <conio.h>
# include <iomanip.h>
```

ఆటగ్రీప్రెస్ లోయ్ అవ్యాపకంగా గ్లోబల్ టెండర్ కి. ఇది తన్న ఐఓఎఫ్ ఫైల్ కి. Own Header file కి లోయ్ అవ్యాపకంగా వ్యాపించి ఉన్న విభజనాలలో ఒకి. అందులో మొత్తం వ్యాపకంగా వ్యాపించి ఉన్న విభజనాలలో ఒకి.

```
# include "Calculate.h"
```

ဒါဟာ own header file တွေကို နောက်ပြီးရင် ပြန်လည်  
ခေါ်ယူ အသုံးချဖို့နည်းလမ်းပါပဲ။ ဒီနည်းလမ်းတွေကိုလာက်တွေ  
လေ့လာခဲ့ကြရတဲ့ C & C++ နယ်ပယ်က Programmer တွေ  
အပိုကတော့ အခု JAVA Language နဲ့ Own Package တွေ  
တည်ဆောက် အသုံးချဖို့နည်းလမ်းဟာ အဆင်ပြ လွယ်ကူနေမှာ  
ပါပဲ။

Note: C & C ++ Language မှာ Header file တွေ ဖန်တီး  
ပြီးရင် System's CLASSPATH ရဲ့ sub-directory  
ဘယ် folder ထဲမှာမဆို saveလုပ်ခဲ့လို ရပါတယ်။  
C + + ရဲ့ Header file / Project .file တွေဟာ ဘယ်  
နေရာမှာ saveလုပ်ခဲ့လုပ်ခဲ့ compileလုပ်နိုင်ပါတယ်။  
ဒီအခါက်တိ သတိပါထားပါ။

ကဲ- ဒါနိုဂင်တော့ အခြေခံကစ္စပြီးတော့ Package file ငယ်လေးတွေ စတင် တည်ဆောက်ကုသွှေ့ကုရအောင်လား။

## Create a Package

Package တစ်ခုကို ဖန်တီးတည်ဆောက်တဲ့အခါမှာ မိမိရဲ့ System's CLASSPATH ရဲ့ သေချာအောင် ပြောရရှုတော့ C:\j2sdk 1.4 > ရဲ့အောက်မှာ sub-directory တစ်ခုကို ဖန်တီး တည်ဆောက်ရမှာပါ။ JAVA ရဲ့ Software Development kit ကို Installation လုပ်ထားတဲ့အပေါ် မူတည်ဖြိုးတော့ System's CLASSPATHဟာ အောက်ပါအတိုင်း ကုပြားနေဖိုင် ပါတယ်။ ဒါဟာ ပြဿနာမဟတ်ပါဘူး။ အတွက်ပါပဲ။

C:\java2>

C:\jdk1.4>

C:\j2sdk1.4.0>

C:\j2sdk 1.4.1-01>

အခိုက်တော့ C: / java\_language\_software ရှိတဲ့ System's CLASSPATH ရဲ အောက်မှာ new sub-directory တစ်ခုကို ပိဋက္ခိယ်တိုင် create လုပ်ရပါမယ်။ ဒါတော့ အသစ်ဖုန်တိုးမယ့် new sub -directory ကို calculate လိုအမည်းပေးပြီးတော့ တည်ဆောက်လိုက်ပါ။ ဒါဟာ ကျွန်ုင်တော်တို့ ဖုန်တိုးမယ့် Package Name ပါပဲ။

ဒီတော့ System's CLASSPATH ဟာ C:/j2sdk 1.4.1-01 / calculate >ဆိတဲ့ System's CLASSPATH အနေနဲ့ ရရှိလာမှာပါ။ အဲဒီ ရရှိလာမယ့် System's CLASSPATH မဲ့ sub -directory ထဲမှာ ကျန်တော်တို့၏ own package file ကို

### Listing 2.23 Create own package file

```
1. public class Cal {  
2.     double length, width, area;  
3.     public double calArea ( )  
4.     { return area = length * width; }  
5.  
6.     public void rectangle ( )  
7.     {Length = L; width = w; }  
8. }
```

ဒီ package file ဟာ ရှေ့က C& C+ + programmer တွေအတွက် ရည်ညွှန်းခဲ့တဲ့ Header file type အတိုင်းပါပဲ။  
ဒီတော့ ဒီ package file ကို ကျွန်ုတ်တော်တို့ တည်ဆောက်ခဲ့တဲ့ System's CLASSPATH ရဲ့ အောက်မှာရှိတဲ့ Calculate ဆိုတဲ့ Sub-directory ထဲမှာ သွားပြီးတော့ Save လုပ် သိမ်းဆည်းပါ မယ်။ Save လုပ်တဲ့အခါမှာတော့ File\_Name ကို "Calculate.java" လို့ အမော်ပေးပြီးတော့ save လုပ်ပါမယ်။ ဒါဆိုရင် ကျွန်ုတ်တို့အနေနဲ့ calculate လို့ အမည်ရှိတဲ့ Package file ကို ဖန်တီး တည်ဆောက်ပြီးသွားပြီးပေါ်နော်။

C:\j2sdk1.4.1-01\Calculate>

ခိုတဲ့ sub-directory ထဲမှာသွားပြီးတော့ program  
ကို ရေးလိုမရပါဘူး။

ဒီအချက်ဟာ အလွန်အရေးကြီးတဲ့ အချက်ပဲ ဖြစ်ပါတယ်။  
အကယ်၍ အခု ရေးသား တည် ဆောက်မယ့် program  
ကိုသာ -

C:\j2sdk 1.4.1-01\Calculate>  
შეიტა system's CLASSPATH အောက်ဖူးသာ ရေး

## ကွန်ပျူးတာသင်ခန်းစာ

သား တည်ဆောက်လိုက်မယ်ဆိုရင် program ကို Run လုပ်တာနဲ့ Error တက်လာမှာပါ။

ဒါဟာ JAVA programmer တွေအဖို့ အလွန် အရေးပါတဲ့ အချက်ပါပဲ။ ဒီလို package file တွေ ရေးဖွဲ့တဲ့ နေရာမှာ System's CLASSPATH ရဲ့ sub-directory တစ်ခုတည်းရဲ့ အောက်မှာသာ package \_file နဲ့ program \_file ကို ဖန်တီး တည်ဆောက်မယ်ဆိုရင် compile လုပ်ပဲအခါ မယ်တယ်။ မယ်ဆိုတဲ့ အချက်ကို စာဖတ်သူရဲ့ စိတ်ထဲမှာ စွဲနေအောင် မှတ်ထားပါ။

Note: C & C++ နယ်ပယ်က programmer တွေအဖို့ ဒီအချက် ကို သတိပြုစိမှာ မပုံတိပါဘူး။ C: \ BC5 > System's CLASSPATH ထဲမှာပဲဖြစ်ဖြစ်။ C: \ BC5\BIN > System's CLASSPATH ထဲမှာပဲဖြစ်ဖြစ်။ ဒါမှမဟုတ် C: \> ရဲ့ အောက်မှာရှိတဲ့ ဘယ် System's CLASSPATH အောက်မှာပဲ ဖြစ်စေ အဆင် ပြေတဲ့ ဘယ် System's CLASSPATH အောက်မှာမထိုး Save လုပ်ပြီး ခေါ်ယူ အသုံးချိန်ပါတယ်။ C++ + ရဲ့ compiler က ဒီအချက်ကို အလွယ်တကူ ဖြေရှင်းပေးနိုင်ပါတယ်။ နောက် အလွန် အရေးပြုးတဲ့ အချက်က C & C++ project တွေကို ဖန်တီးတဲ့ အခါမှာ C: \> System's CLASSPATH ရဲ့ အောက်မှာ sub-directory တစ်ခုကို ဖန်တီးပြီးတော့ အဲဒီ sub-folder ထဲမှာပဲ project အဖြစ် ဖန်တီးမထိုး Header file တွေ၊ program file တွေ၊ icon file တွေကို Create လုပ်ရပါတယ်။ ဥပမာဏပြာရရင် တော့ - C:\B C5\BIN > ရဲ့ အောက်မှာ Calculate ဆိုတဲ့ folder တစ်ခုကို ဖန်တီးလိုက်ပါမယ်။ ဒီတော့ C: \BC5\BIN\Calculate > ဆိုတဲ့ System's CLASSPATH ကို ရရှိလာမှာပါ။ အဲဒီ System's CLASSPATH ကို ရရှိလာတဲ့ အခါမှာ ဖန်တီးရေးဖွဲ့တဲ့ Header file တွေ၊ program file တွေ၊ GUI ဖန်တီးမှတွေ Cursor-movement & icon တွေကို အဲဒီ system's CLASSPATH ထဲမှာပဲ (တစ်နည်းပြုးတော်းပြောရရင် တော့ C: \B C5\BIN\Calculate > ထဲမှာပဲ) လာပြီး save လုပ် သိမ်းဆည်းပါပါတယ်။ Program file ကို ဖန်တီးပြီးလို့ Run & compile လုပ်တဲ့ အခါမှာ Header file နဲ့ program file တွေ System's CLASSPATH တစ်ခုတည်းအောက်မှာ ရှိနေပေမဲ့ Error တက်မလာပါဘူး။

ဒါကြောင့် C & C++ နယ်ပယ်ကလာတဲ့ programmer တွေအဖို့တော့ ဒီ own package တွေ တည်ဆောက်တဲ့ အခါမှာ

အထူးသတိပြုပြီးတော့ ရေးဖွဲ့ကြစေချင်ပါတယ်။ Language နှစ်ခုရဲ့ သဘောတရားကို အတတ်နိုင်ဆုံး ရှင်းလင်း သဘောပေါက်နားလည်စေချင်ပါတယ်။ ဒီတော့ Java ရဲ့ own package တွေ ဖန်တီးပြီးပြုဆိုတော့ main () program ကို တည်ဆောက်ရအောင် လား။ Listing 2.24 အနေနဲ့ ရေးဖွဲ့ထားတဲ့ program code\_line လေးတွေကို အသုံးပြုပြီးတော့ program ကို ဖန်တီး တည်ဆောက်ပါ။

Listing 2.24 use my own package method in JAVA

```
1. import calculate.Cal;
2. public class testOwnPackage {
3.     public static void main (String args [ ]) {
4.         cal obj = new cal ();
5.         double Length = 9.0;
6.         double width = 11.0;
7.         public getArea (double L, double w)
8.             {length = L; width = w;}
9.         System.out.println ("Area is " + obj Cal Area ());
10.    }
11. }
```

output  
Area is 99

Note: ဒီ program ကို ရေးပြီးတဲ့ အခါမှာ ထပ်မံပြီးတော့ သတိ ပေးပါဉိုးမယ်။ ဒီ file ကို စာဖတ်သူရဲ့ Computer ထဲ System's CLASSPATH ဖြစ်တဲ့ -

C: \j2sdk 1.4.1-01 >

C: \jdk 1.4 >

C: \java 2 > အစုံရှိတဲ့ စာဖတ်သူ Instrall လုပ်ထားတဲ့ JAVA Developement kit ရှိရာအောက်မှာ save လုပ်ပြီးမှာပဲဖြစ်ဖြစ် C: \j2sdk 1.4 .1-01 /bin > ရဲ့ အောက်မှာပဲဖြစ်ဖြစ် Save လုပ်ပြီးတော့မှာ Compile လုပ်ရမှာပါ။ ဒါကို အထူးသတိထားပါ။

Analysis

၁။ Listing 2.24 မှာရေးထားတဲ့ program code-line ဟာ စာဖတ်သူအတွက် ဘယ်လို့မျှ complex ဖြစ်စရာ မရှိပါဘူး။ ဒါဟာ အလွန် ရှိုးစင်းလွယ်ကူပြီးတော့ အထပ်ထပ်အခါခါ ရေးသား လေ့လာခဲ့ရတဲ့ Program type ဖြစ်လို့ပါပဲ။ လိုင်း နံပါတ် (၁) မှာ ကျွန်တော်တို့ ရှေ့မှာ ရေးဖွဲ့ဆိုတဲ့ package file ကို import လုပ် ခေါ်ယူပါတယ်။ ဒီအခါမှာ ကျွန်တော် တို့ ဖန်တီးခဲ့တဲ့ package file ဟာ program ရဲ့ Base Class & object အဖြစ် ရောက်ရရှိလာတာပါ။ အဲဒီနောက် လိုင်းနံပါတ် (၂) မှာတော့ test Own Package ဆိုပြီးတော့ Primary Class တစ်ခုကို ဖန်တီးပါတယ်။ အဲဒီ primary

## ကွန်ပျူးတာသင်ခန်းစာ

<p>class ထဲမှာတော့ package file အဖြစ် ရေးသားထားတဲ့ Cal ဆိုတဲ့ Class ကို object အနေနဲ့ Obj လို့ Name ပေးပြီးတော့ ခေါ်ယူလိုက်ပါတယ်။</p> <p>၂။ လိုင်းနံပါတ် (၅) နဲ့ (၆) မှာတော့ length နဲ့ width တို့၏ တန်ဖိုးကို Declare လုပ်ပါတယ်။ Data type ကိုတော့ double data type ကို useage လုပ်ထားပါတယ်။ လိုင်းနံပါတ် (၇) မှာတော့ getArea () ဆိုတဲ့ function လေးကို ရေးဖွံ့ပါတယ်။ အဒီ function မှာ parameter arguments အနေနဲ့ ရေးသားပြီးတော့ function () ရဲ့ body ထဲမှာ assign လုပ်ပြီးတော့ ညီယူထားပါတယ်။</p> <p>၃။ လိုင်းနံပါတ် (၁၀) မှာတော့ program ရဲ့ ပင်မ body ထဲမှာ ပါဝင်တဲ့ data assign value တွေကို အသုံးပြုပြီးတော့ package file အဖြစ် ရေးသားထားတဲ့ Class ထဲက getArea () ဆိုတဲ့ method ကို ခေါ်ယူအသုံးချဖြီးတော့ output ထုတ်ပေးထားပါတယ်။ ဒါဟာ ဒီ package မှာ ပါဝင်တဲ့ code-line တွေရဲ့ သဘောတရားပါပဲ။</p> <p>ဒီနေ့ရာမှာ စာဖတ်သူဟာကန် Complain တက်လာစရာ ရှိတော်တော့ package file အဖြစ် သီးခြားရေးဖွံ့မနေတော့တဲ့ Class အဖြစ် ဖန်တီးမယ်၊ ခေါ်ယူ အသုံးချမယ်ဆိုရင် ဒီ Class ကို ပင်မ main () program ကသာ ခေါ်ယူအသုံးချလို့ ရမှာပါ။ အခြား program ကသာ ခေါ်ယူမယ်ဆိုရင် မရနိုင်တော့ပါဘူး။ ဒီတော့ အခြား user တွေအတွက် support မဖြစ်နိုင်တော့ပါဘူး။</p> <p>ဒီအချက်ဟာ မတူပါဘူး။ User ကို support လေးတဲ့ type ဟာလည်း မတူပြန်ပါဘူး။ ဘာကြောင်လဲဆိုတော့ Class အနေနဲ့ ဖန်တီးမယ်၊ ခေါ်ယူ အသုံးချမယ်ဆိုရင် ဒီ Class ကို ပင်မ main () program ကသာ ခေါ်ယူအသုံးချလို့ ရမှာပါ။ အခြား program ကသာ ခေါ်ယူမယ်ဆိုရင် မရနိုင်တော့ပါဘူး။ ဒီတော့ အခြား user တွေအတွက် support မဖြစ်နိုင်တော့ပါဘူး။</p> <p>ကဲအခု ကျွန်ုတ်ပိုဂျပ်းတဲ့ နည်းလမ်းနဲ့သာ ဖန်တီး တည် ဆောက်မယ်ဆိုရင်တော့ API (Application Programming Interface) method အနေနဲ့ ဖြစ်တည်သွားတာပါ။ ဒီ API method ကို အခြား program တွေ ရေးသားစဉ်မှာ import လုပ်ပြီးတော့ ခေါ်ယူအသုံးချလို့ ရပါတယ်။ ဒါဟာ API method တွေရဲ့ သဘောတရားပါပဲ။</p>	<p><u>Note :</u> API method တွေနဲ့ ပတ်သက်ပြီးတော့ 1.5 Arrays of JAVA ဆိုတဲ့ အခန်းကဏ္ဍမှာ အခြေခံကျကျ ပြည့်စုံအောင် ရှင်းပြပေးခဲ့ပြီပါပြီ။ အခုအချိန်မှာ ပြန်လှုံးလေ့ လာစေခဲ့ပါတယ်။ API method ရဲ့ နောက်ထပ် support ဖြစ်တဲ့ အချက်ကတော့ user တွေအနေနဲ့ ရှုက programmer တွေ ဖန်တီးသွားတဲ့ method တွေကို မိမိထိကြုံကြတဲ့ features တွေ ထပ်မံပြည့်စွဲကြပြီး ဆင့်များ package တွေ ဖန်တီးလို့ရပါတယ်။ ဥပမာေးရရင်-</p> <p><u>Sample</u></p> <ul style="list-style-type: none"> <li>◆ Package A; ကို Programmer A က ဖန်တီးပါတယ်။</li> <li>◆ ဒီ Package A ထဲမှာ API method တစ်ခု ပါဝင်ပါတယ်။</li> <li>◆ Package B ; ကို Programmer B က ဖန်တီးပါတယ်။</li> <li>◆ ဒီ package B ထဲမှာ package A ထဲက API method ကို အခြေပြုပြီးတော့ စိတ်ကြုံကြ API method () ကို ရေးသား ဖန်တီးပါတယ်။ ဒီ package B ထဲမှာ အရှေက programmer A ရေးသားထားတဲ့ package A တစ်ခုလုံး ကို import လုပ်ထားပါတယ်။</li> <li>◆ ဒီအခါ package B ကို ခေါ်ယူ အသုံးချရင် package A ကိုပါ အလိုအလေ့ကြပ် involve အနေနဲ့ ပါဝင်လာမှာပါ။</li> </ul> <p>ဒါဟာ ရှေ့က ဖန်တီးသွားတဲ့ API method တွေကို အခြေပြုပြီးတော့ ဆင့်များ package တွေကို ဖန်တီးတည် ဆောက်တဲ့နည်းလမ်းပါပဲ။ ဒါဆိုရင် အခြေခံ package file တွေ တည် ဆောက်နိုင်လည်းလမ်းကို နားလည်သောပေါက်သွားပြီဖြစ်တဲ့ အတွက် ထပ်မံပြုပြီးတော့ လေ့လာနိုင်အောင် 2.5.2 Sample Application of JAVA ဆိုတဲ့ အခန်းကဏ္ဍကို ပို့ချေပေးထားပါတယ်။ စာဖတ်သူ အတွက် JAVA Language နဲ့ ပတ်သက်တဲ့ အကောင်းဆုံး အထောက်အကြေဖြစ်စေနိုင်မယ်လို့ ယဲ့ကြေညာပါတယ်။ က သေသေ ချာချာလေး ဆက်လက် လေ့လာလိုက်ပါပြီး။</p>
---	---

ကျေးဇူးတွေကြည့်တိုက်များကို အား(၃)အားဖြင့် ဖွဲ့စည်းထုတေသန

ထိသို့ဆောင်ရွက်ရပ်တွင် နိုင်တော်ကြအား၊ စေတနာဂျာ၏ပြည်သူများ၏အားနှင့် ဒေသခံပြည်သူများ၏အားဟာ အား (၃)အော်ဖြင့် ၁၉-၁-၂၀၀၆ ရက်နောက် ကျေးဇူးတွေ့ကြည့်တိကိုပေါင်း (၄၈၆၆၇) တိက် စွဲစည်းပြီး ဖြစ်ပါသည်။ ကျုန်ကျေးဇူးကိုလည်း ယင်းအား (၃)အော် ဖြင့် ကျေးဇူးတွေ့ကြည့်တိကိုများဆက်လက် ဖွင့်လွှာသွားမည် ဖြစ်ပါသည်။

# JAVA DEVELOPER GUIDE

### 2.5.2 SAMPLE APPLICATION OF JAVA

గ్విఫెలోవ్ ట్రీ ఆఫెస్ట్ కిఎఎఫ్స్: గాల్పామ్స్ ఒబజెక్ట్ & క్లాస్ వీటి అధింస్: మూగ్గిప్ టెంచ్ టెంచ్ లేస్: రెగ్యాట్యు: వ్యూ: ఫిన్డ్ పిటాయ్॥ ఉపభాప్రార్థింట్టే: Access Control ను passing Arguments అధింస్:॥ ప్రైస్: టెంచ్: Constructor ను. Destructor Method॥ ఫెఫాక్ట్ కి ఇంట్టే: Class రూ అప్లోద్ పండిం ఫెచ్చర్స్ టెంచ్॥ వీటెగ్గి తాపతల్లు ఆఫెస్ట్ కి షెల్వ్యూగెండ్: షెల్వ్యూగెండ్ క్రొప్పబి॥

Interface အပိုင်းမှာဆိုရင်လည်း: Implementing and using Interface သဘောတရား။ ပြီးတော့ Extending and Methods Inside သဘောတရား စတာတွေ ကျယ်ကျယ်ပြန်ပြန် ပါဝင်နေပါတယ်။

ကဲ – ဒီတော့ စာဖတ်သူအနေနဲ့ အနည်းငယ်ရောထွေးကောင်း  
ရောထွေးနေနိုင်တဲ့ ဒီအခန်းကဏ္ဍမှာ လေ့ကျင့်ရမယ့် program  
အနေနဲ့ အရှေ့က သင်ကြားခဲ့တဲ့ သဘောတရားတွေ အကုန်ပါဝင်  
နေတဲ့ Storefront Application Program ကို လေ့လာကြည့်ကြ  
ရအောင်လား။ ဒါဟာ JAVA programma တစ်လောက်အတွက်  
ပြည့်စုတဲ့ အတွေအကြိုက် ရရှိစေနိုင် အထောက်အကူ ပြစ်စေနိုင်လိမ့်  
မယ်လို့ ယုံကြည်ပါတယ်။ လိုအပ်သလို class & object တွေကို  
ဖန်တီးပြီးတော့ project သဖွယ် ဖန်တီးရယူတောင် သေသေ  
ချာချာ လေ့လာ သင်ယူစေချင်ပါတယ်။ နားလည်အောင် အတော်  
နိုင်ဆုံး ကြိုးစားပို့ချေပေးထားပါတယ်။

## Creating an Online Storefront Application

ဒီ program ဟာ Application program ဖြစ်တယ်ကြောင့် code တွေ ရေးရတာ များပြားမှရိုးသလို တည်ဆောက် ဖွဲ့စည်းမှမှာ လည်း complex ဖြစ်လာနိုင်တဲ့အတွက် Multifile program သဖြင့် ရေးသားပြီးတော့ အဲဒီ ရေးသားထားတဲ့ Multifile program တွေ စုစုပေါင်းမြို့မြို့ အတွက် Executable Program ဖြစ်လာအောင်

နောက်မှတ်ပုံ၊ Project တစ်ခုသဖွယ် ဖန်တီးရတာဖြစ်တဲ့  
အတွက် program ရဲ့ တစ်စိတ်တစ်ခေါ်စီကို ခွဲခြမ်းရေးတဲ့အခါ  
က စလို Date type, variables, Method စတေတွေကို ရရှိ  
တနိက် လေ့လာရပါမယ်။ ဒါ Application project မှာ အမိက  
စွမ်းဆောင်ရည်အပြင် encapsulation ဆိုတဲ့ အချက်ကိုပါ ထည့်  
သွင်း စဉ်းစားကြရပါမယ်။ က ဒါဆိုရင် သင်ခန်းစာတစ်ခုကို  
လေ့လာသလိုပျိုးပဲ သုံးသပ်ကြရအောင်လား။ အမိက ဒါ program  
ရဲ့ ရည်ရွယ်ချက်တော့-

- ♦ Calculating the sale price of each item depending on how much of it is presently in stock
  - ♦ Sorting items according to sale price.

စတဲ့ ရည်ရွယ်ချက်တွေနဲ့ အမိက တည်ဆောက်တာပါပဲ။ ဒါဟာ အခြေခံ အကြမ်းဖျင်းအားဖြင့် တည်ဆောက်တာလည်းပဲ ဖြစ်ပါတယ်။ စာဖတ်သူအနေနဲ့ လိုချင်တဲ့ features တွေကိုလည်း ထပ်မံဖြည့်စွက် ရေးသားလို့ ရှုံးငွေပါတယ်။

Storefront Application မှာ ပင်မ အမိက container, class နှစ်ခုက အမိက ပါဝင်ပါတယ်။ အဲဒါကတော့ Item Class နဲ့ Storefront Class တို့ပဲ ဖြစ်ပါတယ်။ အဲဒီ Class နှစ်ခုဟာ မိမိဂိုလ်တိုင်ပဲ ဖန်တီးရေးသားရမယ့် class တွေပါ။ အဲဒီ class တွေကို ကျွဲ့နဲ့တော်က ရေးသားပြီးတော့ package အဖြစ် ဖန်တီးရယူပါမယ်။ ဒီတော့ Package\_Name အဖြစ် com.perfect.ecommerce ထဲမှာ Organized ထုတ်မှာ ဖြစ်ပါတယ်။ ဒါကြောင့် အရင်ဆုံး စာဖတ်သူရဲ့ System directory ထဲမှာ အဲဒီ package's class ကို store လုပ်ဖို့ directory structure ကိုဖန်တီးဖို့ လိုလာပါပြီ။

## ကွန်ပျူးတာသင်ခန်းစာ

1.4 ကို Instrall လုပ်ထားရင် C:\ jdk 1.4 အနေနဲ့  
တည်ရှိမှာပါ။  
ဒီတော့ ကျွန်တော်တို့အနေနဲ့ C:\jdk 1.4\com\prefect\ecommerce ဆိုတဲ့ System's CLASSPATH ကို for package in the folder listed အနေနဲ့ ဖုန်တိုးရပါမယ်။

ဒါမှသာယျုင် ကျွန်တော်တို့ ဖုန်တိုးမယ့် program file တွေ ကို ဖုန်တိုးပြီးတဲ့အခါမှာ package's file အနေနဲ့ ဒီ folder တွေထဲမှာ save လုပ်ရမှာပါပဲ။ အခြေခံအနေနဲ့ ပြောရင်တော့ Main() Function/ Method မပါတဲ့ class တွေရယ်ပြီးတော့ Method & Variables တွေရယ်ကို ရေးသားပြီးရင် အချုပ်တည်ဆောက်ထားတဲ့ System's CLASSPATH ထဲမှာပဲ Save လုပ်သိမ်းဆည်းပြီးတော့ Main() Function /Method ပါဝင်မယ့် program file ရဲ့ထိပ်မှာ import file အနေနဲ့ ရည်ညွှန်းပြီးတော့ ခေါ်ယူ အသုံးချမှာ ဖြစ်ပါတယ်။ ဒါကို စာဖတ်သူရဲ့ Computer ထဲမှာ လက်တွေ တည်ဆောက်ပြီးတော့ လေ့လာမယ်ဆိုရင် ပိုပြီး အကျိုးရှုလာမှာပါ။

ဂဲ- အချုပ်သားမယ့် Application နဲ့ ပတ်သက်ပြီးတော့ လေ့လာ ဆုံးသပ်ကြည့်ပြီးရင် အမိက package file အဖြစ် ရည်ညွှန်း တည်ဆောက်ရမယ့် Item.java ဆိုတဲ့ class ရပို၊ storefront.java ဆိုတဲ့ class file နှစ်ခုကို အရင် စတင်ပြီးတော့ ရေးသား လေ့လာကြရအောင်လား။ Listing 2.25 အနေနဲ့ Item ဆိုတဲ့ Class ကို ဖုန်တိုး ရေးသားပေးထားပါတယ်။ ဒီ class file ကို C:\jdk 5.4\com\prefect\ecommerce ဆိုတဲ့ system's CLASSPATH ထဲမှာ save လုပ် သိမ်းဆည်းပါမယ်။ ဂဲ Listing 2.25 ကို လေ့လာကြည့်ပါ။

**Listing 2.25 The Full Text of Item.java**

```

1: package com. prefect. ecommerce;
2:
3: import java. util . *;
4:
5: public class Item implements Comparable {
6:     private String id;
7:     private String name;
8:     private double retail;
9:     private int quantity;
10:    private double price;
11:
12:    Item (String idIn, String nameIn, String retailIn, String
13:          quanIn) {
14:        id = idIn;
15:        name = nameIn;
16:        retail = Double .parse Double (retai L/n);
17:        quantity = Integer .parse Int (quan In);
18:        if (quantity > 400)
19:            price = retail * 5 D ;

```

```

20:    else if (quantity > 200)
21:        price = retail * 6D
22:    else
23:        price = retail * 7D;
24:    Price = Math. floor (price * 100 + .5 ) / 100 ;
25: }
26:
27: public int compare To (Object obj) {
28:     Item temp = (Item) obj;
29:     if (this .price < temp .price)
30:         return 1 ;
31:     else if (this .price > temp.price)
32:         return 1 ;
33:     return 0;
34: }
35:
36: public String getId () {
37:     return id;
38: }
39:
40: public String get Name () {
41:     return name;
42: }
43:
44: public double get Retail () {
45:     return retail;
46: }
47:
48: public int get Quantity () {
49:     return quantity;
50: }
51:
52: public double get price () {
53:     return price;
54: }
55: }
```

### Analysis

၁။ ဒီ program code ဟာ package file တစ်ခုကို တည်ဆောက်မယ့် class file တစ်ခုပဲဖြစ်ပါတယ်။ ဒီ class ဟာ အခုပ်တိုးနေတဲ့ Application မှာ အမိက အထောက်အကြံ class တစ်ခု ဖြစ်ပါတယ်။ ဒီ class ထဲမှာ လိုင်းနံပါတ် (၁) မှာ ရေးသားထားတဲ့ package com. prefact. ecommerce ဆိုတဲ့ code line ကတော့ own package တစ်ခုကို တည်ဆောက်မယ်ဆိုရင် အမြတ်စေ ရည်ညွှန်းရ ဆုံး System's CLASSPATH ပါပဲ။ ရည်ညွှန်းရမယ့် system CLASSPATH အတွက်လည်း C:\jdk 1.4 ထဲမှာ sub-directory အနေနဲ့လည်း သွားရောက် ဖုန်တိုးပြီးသားဖြစ်ရပါမယ်။ ဂဲ ဒါဆိုရင်တော့ လိုင်းနံပါတ် (၁) က ဘာကို ရည်ညွှန်းတယ်ဆိုတာကို သဘောပေါက်သွားမှာပါ။ ဒီ class ထဲမှာ product ID code, name, how many are in stock (quantity), and the retail and the

sale price ගොනී ලග්න් තුන්වලුවූ හෝ output  
තුන්වලුවූ පරිජාතයේ function ( ) සේපුද් තුන්වලුවූ Methods  
Variables වෙ එම්බැංකිතයි॥

Note: Methods တွေကို ဖန်တီး ရေးသားတဲ့အခါမှာ get ဆိုတဲ့ စာတုံးရဲ့နောက်မှာ Capitalized letter နဲ့ ID, Name, Retail အစရိတဲ့ methodes\_Name လေးတွေကို ရေးသားထားတာကို တွေ့ရမှုပါ။ ဒီလို ဖန်တီး ရေးသားမှုဟာ JAVA ရဲ့ Class Libairy မှာ own package အဖြစ် ဖန်တီး ရေးသားသူတိုင် လိုက်နာရ မယ့် standard အဖြစ်သတ်မှတ်တဲ့ စည်းမျဉ်းစည်းကမ်းပါပဲ။ ဒါကို စာပတ်သူအနေနဲ့ သိရှိ နားလည်ပြီးတော့ လိုက်နာရပါမယ်။

၂။ နောက်ထပ် သတိပြု လေ့လာရမယ့် အချက်ကတော့ လိုင်း  
 နံပါတ် (၅) မှာ တွေ့ရမှာဖြစ်တဲ့ public class Item  
 implements comparable ဆိုတဲ့ code line လေးကိုပါပဲ။  
 ဒီအချက်ကို တွေ့တာနဲ့ ဒီ Item ဆိုတဲ့ Class ဟာ  
 Comparable ဆိုတဲ့ interface ကို implements လုပ်  
 ထားတာပါပဲဆိုတာကို သိရှိ နားလည် သတေသနပါက်ရပါ  
 မယ်။ Interface တွေကို implements လုပ်တဲ့ သတေသန  
 တရားနဲ့ ပတ်သက်ပြီးတော့ ရှေ့သင်ခန်းစာတွေမှာ အမြောက်  
 အမြား ပိုချေပေးတဲ့အတွက် စာဖတ်သူအတွက် အခက်အခဲ  
 မရှိလောက်ပါဘူး။

၃ Item Class မှာ Comparable interface ကို implements လုပ်ရခြင်း အဓိကရည်ရွယ်ချက်ကတေသာ့ it easy to sort a class's objects အနေနဲ့ ဖန်တီးဖို့ပါပဲ။ ဒါ comparable ဆိတဲ့ interface မှာ has only one method လို့ခေါ်စိုင်တဲ့ Compare To (object) method တစ်ခုတေသား ပါဝင်ပါတယ်။

ဒါဖိုရင်တော့ ကျွန်တော်တို့ ဒီ comparable interface ထဲမှာ ပါဝင်တဲ့ compareTo ( ) method ကို အသုံးပြချင်လို့ implements လုပ်တာပဲဆိုတာ သဘောပါက်ရမယ်နော်။ ဒီ compare To ( ) method အလုပ်လုပ်ပုံကတော့-

- ◆ Class တစ်ခုအတွင်းမှာ objects နှစ်ခုကို နိုင်းယုံပါတယ်။



ဒီ CompareTo () method ကို လိုင်းနဲ့ပါတ် (၂၇) ကအနေ လိုင်းနဲ့ပါတ် (၃၄) အထိ ရေးသားထားပါတယ်။ အလွယ်တကူ ပြောရရှင်တော့ ဒီ CompareTo () method ဟာ Sarting လုပ် ပေးနိုင်တဲ့ method ပဲပေါ့။ ဒီတော့ 1.5.1 sort Algorithm ဆိုပြီးတော့ လေ့လာခဲ့ရတဲ့ sorting method တွေနဲ့ သဘော ပေါက်ခဲ့ရင် ဒီ compareTo () method ဟာ အကျယ်ချဲ့ ရှင်းပြစ်ရာ မလိုပါဘူး။ စာဖတ်သူကိုယ်တိုင်ပဲ လေ့လာကြည့်ပါ။ Compare To () method ဟာ အခု Item Class မှာ အမိက override လုပ်ပေးတဲ့ method ပါပဲ။ ဘာကြောင့်လဆိုတော့ ဒီ method ဟာ Item Class ထဲမှာ call to sort and array, linked List, or Other collection of these objects အနေနဲ့ ဖုန်တီးပေးလိုပါပဲ။ ဒါကို Storefront ဆိုတဲ့ Class ကို ရေး သားပြီးရင် ပိုပြီး နားလည်သဘောပေါက်သွားမှာပါ။ အခါ Item နဲ့ Storefront ဆိုတဲ့ Class နှစ်ခုစလုံးဟာ Comparable ဆိုတဲ့ interface ကို implements လုပ်ထားတဲ့ object ကို အမိက Override လုပ်နေရတယ်ဆိုတာကို စာဖတ်သူ လုံးဝ နားလည် သဘောပေါက်သွားမှာပါ။

၃။ အခု ရေးသားထားတဲ့ Item Class ရဲ့ constructor function ဖြစ်တဲ့ Item ( ) function ကို လိုင်းနံပါတ် (၁၂) က နေ့ လိုင်းနံပါတ် (၅၂) အတွင်းမှာ ရေးသားထားပါတယ်။ အဲဒီ Item ( ) function ကို လေ့လာကြည့်ရင် passing arguments အနေနဲ့ String object ကို အသုံးပြုပြီးတော့ ဒီလို ရေးသားထားပါတယ်။

Item (String idIn, String name In ,String retail In, String quanIn) ଯିଟିରେତେବୁ ପାସ ବ୍ୟାକ୍ ଦିଲ୍ଲାଙ୍କାରୀ ହେବାରେ ପାରିବାରେ ପାରିବାରେ ପାରିବାରେ

```

id = idIn;
name = nameIn;
retail = Double.parseDouble(retailIn);
quantity = Integer.parseInt(quantityIn);

aziPrise:陶ူမှ လိုင်းနံပါတ် (၆) ကနေ လိုင်းနံပါတ် (၁၀) အတွင်း
Declare လုပ်ခဲ့တဲ့ private data variables တွေနဲ့ passing
arguments တွေရဲ့ တိုင်းကို ညီယူပါတယ်။ ဒါဟာ အခိုကျေ
တဲ့ အချက်ဆိုတာကို သဘောပေါက်ရပါမယ်။ ဘာကြောင့်လဲဆို
တော့ အခုခြင်းပြနေတဲ့ Item () function ဟာ constructor
function ဖြစ်တယ်ပါပဲ။
```

ဒုအပ်င် Data Input လုပ်တဲ့အခါမှာ id, name, retail, quantity စတာတောက် string object လို ။

## ကွန်ပါတာသင်ခန်းစာ

data type ကို အသုံးပြုပြီး လက်ခံပါတယ်။ id နဲ့ name ကတော့ တွက်ချက်မှုမှာ ပါဝင်ဖို့ လိုတဲ့အတွက် အရေးမကြိုးပေါမယ့် retail နဲ့ quantity ကတော့ ပါဝင်တွက်ချက်ရှုပါ။

ဒီတော့ String data type ကနဲ့ပြီးတော့ numeric value  
ဖြစ်အောင် converted လုပ်ပေးရပါမယ်။ အဲဒါကြောင့် လိုင်း  
နံပါတ် (၁၅) မှာ -

```
retail = Double.parseDouble(retailIn);
```

```
quantity = Integer . parse Int (quanIn);
```

အနေနဲ့ quantity ဆိုတဲ့ data variable ကို string data type ကနဲ့ Integer data type အဖြစ် convertedလုပ်ပေးထားပါတယ်။

Note: အတိအကျပြောရင်တော့ Double, Integer စတဲ့ data type အဖြစ်တင်မဟုတ်ပါဘူး။ parseDouble , parseInt စတဲ့ ရေးသားမှုကြောင့် numeric values လိုပေါ်တဲ့ Decimal points အထိ အတိအကျ တွက်ထုတ်ပေးနိုင်တဲ့ ကိန်းစတ်တန်ဖိုးအဖြစ် ပြောင်းလဲသွားတာပါ။ ဒါကို အလွယ်တကူ နားလည်နိုင်အောင်လို့ String data type ကနေ Double, Integer စတဲ့ data type ကို converted လုပ်ပေးတယ်လို့ အလွယ်ရင်းပြတာပါပဲ။

package file အနေဖြင့်ပဲ ရည်ညွှန်း အသုံးချမှာဆိတဲ့အချက်ပဲဖြစ်ပါတယ်။

ကဲ - ဒီတော့ Listing 2.26 အနေနဲ့ Storefront Class file ကို ရေးသားပါမယ်။ စာဖတ်သူအနေနဲ့ သေသေချာချာလေ့လာကြည့်ပါ။ ဒီ class file ဟာလည်း အပေါ်က Item Class file လိုပ် package file အနေနဲ့ ရည်ညွှန်စံတဲ့ Class file ဆိတာကို မထွေနဲ့နော်။ သေသေချာချာလေ့လာကြည့်ပါ။

### Listing 2.26 The Full Text of Storefront.java

```
1: package com.prefect.ecommerce;
2:
3: import java.util.*;
4:
5: public class Storefront {
6:     private LinkedList catalog = new LinkedList();
7:
8:     public void add Item (String id, String name, String
9:         price,
10:        String quant) {
11:            Item it = new Item (id, name, price, quant);
12:            catalog.add (it);
13:        }
14:
15:        public Item get Item (int i) {
16:            return (Item) catalog.get (i);
17:        }
18:
19:        public int getSize () {
20:            return catalog.size ();
21:        }
22:
23:        public void sort () {
24:            Collections.sort (catalog);
25:        }
26:    }
```

ອັນ Listing 2.26 ອາຟັກ. ໄຣະວະທະບູ Storefront file ມີ code line (26) line ພິມດີປິຕຍໍ່. ກ່ອນ- ອ່າວີເຕັມ ກົງສື່ຕົວຕິດຕິ່ຕິ່ອັນ Listing 2.26 ອາຟັກ. ສີ Storefront class file ຮຸ່ ພິມດີປິຕຍໍ່ ຕະຫຼອງ Line -By-Line ອາຟັກ. ໝູ້ລາຍລຸ່ມກົງການແພັນລະບຸ Analysis

၁။ အခုရေးသားခဲ့တဲ့ Storefront Class ဟာ စုစဉ်ထူးတဲ့ product တွေကို manage လုပ်ဖို့အတွက် ရည်ရွယ်ပြီးတော့ ရေးသားထူးတာပါဘူး။ အဲဒီ product တွေဟာ Item class ထဲမှာပါဝင်ပါတယ်။ ဒီတော့ private keyword ကို အသုံးပြုပြီးတော့ LinkedList Class ကို Instance variable \_Name အနေနဲ့ Catalog ဆိုပြီး အမည်ပေး ခေါ်ယူ အသုံးချက်ထူးပါတယ်။ လိုင်းနံပါတ် (၆) မှာ စာဖတ်သူအနေနဲ့ တွေ့ရမှာတော့။

## ကွန်ပူးတာသင်ခန်းစာ

Note: Item Class ဟာ Listing အနေနဲ့ ရှုမှာ ကျွန်တော်တို့ ရေးသားခဲ့တဲ့ Class ပါပဲ။ ဒီ class ကို object အနေနဲ့ ခေါ်ယူ အသုံးချလိုက်တာကို လိုင်းနံပါတ်(၁၁) မှာ တွေ့ရ မှာပါ။ သတိပြုရမယ့်အချက်ကတော့ pass By value လုပ်နိုင်ဖို့အတွက် Pass By Arguments လုပ်ထားတာ ကိုပါပဲ။ ဒီအချက်ကို သတိပြုရမှာပါ။

လိုင်းနံပါတ် (၈) ကနေ လိုင်းနံပါတ် (၁၃) အတွင်းမှာတော့ addItem ( ) ဆိုတဲ့ method ကို ဖန်တီးထားပါတယ်။ အဲဒီ addItem () method ဟာ Item object ကို based on လုပ်ဖြီး မှ passing arguments အနေနဲ့ arguments (၄) ခုကို send to method အနေနဲ့ အသုံးပြုထားပါတယ်။ ID, name ,price and quantity ဆိုတဲ့ အရာတွေပါပဲ။ ဒီ variables\_Name တွေကိုတော့ Item Class မှာ တွေ့ခဲ့ရပြီးဖြစ်လို့ ရောထွေးနေစရာ မရှိပါဘူး။ Item Class မှာ Declare လုပ်ခဲ့တဲ့ Variables တွေ ဖြစ်တဲ့အတွက် ထင်မံ Declare လုပ်စရာမလိုတော့ပါဘူး။ အလွယ် တကူ ခေါ်ယူ အသုံးချခဲ့ပါပဲ။ လိုင်းနံပါတ် (၁၁) မှာ Item ကို object အနေနဲ့ ဖန်တီးပြီးတဲ့အခါမှာ Linkedlist Class ရဲ့ add () method ကို ခေါ်ယူလိုက်တာကို လိုင်းနံပါတ် (၂၂) မှာ တွေ့ရ မှာပါ။ Linkedlist Class ကို Object\_Name & Instance Variable\_Name အနေနဲ့ Catalog ဆိုပြီး လိုင်းနံပါတ် (၆) မှာ object အဖြစ် ဖန်တီးခဲ့တာဆိုတော့ ခေါ်ယူ အသုံးချတဲ့အခါမှာ Catalog.add(it);  
ဆိုပြီးတော့ ခေါ်ယူ အသုံးချထားပါပဲ။ ဒါဆိုရင် စာဖတ်သူအနေနဲ့ ရှင်းသွားမှာပါ။ Item object ကို arguments အနေနဲ့ ပြန်လည် အသုံးချထားတာကို တွေ့ရမှာပါ။

Note: လက်ရှိ အသုံးချနေတဲ့ object ကို arguments အနေနဲ့ ပြန်လည် အသုံးချတဲ့ method\_type မျိုးကို ကျွန်တော် အနေနဲ့ ရှုံးသင်ခန်းစာတွေမှာ သေသေချာချာ မလေ့လာ ခဲ့ရင်၊ နားမလည်ခဲ့ရင် ဒီနေရာမှာ သေသေချာချာ လေ့လာကြည့်ပါ။

J။ က ဆက်လက်ပြီးတော့ getItem () နဲ့ getsize () method နှစ်ခုကို ဆက်လက် လေ့လာကြည့်ကြရအင်။ လိုင်းနံပါတ် (၁၅) မှာတော့ getItem () method ကို ရေးသားထားတာ ကို တွေ့ရမှာပါ။ ဒီ method ကို လေ့လာကြည့်မယ်ဆိုရင် linkedlist Class ထဲမှာ get () method ကို တိုက်ရှိက် ခေါ်ယူ အသုံးချထားပါပဲ။ ဒါကြောင့်  
return (Item) Catalog.get(i);

ဆိုပြီး ရေးသားထားတာကို တွေ့ရမှာပါ။ ဒါဟာ object ဖြစ်တဲ့ Item ကိုဝါ store at that location in the

LinkedList အနေနဲ့ ရေးသားထားတာပါပဲ။ ဒါကြောင့် return ခဲ့ နောက်မှာ (Item) လို့ ရေးသားထားတာကို တွေ့ရတာ ပါ။ ပြီးတော့ got () method ကို ခေါ်ယူတဲ့အခါမှာလဲ index\_number ကို arguments အနေနဲ့ ထည့်သွင်း အသုံးချထားလို့ catalog.got(i); ဆိုပြီး ရေးသားထားတာကို စာဖတ်သူအနေနဲ့ တွေ့ရပြီးတော့ သတိပြုမှာပါ။  
လိုင်းနံပါတ် (၁၉) ကနေ လိုင်းနံပါတ် (၂၁) အတွင်း ရေးသားထားတာတော့ getSize () method ပါပဲ။ getSize () method, ဟာလဲ LinkedList class ထဲက size () method ကို ခေါ်ယူအသုံးချထားတာပါပဲ။ ဒါကြောင့်  
return catalog.size ();  
ဆိုပြီး ရေးသားထားတာကို တွေ့ရမှာပါ။

Note: ဒီ program မှာ အခု ခေါ်ယူအသုံးချထားတဲ့ LinkedList class ဟာ အခါရေးသားတဲ့ program ရဲ့ အရေးပါတဲ့ method () တွေဖြစ်တဲ့ add () method, get () method, size () method, sort () method အစုစုတဲ့ Methods တွေ အပါအဝင် ဖန်တီးထားတာဖြစ်ပါတယ်။ ဒီ LinkedList class ဟာ လိုင်းနံပါတ် (၃) မှာ Declare လုပ်ခဲ့တဲ့  
java.util.\*;  
package ထဲ မှာ ပါဝင်ပါတယ်။ ဒါကြောင့် လိုင်းနံပါတ် (၃) မှာ စာဖတ်သူအနေနဲ့  
import java.util.\*;

ဆိုပြီး ရေးသားထားတာကို တွေ့ရမှာပါ။ ဒါဟာ JAVA ရဲ့ package file တွေ ခေါ်ယူ အသုံးချထားတာကို စာဖတ်သူအနေနဲ့ တွေ့မြင် လေ့လာရမှာပါ။ အခု ရေးသား ပို့ချပေးနေတဲ့ အချက်တွေကို စာဖတ်သူအနေနဲ့ သေသေချာချာလေ့လာပါ။

၃။ လိုင်းနံပါတ် (၂၃) မှာ Sort () method ကို ရေးသားထားတာကို တွေ့ရမှာပါ။ ဒီအခါမှာ စာဖတ်သူအနေနဲ့ Listing က Item class ကို ရေးသားထားအခါမှာ Comparable Interface ကို implements လုပ်ပြီးတော့ ခေါ်ယူ အသုံးချထားတာကို တွေ့ရမှာပါ။ အဲဒီ Item Class မှာ အသုံးချခဲ့တဲ့ comparable Interface ဟာ java.util.\*; package ထဲမှာ ပါဝင်ပါတယ်။ အခု ဒီ program မှာလည်း java.util.\*; package ထဲမှာ ပါဝင်တဲ့ collection class ကို ခေါ်ယူ အသုံးချထားပါတယ်။ Collection class ကို Instance\_Name အနေနဲ့ မဟုတ်ဘဲ တိုက်ရှိပဲ ခေါ်ယူ အသုံးချေးထားပါတယ်။ ဒါကြောင့် လိုင်းနံပါတ် (၂၄) မှာ  
Collection.sort(catalog);

## ကုန်ပါတာသင်ခန်းတ

နိုင်ပြီး ရေးသားထားတာကို တွေ့ရမှာပါ။ ဒါ program မှာ ရေးသားထားတဲ့ sort() method ဟာ ကိုယ်တိုင်ဖန်တီးရေးသားထားတဲ့ method တစ်ခု မဟုတ်ပါဘူး။ java.util.\*; package ထဲမှာ ပါဝင်တဲ့ collection Class ထဲမှာပါဝင်တဲ့ sort() method ကို ခေါ်ယူပြီး အသုံးချထားတာပါပဲ။ ဒါဟာ စာဖတ်သူအတွက် JAVA Languages ရဲ features တွေကို ခေါ်ယူ အသုံးပြုစေနိုင်တဲ့ program တစ်ပုဒ်ဆိုရင်လည်း မမှားပါဘူး။

ကဲ – ဒီတော့ ကျွန်ုပ်တော်တို့ ဆက်လက်ပြီး program တည်ဆောက်ကြတာပေါ့။ ကျွန်ုပ်တော်တို့ အခု တည်ဆောက်ထားတဲ့ Application မှာ package file အဖြစ် Class & object တွေကို တည်ဆောက်ပြီးသွားပြီနော်။ ဒီတော့ main () method ပါဝင်တဲ့ program ကို ရေးသားရတော့မယ်ပါ။

ဒီအခါ main () method ပါဝင်တဲ့ program ကို အရာ package file အဖြစ် တည်ဆောက်ထားတဲ့ system's CLASS-PATH ထဲမှာ တည်ဆောက်လို့မရုပါဘူး။ ဒီ package file တွေ တည်ဆောက်တဲ့အကြောင်းကို ရှုံးသင်ခန်းစာမှာ သေသေချာချာရှင်းပြီးခဲ့ပါ၏။

Note: Package file නිර්මාණය කළ විට තෙවැනුවේ system's CLASSPATH තේ මා Giftshop Application Program යාරු program තුදී main ( ) method පිවෙන program file ] ගි ඇඟිල් පිටත දැක්වා ඇත්තා ඇත්තා ඇත්තා Run & Output දැක්වා ඇත්තා Java Compiler නිස්සා error message ගි පෙළාභාපි හෝ සිංහල පිටත ඇත්තා ඇත්තා ඇත්තා Java programmer නිස්සා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා package file නිස්සා CLASSPATH නිස්සා C:\jdk 1.4\com\prefect\ecommerce තේ මා Application program ගි create දැක්වා ඇත්තා ඇත්තා ඇත්තා ඇත්තා ඇත්තා

ကဲ- ဒါဆိုရင် C: \jdk 1.4 > ရွှေအောက်မှာ သီးသနီး folder တစ်ခုကို ဖွံ့ဖြိုးကြရအောင်။ စာဖတ်သုစ္စတိကြုံကြပ်လည်း ဖန်တီး

လို့ ရတယ်နော်။ အဲဒီတော့ C: jdk 1.4\jpro ဆိုတဲ့ System's CLASSPATH ကို တည်ဆောက်ပါ။ အဲဒီ folder ထဲမှာသွားပြီး တော့ အခု ရေးသားထားတဲ့ Listing 2.27 ကိုရေးသာပါ။  
 Giftshop လို့ အမည်ပေးထားတဲ့ Application Program ကို အရင်ရေးသားပြီးမှ program output တွေနဲ့အတူ လေ့လာ သုံး သပ်ကြော်ရတာပါ။ က အာဇာတော့ သေသေချုချု လေ့လာလိုက် ပါ၌။ ဒါ ဒီ program မှာတော့ ရှုံးမှာ တည်ဆောက်ခဲ့တဲ့ own package file တွေကို ပေါ်ယူပြီး အသုံးချိန်ဖို့ ပေါ်ယူ Declare လုပ်ထားတဲ့အချက်၊ ပြီးတော့ Data Input လုပ်ထားတဲ့အချက် စတာတွေကို စာဖတ်သုံးအနေနဲ့ လေ့လာ တွေ့ရှုရမယ်ပါ။

### **Listing 2.27 The Full Text of Giftshop.java**

```
1: import com.preact.ecommerce.*;
2:
3: public class GiftShop {
4:     public static void main (String [ ] arguments) {
5:         Storefront store = new Storefront ();
6:         store .add Item ("co 1 ", "MUG", "9.99", "150");
7:         store.add Item ("co 2 ", "LG MUG ", "12.99", "82");
8:         store .add Item ("co3", "MOUSEPAD", "10.49",
9:                           "800");
10:        store .add Item ("DO1", "T SHIRT", "16.99", "90");
11:        store .sort () ;
12:        for (int i = 0; i < store.getSize () ; i + +) {
13:            Item show = (Item) store.get Item ( i);
14:            System.out.print In ("\n Item ID : " + show.getId () +
15:                                "\n Name : " + show.getName () +
16:                                "\n Retail Price : $ " + show.getRetail () +
17:                                "\n Price: $" + show.getPrice () +
18:                                "\n Quantity: " + show.getQuantity ());
19:        }
20:    }
21: }
```

The output of this program is the following :

Item ID : DO1  
Name : T SHIRT  
Retail: Price : \$ 16.99  
Price: \$ 11.89  
Quantity: 90

Item ID: CO2  
Name : LG MUG  
Retail Price : \$ 12.99  
Price : \$9.09  
Quantity: 82

Item ID : CO1  
Name : MUG  
Retail Price : \$ 9.99  
Price : \$6.99

## ကွန်ပူးတာသင်ခန်းစာ

<p>Quantity : 150</p> <p>Item ID : CO3</p> <p>Name : MOUSEPAD</p> <p>Retail Price : \$ 10.49</p> <p>Price : \$ 5.25</p> <p>Quantity : 800</p> <p>အပေါ်က ရေးသားပို့ချကဲ့ program ကို Run လုပ်လိုက်ရင် output ကို အထက်ပါအတိုင်း တွေ့ရမှာပါပဲ။ ဒီတော့ program code_line တွေကို သေသေချေချာ လေ့လာ သုံးသပ်ကြည့်ကြရ အောင်လား။</p> <p><b>Analysis</b></p> <p>၁။ ဒီ program ကို လေ့လာကြည့်မယ်ဆိုရင် အရင် လေ့လာ ကြည့်ရမှာက Giftshop Application program ရဲ့ လုပ်ဆောင်ပေးနိုင်တဲ့ features တွေဟာ ဘာတွေလဲဆိုတာ ကို အရင် သုံးသပ်ကြည့်ရမှာပါပဲ။ ဒါတွေကတော့</p> <ul style="list-style-type: none"> <li>◆ Add Items to it</li> <li>◆ Sort the item by sale price</li> <li>◆ Loop through a list of items</li> <li>◆ Display information about each one</li> </ul> <p>အစရှိတဲ့ အချက်တွေပဲ ဖြစ်ပါတယ်။ ဒီ program ရဲ့ လိုင်း နံပါတ် (၁)မှာ ကွန်တော်တို့ရှေ့မှာ ဖန်တီးခဲ့တဲ့ package file ဖြစ်တဲ့ com.prefect.ecommerce.*; ကို import လုပ် ခေါ်ယူ အသုံးချထားပါတယ်။ ဒါကြောင့် စာဖတ်သူ အနေနဲ့</p> <pre>import com.prefect.ecommerce.*;</pre> <p>ဆိုပြီး တွေ့ရတာပါပဲ။ အဲဒီနောက်မှာမှ Giftshop ဆိုတဲ့ primary class ကို ဖန်တီးပါတယ်။ package file အဖြစ် ရေးဖွဲ့စည်က တည်ဆောက်ခဲ့တဲ့ storefront class ကို object တစ်ခုအနေနဲ့ ခေါ်ယူ အသုံးချဖို့အတွက် လိုင်း နံပါတ် (၅) ခေါ်ယူထားတာကို တွေ့ရမှာပါ။</p> <p>လိုင်းနံပါတ် (၆) မှာတော့ အဲဒီ class ရဲ့ member method (၈) တွေကို ခေါ်ယူ အသုံးချပြီးတော့ Data -Input လုပ်ထားတာကို တွေ့ရမှာပါ။ အဲဒီတွေကတော့ လိုင်းနံပါတ် (၆) ကနေ လိုင်း နံပါတ် (၉) အထိ ရေးသားထားတဲ့</p> <ol style="list-style-type: none"> <li>6. store.add Item ("CO1", "MUG", "9.99", "150");</li> <li>7. store.add Item ("CO2", " LG MUG", "12.99", "82");</li> <li>8. store.add Item ("CO3", "MOUSEPAD", "10.49", "800");</li> <li>9. store.addItem ("DO1", "T SHIRT", "16.99", "90");</li> </ol> <p>Double quote " ကြားထဲမှာရေးထားတဲ့ Input_Data တွေနဲ့ ပတ်သက်ပြီးတော့ ရှင်းပြရရင်တော့</p> <p>"CO1" ဆိုတာ Item ID အနေနဲ့ output ပြပေးရမယ့် Data "MUG" ဆိုတာ Product Name အနေနဲ့ output ပြပေးရမယ့် Data</p>	<p>"9.99" ဆိုတာ Retail Price အနေနဲ့ output ပြပေးရမယ့် Data</p> <p>"150" ဆိုတာ Quantity အနေနဲ့ output ပြပေးရမယ့် Data</p> <p>တွေပဲ ဖြစ်ပါတယ်။ ဒါဟာ store ဆိုင်တစ်ဆိုင်မှာ ရောင်းရန် ရှိတဲ့ ကုန်စည်တွေကို ဖွော်လုံးအမျိုးအစား၊ အရေအတွက်၊ ကျသော တန်ဖိုး ID code စတာတွေနဲ့ Data Input လုပ် ဖန်တီးပြီးတော့ program တည်ဆောက်ထားတဲ့ သဘောတရားပါပဲ။ Application project တစ်ခုအသွင် ဖန်တီးထားတဲ့ ဒီ program ငယ်လေး စာဖတ်သူအတွက် JAVA Language နဲ့ ပတ်သက်တဲ့ အတွေ့ အကြိုတွေကို ပေးစွမ်းနိုင်လိုပ်မယ်ဆိုတာကို ယုံကြည့်ကြပါတယ်။ JAVA ရဲ့ Professional Certification Exam; တွေမှာလည်း ဒီ program type ဟာ အမြဲတမ်း ပါဝင်နေပါတယ်။</p> <p>၂။ လိုင်းနံပါတ် (၁၂) မှာတော့ Data တွေကို output အဖြစ်ပြပေးနိုင်ဖို့ fa looping နဲ့ တွဲဖက်ပြီးတော့ ရေးသားထားတာကို စာဖတ်သူအနေနဲ့ တွေ့ရမှာပါ။ အဲဒီ looping Statement နဲ့ ပတ်သက်ပြီးတော့ ဘာမျှ အကျယ်ချုံပြီး ရှင်းပြစ်ရ မလိုအပ်ပါဘူး။ Data တွေရဲ့ သို့လောင်ထားတဲ့ Input လုပ်ထားပုံပေါ်ကို မှတ်လုပ်ပြီးတော့ getSize() method ကို အသုံးပြထားတာကို မမေ့နဲ့နော်။ ဒီအချက်ဟာ အရေးပါဝါတယ်။ လိုင်းနံပါတ် (၁၃) မှာတော့ System.out.println() Statement ကို အသုံးပြပြီးတော့ output ကို တွက်ထုတ်ထားပါတယ်။ လေ့လာကြည့်မယ်ဆိုရင်</p> <pre>"In Item ID :" + show.getItemId() + "In Name : " + show.getName() + "In Retail Price : \$" + show.getRetail() + "In Price : \$" + show.getPrice() + "In Quantity :" + show.getQuantity() );</pre> <p>ဒီ close_bracket ဟာ System.out.println() method က အပိုတ် Close -bracket ပါပဲ။</p> <p>ဒီနေရာမှာ Input-Data အနေနဲ့ ထည့်သွင်းထားတဲ့ Data တွေကို Item ID, Name, Retail Price \$, Price \$. Quantity စတာတွေနဲ့ ယဉ်တဲ့ ဖော်ပြန့်အတွက် ရေးသားပါတယ်။ Data အားလုံးကို တွက်ထုတ် ပြသပေးနိုင်ဖို့အတွက် looping statement ကို ရေးသားရာမှာ ပါဝင်တဲ့ သတိပြုရမယ့် အချက်ကတော့ getSize() method ကို အသုံးပြထားခြင်းပါပဲ။ ဒါကြောင့် လိုင်းနံပါတ် (၁၃) မှာ</p> <pre>For (int i=0 ; i &lt; store.getSize() ; i ++ )</pre> <p>လို့ ရေးသားထားတာပါပဲ။ ကဲ ဒါဆိုရင် program code line</p>
--	--

## ကွန်ပျူတာသင်ခန်းစာ

မှာပါဝင်တဲ့ statements တွေကို နားလည်ပြီဖော်။ ဒါဆိုရင် program ပဲ output ကို လေ့လာကြည့်ရအောင်လား။ Output ထွက်လာပဲကို စာဖတ်သူအနေနဲ့ လေ့လာကြည့်ပါ။

The output of this program is the following :

Item ID: DO1

Name: T SHIRT

Retail Price : \$ 16.99

Price : \$ 11.89

Quantity : 90

Item ID: CO1

Name : LG MUG

Retail Price : \$ 12.99

Price : 59.09

Quantity : 82

Item ID : CO2

Name : MUG

Retail Price : \$9.99

Price : \$6.99

Quantity: 150

Item ID: CO3

Name : MOUSEPAD

Retail Price : \$ 10.49

Price : \$ 5.25

Quantity: 800

ဒီအခါမှာတော့ အပေါ်မှာ မြင်ရတဲ့အတိုင်း output ဟာ လာပြမှာပါ။ ဒါဟာ ကျွန်တော်တို့ ကြိုးစား အားထုတ်ခဲ့ရတဲ့ program လေးရဲ့ အသီးအပွင့်ပါပဲ။ ဒီ output ထွက်လာပဲ ကျွန်တော်တို့အနေနဲ့ Package file တွေ ဖန်တီးရေးသားရတယ်။ အဲဒီကမှတစ်ဆင့် class တွေကို object အသွင် ဖန်တီးတယ်။ လိုအပ်တဲ့ method တွေ၊ statements and looping structure တွေကို ဖန်တီးရပါတယ်။

အခု တည်ဆောက်လေ့လာခဲ့တဲ့ program ကို စာဖတ်သူ ကိုယ်တိုင် computer ထဲမှာ တည်ဆောက်ကြည့်ပါ။ လိုအပ်တဲ့ package file တွေကို ဖန်တီးတယ်ဆိုတဲ့ features တွေကို လည်း ထပ်မံ ဖြည့်သွင်းနိုင်ပါတယ်။ ဒီလို စွမ်းဆောင်ရည် features တွေကို တွေ့ဖြန့်နေအောင်လို့ အခုလို package file နဲ့ program တည်ဆောက်တာတွေကို စာဖတ်သူဘက်က စနစ် တကျ လေ့လာ နားလည်ဖို့လည်း လိုအပ်ပါလိမ့်မယ်။

Note : အရေးကြီးတဲ့ သတိပြုရမယ့်အချက်ကတော့ package file တွေ ဖန်တီးတဲ့ system's CLASSPATH မှာ main ( ) method ပါဝင်တဲ့ program ကို တည်ဆောက်လို့

မရနိုင်ပါဘူး။ ဒါအပြင် Data Input လုပ်တဲ့ သွေးတဲ့ အခါမှာလည်း

Item ID:

Name:

Retail Price : \$

Price: \$

Quantity:

တဲ့ အချက်အလက်တွေနဲ့ ညိုယူပြီး ဖြည့်သွင်းရမှာပါ။ အကယ်၍ Data\_Input လုပ်စဉ်မှာ Data တစ်ခုရဟာ ဖြည့်သွင်းရှုံးရှင် မေးလျှော့ရင် ကျွန် Data တွေမှာလည်း အစီအစဉ်တာကျ မဖြစ်လာနိုင်ဘဲ Error အနေနဲ့ လာပြနိုင် ပါတယ်။ ဒီအချက်တွေဟာ စာဖတ်သူဘက်က ရရှိပြု မှတ်သားရမယ့် အချက်အလက်တွေပါပဲ။

ဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ Own's package file တွေ တည်ဆောက်တဲ့အပိုင်းကို သင်ခန်းစာအပိုင်း ခွဲခြားပြီး တွေ့ဖြတ်ပေါ်တဲ့အတွက် နားလည် သဘောပေါ်သွားလောက် ပါပြီ။ လိုအပ်ရင် package file နဲ့ ပတ်သက်ပြီး စတင် ပို့ချုပ်တဲ့ ရေးသားပို့ချုပ်သားတာပါပဲ။

ကျွန်တော်အနေနဲ့ 2.6 Installing, Using the JAVA 2 Software Development kit ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် ပို့ချုပ်သားပြီး အဲဒီသင်ခန်းစာအပိုင်းကတော့ JAVA Software ကို Installing လုပ်တဲ့ Step By Step Guide ကို ရေးသားပို့ချုပ်သားတာပါပဲ။

JAVA Language ကို အသုံးပြုတဲ့အခါမှာ အခုမှ programming language ကို စတင်လေ့လာတဲ့သူတွေအနဲ့ Install လုပ်ပြီး အသုံးပြုမယ်ဆိုရင် အဆင်သိမ်မပြေပါဘူး။ ဒါဟာ programmer တော်တော်များများ တွေ့ကြိုရတဲ့ ပြဿနာပါပဲ။

တုဖို့မှာလည်း JBuilder တို့လို IDE ကိုသုံးပြီး program ရေးမလား။ Sun ONE studio ကိုပဲ အသုံးချုပ်မလား။ ဒါမှမဟုတ် text editor တွေကိုပဲ အသုံးပြုပြီးတော့ program ရေးမလားဆိုတာကို ခေါ်းရှုံးရှင်အောင် စဉ်းစားရမှာပါ။

က Text editor ကို သုံးမယ်ဆိုရင်လည်း Plain text လို့ခေါ်တဲ့ Start - All program - Accessories- Notepad ကိုပဲ သုံးမလား။ ဒါမှမဟုတ် wordpad/ Microsoft word အစရိတ်တွေကိုပဲ သုံးမလား။ နားလည်ကျော်ကြားလိုတဲ့ emacs, pico, vi; စတာတွေကို သုံးမလား စသဖြင့် ရေးတွေ့နိုင်ပါတယ်။

ဒါကြောင့် JAVA Developer Guide ကို သင်ယူနေစဉ်မှာ အလွယ်တကူ လေ့လာ သင်ယူနိုင်ဖို့အတွက် 2.6 Installing , Using the JAVA 2 Software Development kit ဆိုတဲ့ အခန်းကဏ္ဍကို ပို့ချုပ်သားပါတယ်။ လေ့လာကြည့်ပါ။

KYAW ZAYAR LAY [ IT ENGINEERING ]

## JAVA DEVELOPER GUIDE

### 2.6 INSTALLING THE JAVA 2 SOFTWARE DEVELOPMENT KIT

Java Language ဖြစ်တဲ့ Software Development Kit ကို Install လုပ်ဖို့ဟာ သာမန် user တွေအဖို့တော့ (ဥပမာအနေနဲ့ ပြောရရင် C & C++, VC++, Visual Basic အစရိတဲ့ language တွေမှာလို) user Defined မဖြစ်ပါဘူး။

Program တစ်ခုကို ဖန်တီး ရေးသားလိုအတွက် code-line တွေကို ရေးသားပြီးတော့ compile လုပ်မယ်။ Run လုပ်တဲ့ သဘောတရားကတော့ C++ နဲ့ အနည်းငယ် ဆင်တူပါတယ်။

အမိကအနေနဲ့ ကျွော်မြှေးနားတာကတော့ Java 2 software Development Kit ကို Install လုပ်ပြီးရင် Text Editor တွေကို အသုံးပြုပြီးတော့ Program တွေကို ဖန်တီးမှာလား။ ဒါမူ မဟုတ် Java Programming tools တွေပါလို့ နာမည် ကျော်ကြားလှတဲ့ Borland JBuilder, IntelliJ IDEA, IBM visual Age အစရိတဲ့ IDE tools တွေကိုပဲ အသုံးပြုမှာလား။

ဘာပဲပြောပြော ကျွန်တော်ကတော့ Java Language ကို အသုံးပြုပြီးတော့ program ကို ဖန်တီးရာမှာ Text Editor ကို အသုံးပြုရတာကိုပဲ ပိုပြီးတော့ စိတ်တိုင်းကျပါတယ်။ ဘယ် Text Editor ကိုမဆို လိုအပ်သလို အသုံးပြုလို့ ရနိုင်တာကိုး။

ဒီတော့ Java ရဲ့ Software Development Kit ကို Install လုပ်ပြီးတော့ Program တွေ ဖန်တီး ရေးသား လေ့လာကြတာပေါ့။

**Note :** အခါ သင်ခန်းစာကို ရေးသားလို့ချစဉ်က Java language အနေနဲ့ J2SDK 1.4 ကိုပဲ Install လုပ်ထားပေမယ့် နောက် version အသစ်တွေဖြစ်တဲ့ Java Language version တွေကို setup လုပ်တဲ့ နည်းလမ်းဟာ အတူတူပါပဲ။ Setup လုပ်ရာမှာ စနစ်တကျ configuration လုပ်တတ်ဖို့ပဲ လိုအပ်တာပါ။

#### ◆ Choosing a Java Development Tool

စာဖတ်သူအပါအဝင် ကျွန်တော်တို့ဟာ Microsoft Windows, ဒါမှမဟုတ် Apple Mac OS system ၊ ဒါမှမဟုတ် Linux အစရိတဲ့ OS တစ်ခုခုကို အသုံးပြုနေတယ် ဆိုကြပါစို့။ ဒါဆိုရင် စာဖတ်သူရဲ့ Computer ထဲမှာ Java Interpreter ကို

Installed လုပ်ထားပြီးသား ဖြစ်နေမှာပါ။ ဒါဟာ Java Program တွေကို Run-Out လုပ်နိုင်တဲ့ အဆင်သင့် အနေအထားပါပဲ။

ဒါဟာ ဘာကြောင့်လဲဆိုတော့ လွန်ခဲ့တဲ့ နှစ်ပေါင်းအနည်းငယ်ကတော့ Java ရဲ့ Interpreter ဟာ Microsoft Internet Explore, Netscape Navigator အစရိတဲ့ Web Browsers တွေအထဲ built-in အနေနဲ့ပါရှိလာတာပါ။ ဒါဟာ ဘာကြောင့်လဲဆိုတော့ ဒီနေ့စောင့်ရဲ့ Web Programs အများစုံဟာ Java Language နဲ့ တည်ဆောက်ထားတာကြောင့်ပါပဲ။

ဒီချုပ်ဟာ Java Programs တွေကို Run Out လုပ်နိုင်ဖို့ လေ့လာလို့ ရတယ်လို့ပဲ ပြောတာပါ။ တကယ်လို့ စာဖတ်သူဟာ Java Program တွေကို Develop လုပ်မယ်ဆိုရင် interpreter ပါရှိနေချုပ်လောက်နဲ့ မလုံလောက်တော့ပါဘူး။ ဒီတော့ ကျွန်တော်တို့ဟာ Java Compiler and Other tools တွေတို့ပါ လိုအပ်လာပါပြီ။ ဒါဟာ Java Program .code တွေနဲ့ Program တွေကို Create, run and test program လုပ်နိုင်ဖို့အတွက်ပါပဲ။

ဒါဆိုရင်တော့ ဒီချုပ်အထိ စာဖတ်သူ နားလည်သဘောပေါက်လောက်ပါပြီဖော်။

ဒီနေ့ရာမှာ အကျယ်ချုပြီးတော့ ရှင်းပြရင်တော့ Java ရဲ့ Software Development kit ထဲမှာ compiler, Interpreter, Debugger, File archiving Program နဲ့ serval other program တွေ ပါဝင်ပါတယ်။

Java ရဲ့ Software Development kit ဟာ အခြားသော Software Development tools တွေထက် ပိုပြီးတော့ ရှိုင်းလွယ်ကြပါတယ်။ GUI လိုပေါ်တဲ့ Graphical User Interface လည်း မလိုအပ်သလို အလွယ်တကူ Text Editor ကို အသုံးပြုပြီးတော့ Program တွေကို တည်ဆောက်ဖန်တီးလို့ ရပါတယ်။

ဒါဟာ အခြားသော programmer တွေထက် Java Program တွေရဲ့ အရှိုးရှင်းဆုံး အားသာချက်နဲ့ Windows Interface နဲ့ ရင်းနှီးကျေမ်းဝင်ပြီးသား သာမန် user တွေအဖို့လည်း Java Language ကို လေ့လာ သင်ယူရာမှာ ပိုမိုဖြန်ဆန်စွာ နားလည်သဘောပေါက်စေနိုင်တဲ့ အမိက အချက်ပါပဲ။ ကဲ-ဒါဆိုရင်တော့ ဆက်လက် လေ့လာကြတာပေါ့။

ကျော်တော်တို့ဟာ Java ရဲ့ Software Development kit ကို အသုံးပြုမယ်ဆိုရင် commands ထွေကို text Prompt အနေနဲ့ type ရှုကြပြီးမှ စေခိုင်းရတာပါ။

MS DOS, Linux and Loxix users တွေအနေနဲ့ကတော့ ဒီလိုမျိုး text prompt တွေ ရှိကြပိုး စေခိုင်းရတာကို family ဖြစ်ပြီးသားပါ။ အသုံးသော Programmer တွေအနဲ့ကတော့ Java Language နဲ့ မနီးစပ်ခဲ့ရင်၊ C & C++ + ကို မလေ့လာခဲ့ဖူး ရင်တော် MS-DOS (or) Command .com ကိုထော့ လေ့လာဖူး ကြမှာပါ။ ဒါကြောင့် ဒီ software Development kit ကို comfortable ဖြစ်တယ်လို့ ပြောလို့ရပါတယ်။

အခါး Installing လုပ်မှာက Java ရဲ့ version 1.4 ကို Installing လုပ်ပြမှာပါ။ ဒါဟာ ကျော်တော်တို့ အခါး လေ့လာသင်ယူ နေတဲ့ Tutorial Programs တွေကို Development လုပ်နိုင်နိုင်ပါပဲ။

#### ◆ Installing the Software Development Kit

Java ရဲ့ Software Development kit ကို Web Site ကနေတစ်ဆင့် download လုပ်ယူမယ်ဆိုရင်တော့ Sun's Java Web Site ဖြစ်တဲ့ <http://java.sun.com> ကနေ အလွယ်တကူ Download လုပ်ပြီးတော့ ရယူနိုင်ပါတယ်။

အဲဒီ Web Site မှတော့ Java ရဲ့ versions အမျိုးမျိုးကို အလွယ်တကူ Download လုပ် ရယူနိုင်ပါတယ်။ နာမည်ကြီး Sun ONE studio and other products တွေဟာလည်း ဒီ language အတွက်တော့ ဆက်နှံယူနေတဲ့ လိုအပ်ချက်တွေပါပဲ။

Java SDK 1.4 ကို Installing လုပ်ပယ်ဆိုရင်တော့ သင့်တော်တဲ့ Platform တွေကတော့

- Windows 95, 98, Me, NT (with service Pack 4), 2000 and XP
- Solaris SPARC and Intel
- Linux

အစရိတ် Windows OS တွေထဲက Windows Version တစ်ခုးမျိုးကို ရယူထားမယ်ဆိုရင် ဖြစ်နိုင်ပါတယ်။ Java ရဲ့ SDK ကို အသုံးပြုမယ်ဆိုရင်တော့ စာဖတ်သူရဲ့ computer မှာ ရှိရမယ့် အနိမ့်ဆုံး လိုအပ်ချက်ကတော့ –

- Pentium Processor that is 166 MHz or faster
- 32 MB of Memory
- 70 HB of Free disk space

ဖြစ်ပါတယ်။

အဲ စာကယ်လို့ စာဖတ်သူရဲ့ computer ထဲမှာ Java Applets တွေကို ဖန်တီး လေ့လာပယ်ဆိုရင်တော့ –

- at least 48 MB of Memory ရှိရမယ်လို့ Sun ကနေ recommends လုပ်ထားပါတယ်။

**Tip :** Apple ရဲ့ Maentosh ကို ဖက်တွယ်ထားသူတွေအတွက် လည်း SDK 1.4 ဟာ သင့်တော်ပြီးသားပါ။

<http://download.apple.com/java/> မှာ Mac OS x Developers Tools တွေအဖြစ် အဆင်သင့် Support ပေးထားပြီးသားပါ။ SDK 1.3 နဲ့ Other java development tools တွေဟာ fully depended ဖြစ်ပြီးသားပါ။

ဒီနေရာမှာ စာဖတ်သူဟာ SDK ကို Install မလုပ်ခင်မှာ SDK ရဲ့ version number ကို third number အဖြစ် တွေရမှာပါ။ ဘယ်နေရာမှာလဲဆိုတော့ 1.4 ရဲ့ နောက်မှာပေါ့။ ဥပမာ-SDK 1.4.1 အနေနဲ့ပေါ့။ Sun က Version အသစ်ကို New version အနေနဲ့ release လုပ်တိုင်း periodically issues လုပ်ပါတယ်။ အဲဒီအခါမှာ Update Version ဟာ main version ရဲ့ အနောက်မှာ extra period အနေနဲ့ ပါဝင်လာပါတယ်။ တစ်နည်းပြောရရင်တော့ extra period and digit အနေနဲ့ပေါ့။ ဥပမာ 1.4.1, 1.4.2 or higher အစရိတ်ဖြင့်ပေါ့။

**Tip :** Web site ကနေပဲ download လုပ်လုပ်၊ Install CD ထဲကနေပဲ Install လုပ်ယူ အောက်ပါ version နှစ်ခု စလုံးကို ရွေးချယ်စရာ မလိုပါဘူး။ တစ်ခုခုကို ရွေးချယ် Install လုပ်နဲ့ လုံလောက်ပါတယ်။ အဲဒီ version နှစ်ခု ဟာ အတူတူပဲမို့ပါ။

- The Java 2 Runtime Environment, Standard Edition, Version 1.4,
- The Java 2 Software Development kit, Standard Edition Source Relase Web Site ကနေပဲ download လုပ်ယူဆိုရင်တော့ download page ကို အောက်ပါ web address အတိုင်း တိုက်ရှိက်သွားနိုင်ပါတယ်။

<http://java.sun.com/j2se/1.4>

စာကယ်လို့ စာဖတ်သူဟာ Software Development kit ကို setup လုပ်မယ်၊ download လုပ်ပြီး Install Program ကို run မယ်ဆိုကြပါနဲ့။ (တစ်နည်းပြောရရင် Install CD က တစ်ဆင့် Setup လုပ်မယ် ဆိုကြပါနဲ့။)

အဲဒီ Setup Program file ဟာ 35-40 MB အရွယ်အစား ရှိပါတယ်။ ဒီတော့ စာဖတ်သူရဲ့ computer ထဲမှာ လိုအပ်နိုင်တဲ့ Memory Free Space အရွယ်အစားကို တွက်ချက်လို့ ရှိရမယ်ပြီး။ Caution: ကျော်တော်ကတော့ Install လုပ်ရမှာ လွယ်ကူအောင် Sam Publications က ထုတ်ဝေတဲ့ Java 21 days မှာပါဝင်တဲ့ Install CD နဲ့ Demonstrate လုပ်ပေးထားပါတယ်။ စာဖတ်သူ တို့ အလွယ်တကူ ရှာဖွေလို့မရရင် copy ကူးယူ အသုံးပြုနိုင် အောင် သင်ခန်းစာနဲ့အတူ ဂျာနယ်တိုက်ကို ပေးပို့ပေးထားပါတယ်။

ကွန်ပျူးတာသင်ဆန်းစဉ်

- ◆ Windows Installation on Windows 95, 98 or Me/  
on Windows NT, 2000, or XP

အလွန်အရေးကြီးတဲ့အချက်ကို သတိပေးချင်တာက SDK 1.4 ကို Installing မလုပ်ခင်မှာ အခြားသော Java Development tools တွေကို installed လုပ်ထားတာမျိုးဟာ သင့်ရဲ့ computer ထဲမှာ မရှိရပါဘူး။ တကယ်လို့ Java Programming tool ဟာ တစ်ခုထက်မက သင့်ရဲ့ Computer ထဲမှာ Install လုပ်ထားခဲ့ရင် SDK 1.4 ကို Install လုပ်ဖြစ်ထား၏ configuration Problems တွေဟာ သင့်ရဲ့ system ထဲမှာဖြစ်ပွားလာနိုင်ပါတယ်။

କି କାଣ୍ଡରୁ Computer CD Drive ଯେହି Install CD ଡେଟ୍ସଲ୍‌ଟିକ୍‌ର୍ ଓେକର୍ ପି Auto-Run Splash Screen ତାଙ୍କରିଲା ମୂଳି ପି ତାତେବେଳୁ କମ୍ପ୍ୟୁଟର ଯେହି ମୂଳ ଫିଗ୍‌୨.୬.୧ କି ଦୁଆରୀର୍ଦ୍ଦିତ ଦେଖାଇଲାପି ॥

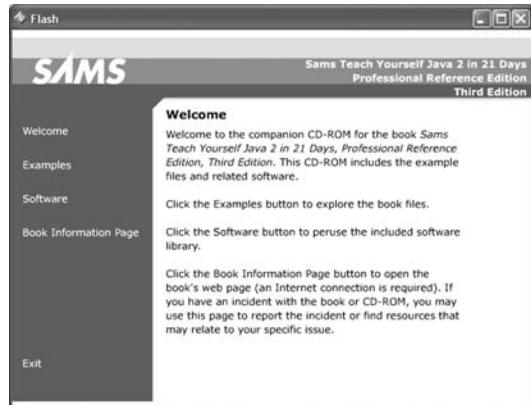


Figure 2.6.1 The 1<sup>st</sup> Step of the Install J2SDK

အေဒီ Splash ထဲက Mouse Pointer တင်ထားတဲ့ software ဆိုတဲ့နေရာကို နိုင်လိုက်ပါ။ အောက်မှာပြထားတဲ့ Figure 2.6.2 က ပုံစံပျိုးဖြစ်သွားပါလိမ့်မယ်။

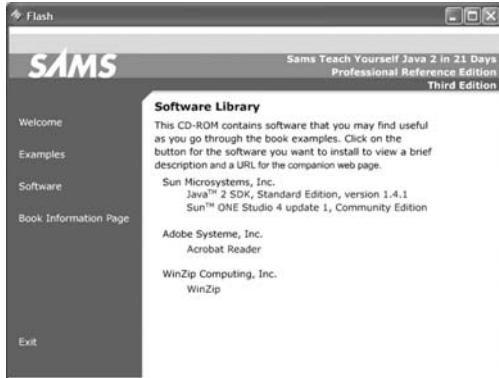


Figure 2.6.2 The 2<sup>nd</sup> Step

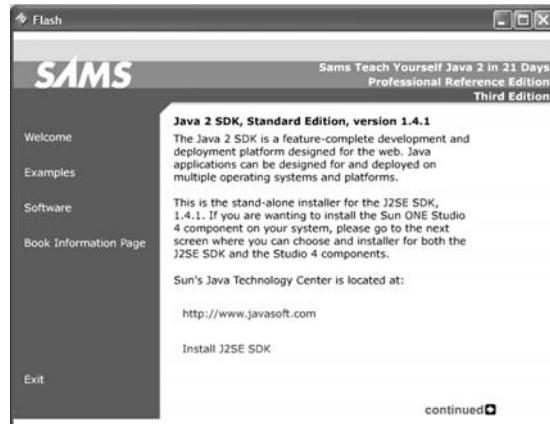


Figure 2.6.2 The 2<sup>nd</sup> Step

ဒီအခါမှာ Software library ဆိုတဲ့ Choice Menu အောက်က Java™ 2 SDK , Standard Edition, Version 1.4.1 ဆိုတဲ့ Menu ကို t-click နိုင်ပါ။ ဒီအခါမှာ Splash Screen ရဲ့ ပုံစာဟာ Figure 2.6.3 က ပုံစံအတိုင်းဖြစ်လာမှုပါ။

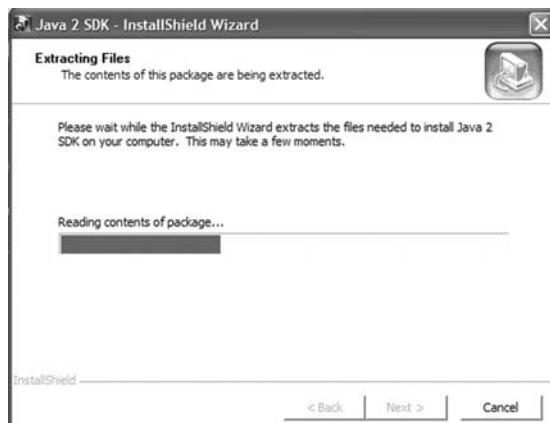


Figure 2.6.3 The 3<sup>rd</sup> Step



۶۹

အဲဒီအခါမှာ Install J2SE SDK ဆိုတဲ့ Menu ကိုပဲ click လုပ်ပါ။ ဒီအခါမှာ Java 2 SDK Installshield Wizard တက်လာပါပြီ။ Figure 2.6.4 ရှိကြည့်ပါ။ ခဏဖော်ပါ။



Figure 2.6.4 The 4<sup>th</sup> Step

အဲဒီအခါမှာ Installshield Wizard ဟာ software ကို Installing လုပ်ဖို့အတွက် Step-By-Step Guide အနေနဲ့ ညွှန်ပြပေးမှာပါ။

စာဖတ်သူဟာ sun's ရဲ့ terms and conditions တွေကိုလိုက်နာပါမယ်လို့ accept လုပ်တာနဲ့တစ်ပြင်နက် Figure 2.6.5 မှာ ပြထားတဲ့ အတိုင်း Choose a Distination folder ဟာ ပေါ်လာမှာပါ။ အောက်မှာ Figure 2.6.5 ကို ဖော်ပြထားပါတယ်။ အဲဒီ Installshield Wizard မှာပါတဲ့ Next > Button ကို နှိပ်လိုက်ပါ။

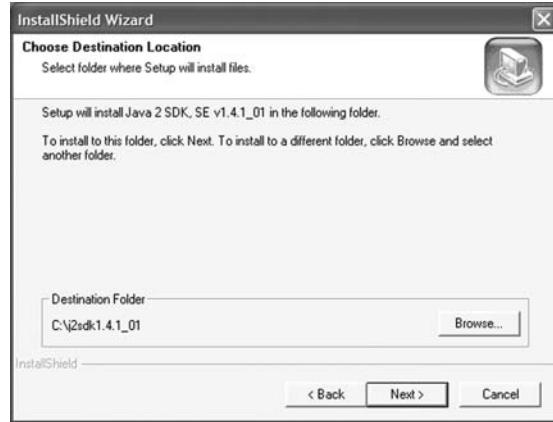


Figure 2.6.5 The 5<sup>th</sup> Step

အဲဒီအခါမှာ Software Development kit ကို Install လုပ် နိုင်ရှိ selecting components Dialog Box ဟာ Figure 2.6.6 မှာ ပြထားတဲ့ အတိုင်း ပေါ်လာမှာပါ။



Figure 2.6.6 The 6<sup>th</sup> Step

အဲဒီ Dialog Box မှာပါဝင်တဲ့ Install components တွေ ကို ရှင်းပြရမယ်ဆိုရင်တော့ -

- Program Files

executable program လို့ခေါ်တဲ့ (.exe) Files တွေကို ဖုန်တိုးမယ်။ Compile လုပ်ပြီးတော့ Java Projects တွေ ဖုန်တိုးမှုပို့အတွက် သက်ဆိုင်တဲ့ Components ပါပဲ။

- Native Interface Header Files

Java Programs Files တွေကို ရေးသားအသုံးပြုရမှုမကဘဲ အခြား programming languages တွေနဲ့ပါ ပေါင်းစပ် ရေးသားပြီးတော့ အသုံးပြုနိုင်စေနို့ ပါဝင်တဲ့ Components tools ပဲ ဖြစ်ပါတယ်။

- Demos

ဖန်တီးရေးသားပြီးသား Java 2 Programs ရဲ့ Source Code files တွက် Run and Test လုပ်ပြီးတော့ လေ့လာနိုင်ဖို့ ပါဝင်တဲ့ Components ပါပဲ။ ဒါ Components ဟာ စာဖတ်သူ အတွက် Java Languages ကို ပိုမိုပြီးတော့ ကျမ်းဝင် ရင်းနှီးစေ နိုင်ပါလိမ့်မယ်။

- Java Sources

Java ና Classes ገዢ (ተትቅኝነት በJava የJava Sources Code File ገዢበት) ታስቦ Components እና Java 2 Class Library ደንብዎች Components ሲያስተካክል။

- Java 2 Runtime Environments

ဒါဟာ Selecting Components တစ်ခုချင်းစိတိ အသေးစိတ် ရှင်းပြပေးတာပါပဲ။ သက်ရွိုင်ရာအခန်းကဏ္ဍအလိုက်မှာတော့ ထပ်မံ ရှင်းပြပေးပါဉိုးမယ်။

ဒီနေရာမှ Default အတိုင်း ထားပြီးတော့ Installation ဆက်လုပ်စေချင်ပါတယ်။ ဒီ Components အားလုံးကို ရယူချင် တယ်ဆိုရင်တော့ 72 MB of Free space on Herdisk ဖြစ်ရ ပါမယ်။ ဒီ Memory space ပမာဏဟာ မပြောပလောက်တဲ့ ပမာဏပါပဲ။ Components အားလုံးကို လက်ခံသင့်ပါတယ်။ ဘာကြောင်လဲဆိုတော့ –

- Demos
  - Java 2 Runtime Environments

လိုအပ်တဲ့ Components နှစ်ခါးဟာ extermely useful လို့ ပြောရလောက်အောင် မရှိမဖြစ် အလွန်တရာ အရေးပါ အသုံးဝင်လှတဲ့ components တော့ ဖြစ်ထိပါပဲ။

## တစ်ဖက်က ကြည့်ပို့ရင်လဲ

- Native Interface Header Files

က Software Development kit ကို လေ့လာဖြီးရင်တော့  
အဲဒီပါဝင်တဲ့ components တွေကို သတိပြုမှာပါ။ ဒီအခါမှာ  
အဲဒီ Install Shield Wizard(တစ်နည်းလျှော့ရှင်တော့ Seleting  
Components Dialog Box)မှာ ပါဝင်တဲ့ Next > Buttom ကို  
Click နိုင်ပါ။

အဲဒီအခါမှာ Java ပုံ၊ Plug- In with web Browser ကို  
စာဖတ်သူရဲ့ System ထဲမှာ ထည့်သွင်း Install လုပ်ပေးမယ့်  
Install Shield Wizard ကို ဆက်ပြပေးပါလိမ့်မယ်။ အဒီကို  
တော့ အောက်မှာဖော်ပြထားတဲ့ Figure 2.6.7 က အတိုင်း တွေ့မြင်  
ရမှာပါ။ လေ့လာကြည့်ပါ။

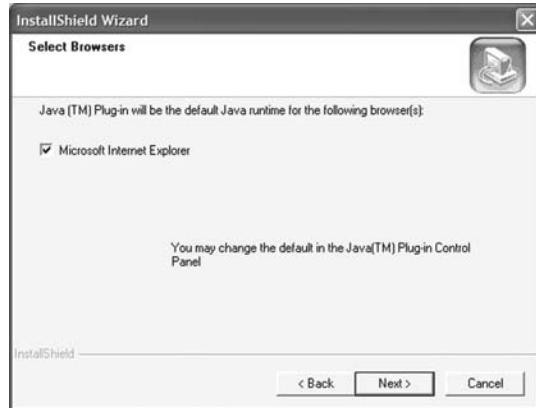


Figure 2.6.7 The 7<sup>th</sup> Step

ဒီအခါ Windows Versions ကို သုံးစွဲသူ user တွေ အတွက်ကတော့ Microsoft Internet Explorer ကိုပဲ Plugin အနေနဲ့ ထည့်သွင်းပေးမှာပါ။ ဒီတော့ Next > Button ကို Click လုပ်ပါ။ ဒါဆိုရင်တော့ Install Shield Wizard ရဲ့ last Dialog Box ဖြစ်တဲ့ Figure 2.6.8 မှာပြထားတဲ့ Dialog Box ပေါ်လာရင်တော့ အဲဒီ Dialog Box မှာ ပြထားတဲ့အတိုင်း finish > Button ကို click လုပ်ဖြစ်ပြင် Java 2 SDK ကို စာဖတ်သူရဲ့ System's မှာ လုပ်ဆောင်စေမယ့် Install Shield Wizard ကို အဆုံးသတ်နိုင်ပါပြီ။

Figure 2.6.8 ကို သေသေချာချာ လွှဲလာကြည့်ပါ။ ဒါမှသာ  
Install လုပ်တဲ့အခါမှာ အက်အေးမရှိ လုပ်ဆောင်နိုင်မှာဖြစ်တယ်။



Figure 2.6.8 The 8<sup>th</sup> Step

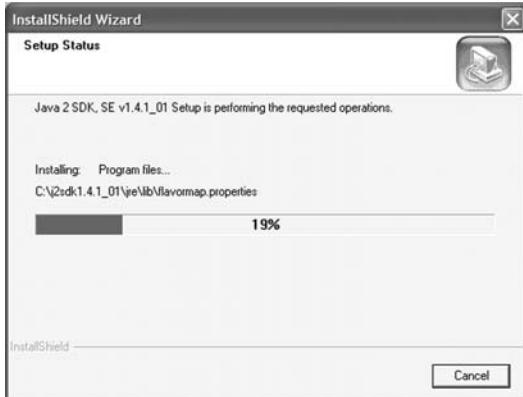


Figure 2.6.8 The 8<sup>th</sup> Step

ကဲ– ဒါဆိုရင်တော့ စာဖတ်သူရဲ့ computer ထဲမှာ SDK 1.4 ဟာ Install step ဟာ Complete step ကို ရောက်နေပါပြီ။ ဒါပေမယ့် သတိပြုရမှုက Application Software တွေလို ဥပမာ Word, Excel, Page Maker, Photoshop အစရိတဲ့ အသုံးချု Software တွေလိုမျိုး၊ အခြားသော Programming Languages တွေဖြစ်တဲ့ C & C++, VC++, Visual Basic အစရိတဲ့ Languages တွေမှာလိုမျိုး။ Install လုပ်ပြီးပြုဆိတာနဲ့ တစ်ခါ တည်း used လုပ်လို မရ ပါဘူး။ အဲဒီ SDK 1.4 ကို စာဖတ်သူရဲ့ System's Class PATH မှာ configuration လုပ်ရေးမှာပါ။ ဒါမှသာ Java Language ကို ရေးသား ဖန်တီးပြီးတော့ အသုံးချု လို ရမှာပါ။ ဒီအချက်ကို စာဖတ်သူအနေနဲ့ အထူးတလည် သတိပြုမိန့် လိုအပ်ပါတယ်။ ဒါကြောင့် 2.6.2 Configuration the Software Development Kit ဆိုပြီးတော့ ဆက်လက် သင်ကြား ပို့ချေပေးထားပါတယ်။ စာဖတ်သူအနေနဲ့ အချိန်ယူပြီး သေသေချာချာ လေ့လာစေချင်ပါတယ်။

အခု ဆက်လက် ဖော်ပြေပေးထားတဲ့ Figure တွေကတော့ Install လုပ်ရမှာ အဆင့်ဆင့် Step By Step Wizard နဲ့ ဆောင်ရွက်ရမှာ အောင်မြင်ပြည့်စုံစွာ Install လုပ်နိုင်တယ်ဆိုတာ ဖော်ပြေပေးထားတဲ့ ဂုဏ်ပြုပါပဲ။ Figure 2.6.9 အနေနဲ့ ယုံ့တွဲပြီးတော့ ဖော်ပြေပေးထားပါတယ်။ ဒီ Step မှာတော့ အချိန်အနည်းငယ် လောက်တော့ စောင့်ဆိုင်းရပါတယ်။ ကဲ– လေ့လာလိုက်ပါပြီး။



Figure 2.6.9 The 9<sup>th</sup> Step (or) The Complete Step



Figure 2.6.9 The 9<sup>th</sup> Step (or) The Complete Step



Figure 2.6.10 To Show the README File

အခု ဖော်ပြထားတဲ့ Figure 2.6.10 ဟာဆိုရင် အရှေ့က Figure 2.6.9 မှာပါတဲ့ Install Shield Wizard Complete ဆိုတဲ့ The last Step Dialog Box ၏ Finish Button ကို မနိုင်မဲ့ Yes ,I want to view the README file ဆိုတဲ့ စာသားရှေ့မှာရှိရှိတဲ့ Check Box ကို 1-Click ခေါက်ပြီးတော့မှ Finish Button နှင့်လိုက်ရင် ဖော်ပြေပေးလာမယ့် README Text File ပါပဲ။

ဒါဆိုရင်တော့ Install လုပ်တာဟာ အောင်မြင်သွားပါပြီ။ ကျွန်တော်ရှေ့မှာ ရေးခဲ့သလိုမျိုး Install လုပ်ပြီးရှုံးနဲ့တော့ Java 2 SDK ကို တန်းပြီး အသုံးချုလို မရသေးပါဘူး။

ဒါကြောင့် Java 2 SDK ကို စာဖတ်သူရဲ့ System's CLASSPATH မှာ သွားရောက်ပြီးတော့ Configuration လုပ်ကြရအောင်လား။

ကဲ– ဒီတော့ Sub-title ဖြစ်တဲ့ 2.6.2 Configuration the Software Development Kit ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက်လေ့လာကြပါစိုး။

KYAW ZAYAR LAY [ IT ENGINEERING ]

## JAVA DEVELOPER GUIDE

### 2.6.1 CONFIGURATION THE SOFTWARE DEVELOPMENT KIT

ကွန်တော် အရှေ့မှာ ရေးသားပို့ချခဲ့သလိုပါပဲ။ Java Language ကို Install လုပ်ပြီးပြုဆိုတာနဲ့ အခြား language တွေမှာ လိုပျိုး တိုက်ရိုက် ဆောင် အသုံးချလို့ မရပါဘူး။ Configuration လုပ်ပေးဖို့ လိုအပ်ပါတယ်။

ဒီနေ့တော်မှာ Useage ဖြစ်ပြီးတော့ အသုံးများတာကတော့ Windows XP ဖြစ်ပါတယ်။ ဘာကြောင့်လဆိုတော့ ယနေ့ windows user တော်တော်များများဟာ Windows XP ကို အသုံးများလို့ အခုခင်ခန်းစာများ Windows XP ကို အခြေခံထားပြီး တော့ ရှင်းပြပေးထားပါတယ်။ Windows 95,98, or Me သမားတွေအတွက်လည်း ထည့်သွေးပြီးတော့ ရှင်းပြပေးထားပါတယ်။

Java ကို Install လုပ်ပြီးပြုဆိုတာနဲ့ command-line Interface လိုပေါ်တဲ့ DOS ထဲမှာ ဝင်ပြီးတော့ configuration လုပ်ရမှာပါ။

ဤတော့ Windows ရဲ့ System's CLASSPATH ထဲမှာ လည်း ဝင်ရောက်ပြီးတော့ configuration လုပ်ရမှာပါ။

Note : Windows 95,98,or Me ကို အသုံးပြုမယ်ဆိုရင်တော့ Start Button ကနေ Programs, then Ms-DOS Prompt ကို ရွေးရမှာပါ။ Windows XP professional သမားတွေ ဝင်ရောက်ရမယ့် path လမ်းကြောင်းကို Figure 2.6.1 မှာ ပြထားပေးပါတယ်။ လေ့လာတွေ့ပါ။

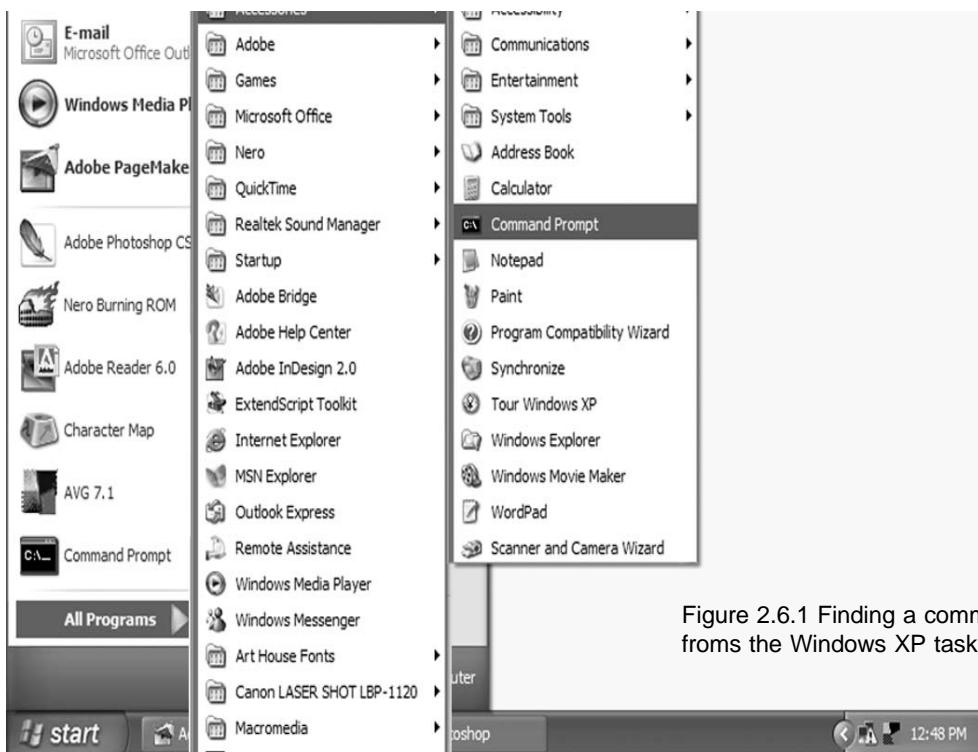


Figure 2.6.1 Finding a command - line froms the Windows XP taskbar

## ကွန်ပါ။တာသင်ခန်းစာ

တကယ်လို့ ကျွန်တော် အရှေ့မှာပြောခဲ့သလိုပဲ စာဖတ်သူဟာ ဒီနေ့ခေတ်မှာ အသုံးပြုများတဲ့ Windows XP ကို အသုံးပြုတယ် ဆိုရင်တော့

Click Start Button → All Programs → Accessories  
→ Command prompt କିମ୍ବା ଟିପ୍ପଣୀ ଫଳ୍ଗୁନୀ ରଖାଯାଇବାପାଇଁ

Windows NT, or 2000 ကို အသုံးပြုမယ်ဆိုရင်တော့  
Click the Start Button → Programs → Accessories  
→ Command Prompt ကို ရွေးချယ်ရမှာပါ။

Note : ဒီနေရာမှာ တစ်ချက်ရှင်းပြလိုတာကတော့ ကျွန်ုတ်တော်က  
တော့ Windows XP ကို အသုံးပြုပါတယ်။ ဒါပေမယ့်  
Windows XP ကို အသုံးပြုရာမှာ taskbar ပေါ်မှာ  
R.Click ခေါက်ပြီးတော့ properties ကို ရွေးချယ်ပါ  
တယ်။ အဲဒီကမှ Properties Dialog Box ပေါ်လာရင်  
Start Bar ရဲ့ style ကို Windows Classic Style  
ကို ပြောင်းထားတာပါ။ Windows 95, 98, or Me  
ကနေ ပြောင်းလဲလာတဲ့ သူတော့ အနိုင်ကတော့ အဲဒီလို  
ပြောင်းလဲထားတာ အကောင်းဆုံးပါပဲ။ Windows XP  
style < မှလ Default အတိုင်း > ထားရင်တော့ Start  
→ All Programs → Accessories → Command  
Prompt ကို ရေးချယ် ဖွင့်လှစ်ရမယ်ပါ။

ပုံမှန်မြင်ရတဲ့အတိုင်း Windows XP users ထွေဟာ command-line Windows (Ms-Dos ပဲ ဆိုကြပါစို့) ပွင့်လာဖြစ်ဆိုရင် အောက်ပါ Command Line ကို ရိုက်သွင်းပြီးတော့ Enter key ကို နိုင်ပါ။

CD<sub>1,1</sub> ↴

ဒီအခါမှာ Command - Line Windows တဲမှုရှိတဲ DIR  
ဟာ တစ်ဆင်ပြေတိယက်သားပြီးတောာ

ဆိတ် Prompt ကိုပဲ ထွေရမှာပါ။ အဲဒီအခါမှာ အောက်မှာရေးထားတဲ့ command ကိုပဲ ဖောက်တစ်ကြိမ် ရိုက်သင်းပါဘီ။

CD

C: ଯାକ୍ଣି ଠିକ୍ ରେଖାଗଳିତ କରିବାକୁ ପାଇଁ Step କି ଫିଗ୍ଯୁରୁ 2.6.2 ଏବଂ ଫିଲ୍ ପାଇଁ ପରିଚୟ ଦିଆଯାଇଛି।

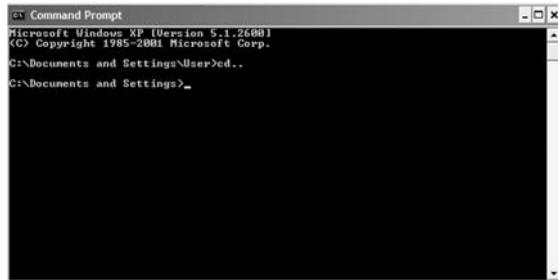


Figure 2.6.2 Using a newly opened command -line Windows

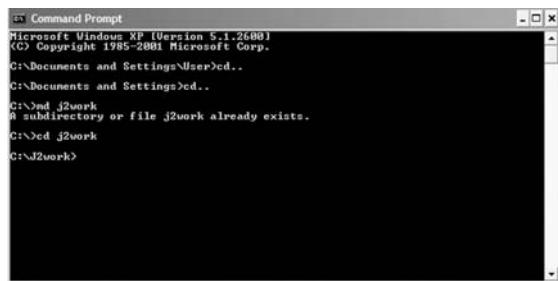
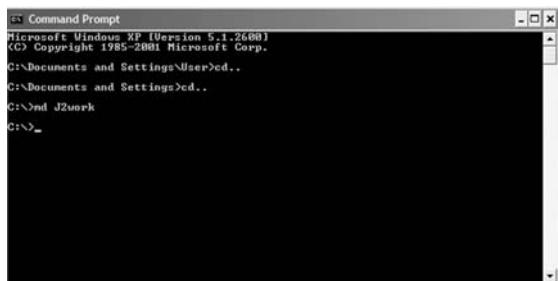


Figure 2.6.3 Creating a new folder in a command-line Windows

ကဲ- အခါအခါမှာ Figure 2.6.3 မှာ တွေရတဲ့အတိုင်း C:> MD J2Work လို့ ရိုက်သွင်းပြီးတော့ J2Work လို့ အမည်ရှိရတဲ့ Directory တစ်ခုကို တည်ဆောက်ပါ။ MD ဆိုတာက တော့ MS- DOS ကို လေ့လာဖူးသူတွေအတွက်ကတော့ အရေး မကြိုးပေမယ့် မလေ့လာဖူးသေးတဲ့ user တွေအတွက်ကတော့ Directory တစ်ခုကို ဖော်ထိုး တည်ဆောက်တယ်လို့ သာတော့ ပေါက် နားလည်ထားပါ။ M Directory လို့ခေါ်တဲ့ Command ပါပဲ။ ဒါကိစ်ရင်တော့ ပင်မ Hard disk C: ထဲမှာ J2Work ဆိုတဲ့ Folder တစ်ခု ပေါ်လာပြီပေါ့။ အခါအတိုင်း Folder တည် ဆောက်လို့ရသလို အောက်မှာ ရေးထားတဲ့ command နဲ့လည်း အတွေ့ပါပဲ။

## ကွန်ပျူတာသင်ခန်းစာ

MD C:\ J2Work ↵

ဒီ command ဟာ အပေါ်မှာရေးခဲ့တဲ့ command နဲ့ အတူ တူပါပဲ။ ဒါမှမဟုတ် စာဖတ်သူဟာ အခုလို command line Interface ကို အသုံးပြုပြီးတော့ မတည်ဆောက်ချင်ရင် အောက်မှာ ဖော်ပြထားတဲ့ path အတိုင်း သွားပြီးတော့ အဂ္ဂလိုတကူ တည် ဆောက်နိုင်ပါတယ်။

My Computer → Drive C: ကို Double-Click ခေါက်ပါ။ ဒီအခါမှာ Drive C: ဟာ ပွင့်လာမှုပါ။ ပွင့်လာတဲ့ Drive C: ထဲမှာ Right Click ခေါက်ပြီးတော့ ကျလာတဲ့ Pop-Menu ထဲက New ကိုရွေးပါ။ ဒါကို Figure 2.6.4 အနေနဲ့ ပြပေးထားပါတယ်။

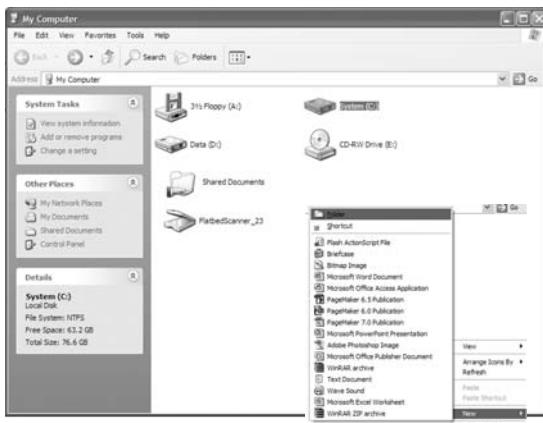


Figure 2.6.4 Inside Drive C: and Create New Folder

ဒီအခါမှာ New Folder တစ်ခုကျလာရင် အဲဒီ Folder ရဲ့ Name ကိုတော့ J2Work လို့ အမည်ပေးလိုက်ပါ။ အောက်မှာရှိတဲ့ Figure 2.6.5 ကို ကြည့်ပါ။ ဒီနည်းဟာလည်း လိုချင်တဲ့ Directory တစ်ခုကို ဖုန်းတိုးနည်းပါပဲ။



Figure 2.6.5 Create New\_Folder <Make Directory>  
Naming J2Work

ကဲ ဆိုကြပါစို့။ အရှေ့မှာ ဖော်ပြခဲ့တဲ့ Directory တစ်ခုကို ဖုန်းတိုးပြီးဆိုတာနဲ့ အောက်မှာဖော်ပြထားတဲ့အတိုင်း command -line Interface ထဲကို ဝင်ပြီးတော့ Drive C: ရှိရာကို ရောက်

အောင်သွားပါ။ ကျွန်တော့အနေနဲ့ အဆင့်ဆင့် ဝင်ရောက်ပုံကို Figure 2.6.6 အနေနဲ့ ဖော်ပြပေးထားပါတယ်။

```
cmd Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\user>cd..
C:\Documents and Settings\user>cd..
C:\>cd J2work
A subdirectory or file J2work already exists.
C:\>cd J2work
C:\J2work>cd..
C:\>java -version
```

Figure 2.6.6 Run the command java -version in that folder

အရှေ့မှာ ရေးသားခဲ့တဲ့ J2work ဆိုတဲ့ Directory ကို မတည်ဆောက်ရင်လည်း ရပါတယ်။ အဲဒီ Directory ကို တည် ဆောက်ရခြင်းကတော့ သီးသန့် Class & Object တွေကို ဖုန်းတိုး ဖို့ပါပဲ။ မတည်ဆောက်ခဲ့ရင်လည်း ရပါတယ်။ က အဲဒီတော့ အပေါ်မှာ ဖော်ပြခဲ့တဲ့အတိုင်း Drive C: ထဲကို ရောက်သွားပြီ ဆိုရင်တော့ အောက်ပါ command - line ကို ရှိပြီးတော့ Java Interface Run-Out လုပ် မလုပ် စမ်းကြည့်ပါ။

Java - Version ↵ underscore မဟုတ်ပါ။ သချိုအကွဲရာ  
(-) ဖြစ်ပါတယ်။

အဲဒီအခါမှာတော့

Java version "1.4.0-rc"

Java <TM> 2 Runtime Environment, Standard Edition <Build 1.4.0-rc-bg1>

Java HotSpot <TM> Client VM <build 1.4.0-rc-bg1, mixed mode> ဆိုတဲ့ prompt text ဟာ ဖော်ပြပေးမှုပါ။ ဒါအဲရင်တော့ သင့်ရဲ့ Computer ထဲမှာ Java Interpreter ဟာ Run နေပြီဆိုတာ သေချာသွားပါပြီ။ Figure 2.6.7 မှာ ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```
cmd Command Prompt
C:\Documents and Settings\user>cd..
C:\Documents and Settings\user>cd..
C:\>cd J2work
A subdirectory or file J2work already exists.
C:\>cd J2work
C:\J2work>cd..
C:\>java -version
java version "1.4.1_01"
Java(TM) 2 Runtime Environment, Standard Edition <build 1.4.1_01-b01>
Java HotSpot(TM) Client VM <build 1.4.1_01-b01, mixed mode>
C:\>
```

Figure 2.6.7 Running the Java Interpreter in a command line window

တကယ်လို့များ စာဖတ်သူဟာ Installing လုပ်ရာမှာ အနည်းငယ် မှားယွင်းခဲ့မယ်။ ဒါမှမဟုတ် System's Class PATH ထဲက java.exe ကိုရှာတဲ့ မတွေ့ဘူးဆိုရင်တော့ သင်ဟာ C:\> Prompt ပေါ်နေခိုန်မှာ java -version ကို ရှိက်သွင်းလိုက်ရင် အောက်ပါ error messages ဟာ ဖော်ပြလာမှာပါ။

Bad command or file name.

'java' is not recognized as an internal or external command, operable program, or batch file

ကဲ ဒီ error message တက်လာပဲကို Figure 2.6.8 အနေနဲ့ ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 2.6.8 Con't find the Java .exe in your system, to show the error message

ကဲ ဒီ error message ပြသည်ဖြစ်စေ၊ Java interpreter Run လုပ်သည်ဖြစ်စေ ကျွန်ုတ်တော်တို့ဟာ Java platform ကို အသုံးပြန်ဖို့တွက် System's PATH variable ထဲကို ဝင်ပြီး တော့ configure လုပ်ရမှာပါ။

#### ◆ Setting the path on Windows NT, 2000, or XP

အခုတင်ပြမှာကတော့ ယနေ့တော်မှာ အသုံးများနေတဲ့ Windows NT, 2000, or XP တွေမှာ Configuration လုပ်နည်းကိုပဲ အရင်တင်ပြပေးပါယ်။

Windows NT, 2000, or XP တွေမှာ System's PATH variable ကို ပြုပြင်မယ်ဆိုရင် Environment Variables Dialog Box ကို ခေါ်ယူ အသုံးပြုရမှာပါ။

Environment Variables Dialog Box ကို ခေါ်ယူအသုံးပြုပဲကတော့

- (1) Desktop Screen ပေါ်မှာရှိပဲ My Computer icon ကို ဖြစ်စေ၊ Start Menu ကို ဖြစ်စေ Right Click ခေါက်ပါ။
- (2) ပေါ်လာတဲ့ Pop-up Menu ထဲက Properties ကိုရွေးရင် System Properties Dialog Box ကို ဖွင့်လာမှာပါ။

Figure 2.6.9 ကိုကြည့်ပါ။



Figure 2.6.9 (a) R-Click in star bar

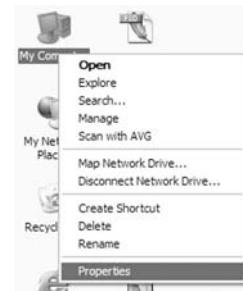


Figure 2.6.9 (b) R-Click In My Computer Icon

- (3) အဲဒီ Dialog Box ထဲမှာပါဝင်တဲ့ Advanced Tab ကို နှိပ်လိုက်ပါ။ Figure 2.6.10 အနေနဲ့ ဖော်ပြထားပါတယ်။



Figure 2.6.10 The System Properties dialog Box Open

- (4) အဲဒီတဲ့ ကျွန်ုတ်တော် မှာ နှိပ်လိုက်ရင် Environment Variables Dialog Box ပွဲလာတာကို တွေ့ရမှာပါ။ Figure 2.6.11 မှာ ပြထားပေးပါတယ်။

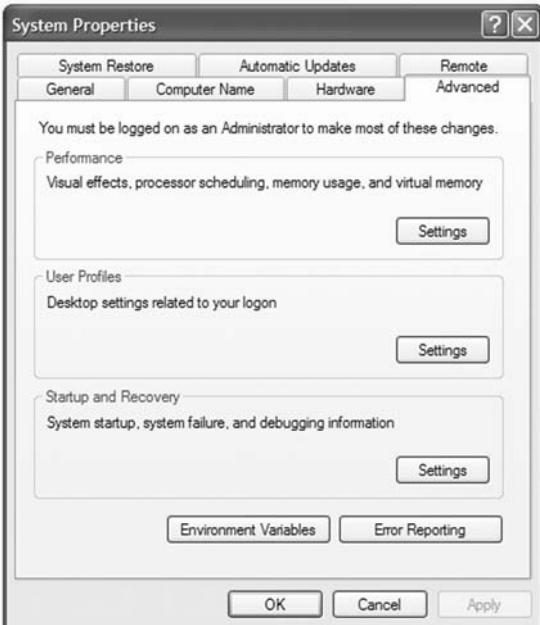


Figure 2.6.11 (a) To Click the Advanced tab

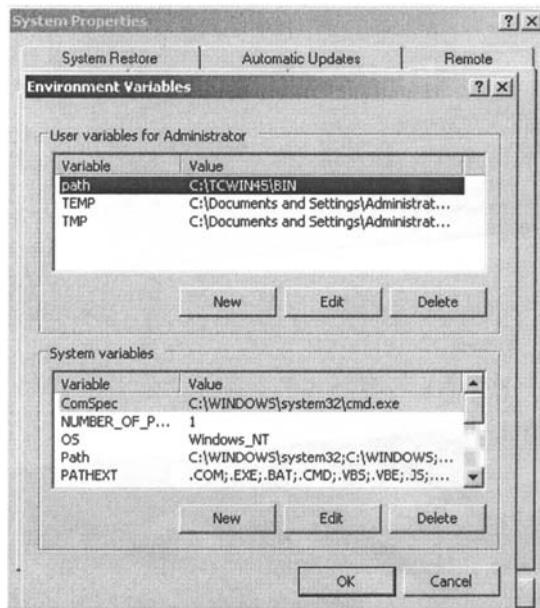


Figure 2.6.11 (b) To Click the EnvironmentVariables

Installလုပ်စဉ်က Java SDK 1.4 ဟာ C:\j2SDK 1.4.0 folder ထဲမှာ java.exe အနေနဲ့ တည်ရှိနေတာပါ။ ဒီတော့ Environment Variables Dialog Box ထဲမှုရှိနေတဲ့ System Variables list ရဲ့ အောက်မှာရှိတဲ့ pathကို 1 click ခေါက်ပြီး တော့ Edit Button ကိုနှိပ်ပါ။ ဒီအခါမှာ Edit System Variables Dialog Box ဖွင့်လာပါပြီ။ အဲဒီ System Variables Dialog Box ဖွင့်လာပဲကို Figure 2.6.12မှာ ဖော်ပြထားပါတယ်။ Step By Step Guid အနေနဲ့ ဖော်ပြထားတဲ့အတွက် မှားယွင်းစရာ မရှိပါဘူး။ သေသေချာချာ လေ့လာကြည့်ပါ။



Figure 2.6.12 To Click the Edit Buttomm In the Environment Variables

## ကွန်ပူးတာသင်ခန်းစာ

အဲဒီ Edit System Variables Dialog Box မှာရှိတဲ့ Variables Name နောက်မှာရှိတဲ့ Text Box ထဲမှာရှိတဲ့ path လမ်းကြောင်းကို ဖြင့်စရာ မလိုပါဘူး။ Variable value နောက်မှာရှိတဲ့ Text Box ဖြစ်တဲ့ Field ထဲမှာ mouse ကို 1-Click ခေါက်ပြီးတော့ cursor ကို စာကြောင်းအဆုံးကို ရောက်သွားအောင် keyboard က end key ကို နိုင်ရင် စာကြောင်းအဆုံးကို ရောက်သွားမှာပါ။

Note : အဲဒီ System Variable Value Field ထဲက စာသားတွေ ကို လုံးဝမပြပြင်စိန့်က အထူးအရေးကြုံးပါတယ်။ မျက်စိ ပြမိုခြင်း မရှိအောင် သတိထားပါ။ မတော်တဆ မျက်စိ သွားရင် OK Button ကို နိုင်ပဲဘူး၊ cancel Button ကို နိုင်ပြီးတော့ ပြန်ထွက်ပါ။ ပြီးမှ Edit Button ကို ပြန်နိုင်ပြီးတော့ ပြန်ဝင်ရမှာပါ။ Cancel လုပ်ခြင်းဟာ မှားယွင်းလုပ်မိသမျှကို အတည်ပြုပေးခြင်း မရှိပါဘူး။

ကဲ cursor ဟာ Variable value နောက်မှာရှိတဲ့ Text Box ရဲ့အတွင်းထဲမှာရှိတဲ့ text တွေရဲ့ နောက်ခုံးတို့ ရောက်သွားရင် ; C: \ j2SDK 1.4.0 \ bin ကို ရှိခိုက်သွားပါ။ ; (semicolon) ပါရဂါမယ်။ စာကြောင်းအဆုံးမှာတော့ ; (semicolon) မပါပါ ဘူးနော်။ ကဲ အဲဒီလို ရှိခိုက်သွားတာကို Figure 2.6.13 မှာ ဖော်ပြထားပါတယ်။ လေးလာကြည့်ပါ။

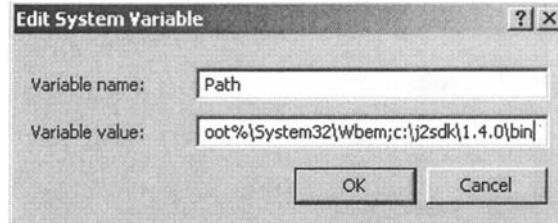


Figure 2.6.13 Type the ; C: \ j2sdk \ 1.4.0\bin path in the variable value Text Box

ကဲ variable value ထဲမှာ path လမ်းကြောင်းလိုအပ်တဲ့ စာသားတွေ ထင်ဖြည့်ပြီးသွားရင်တော့ အဲဒီ Edit System variable Dialog Box ကို OK လုပ်ပြီးတော့ ပိတ်ပါ။ အဲဒီနောက် Environment Variables Dialog Box ကိုလည်း OK Button ကို နိုင်ပြီးတော့ ပိတ်လိုက်ပါ။

Note : ကျွန်ုတ်အခုံတင်ပြေားတဲ့အတိုင်း System's PATH ကို ပြပြင်ပြီးရင်တော့ အကောင်းဆုံးနည်းလမ်းကတော့ Windows ကို Restart လုပ်တာ အကောင်းဆုံးပါပဲ။

ကဲ ဒါဆိုရင်တော့ သင့်ရဲ့ computer ထဲမှာ Java Software Development kit ကို Install လုပ်ပြီးသွားပါပြီ။ ကဲ GUI windows တွေနဲ့ ပတ်သက်တဲ့ Chapter [3] ကို လုပ်ငန်းခွင်က Program Type တွေနဲ့ သင့်ကြေားပို့ချေပေးထားပါတယ်။ လေးလာကြည့်ပါ။

KYAW ZAYAR LAY [ IT ENGINEERING ]

## စမတ်ကျကျမြင်ကွင်းကျယ်ကျယ်

၃၇ လက်မအရွယ် LCD ဖြင်ကွင်းက Wi-Fi သို့မဟုတ် Ethernet ကနေ အင်တာနှင်းသို့ တိုက်ရှိက် ချိတ်ဆက်ပေးနိုင်ပါတယ်။ ဗိုဒ်ပို့ယို့ တေးဂိုဏ် အစီအစဉ်တွေသာမက DivX ဖိုင်များနဲ့ still ပုံတွေကိုပါ ဖော်ယူနိုင်ပါတယ်။ 6-millisecond pixel အထိ အကွက်နောက် ဖြင်ကွင်းက ပြတ်သားတဲ့ ပုံရိပ်များကို ဖော်ဆောင်ပေးမှာပါ။ တို့ပို့ယို့ပုံးသုံး၊ ပို့ပို့နဲ့ ဆက်လိုက်ဆက်၊ တက်ယူစမတ်ကျကျတဲ့ ဖြင်ကွင်းစက်တစ်လုံးဖြစ်ပါတယ်။ SLC3760N လို့ ဖော်ပြထားတဲ့ အဆိုပါ ဖြင်ကွင်းစက်အတွက် ဈေးနှုန်းဖော်ပြုမထားပါ။ လိုပ်စာက www.hp.com ဖြစ်ပါတယ်။ 'PC Magazine' မှ အချက်အလက်များ ဖြစ်ပါသည်။





### 3.1 How To Run JAVA PROGRAM IN WINDOWS PLATFORM

ကဲ- ကျွန်တော်တို့ Java Programming Tools ကို စာဖတ်သူရဲ့ computer ထဲမှာ Install လုပ်ပြီး။ System's CLASSPATH မှာလည်း ပြင် ဖန်တီးပြီးချိတ်တော့ အရှေ့မှာ သင်ကြားပို့ချဲခဲ့တဲ့ listing တွေ အခါ ဆက်လက်ပို့ချုပ်ယော် သင်ခန်း စာတွေကို ဆက်လက်ပြီး Run လုပ် လေ့လာလို့ရပြီးပေါ့နော်။

ဒီနေရာမှာ ကျွန်တော် မူးလျော့ခဲ့တဲ့အချက် ရှိနေတာကို အမှန် အတိုင်း ဝန်ပံ့ရရစေ။ ဘာကြောင့်လဲဆိုတော့ ဒီ Java 2 Software Development Tools ကို Install လုပ်တဲ့ Install Guide ကို သင်ခန်းစာ အစကာတည်းက ပို့ချုပ်မှာကို မပို့ချေပေးမိတဲ့ အချက်ပါပဲ။ တကယ်လို့သာ J2SDK Install Guide ကို အရင် ပို့ချဲခဲ့ရင် စာဖတ်သူ လေ့လာရတာဟာ ပို့ပြီးအဆင်ပြုမှာပါ။ အချို့ programming အတွေအကြံရှိပြီးသား ပါရင် programmer တွေ ကတော့ ဒီအချက်ကို သတိမပြမ်မပေးမယ့် programming အတွေ အကြံမရှိတဲ့သူတွေအဖို့ Java Language နဲ့ Program တွေ ဖန်တီး ရေးသားဖို့အတွက် အတော်လေးကို အခက်အခဲရှိလာကြ လိုပါပဲ။

အဲဒါကြောင့် ကျွန်တော်အနေနဲ့ ကြားဖြတ်သင်ခန်းစာအနေနဲ့ J2SDK Install Guide ကို ပို့ချေပေးလိုက်ရတာပါပဲ။ ကဲ - အခါ Chapter [3] အနေနဲ့ ကျွန်တော်တို့ 3.1 How To Run JAVA Program In Windows Platform ဆိုပြီးတော့ Windows Base GUI programming တွေကို စတင် လေ့လာ သင်ယူကြ တာပေါ့။

#### ◆ Writing Program Code-Line

Java Language ရဲ့ အားသာချက်ကတော့ အခြား language တွေမှာလိုမျိုး Text Area (or) Text Pad မပါဝင်ပါဘူး။

ဒီတော့ အရှေ့မှာ သင်ကြားပို့ချဲခဲ့တဲ့ သင်ခန်းစာမှာ ပြောခဲ့လိုပါပဲ။ Windows နဲ့အတူ Built-In ပါလာတဲ့ Note Pad ဒါမှာမဟုတ် Word Pad အစရှိတဲ့ Program တွေကိုပဲ အသုံးပြု မလေး။

ဒါမှမဟုတ် Microsoft Office အပ်စာတဲ့ MS-Word ဒါမှမဟုတ် Word Perfect အစရှိတဲ့ Word processing software တွေကိုပဲ အသုံးပြုမလေး။ စာဖတ်သူနှစ်သက်ရာတိ ရွေးချယ် နိုင်ပါတယ်။

ကဲ- ကျွန်တော်ကတော့ Note Pad ကိုပဲ အသုံးပြုထားပါ တယ်။ Start → Program → Accessories → Note Pad ကို သွားပါ။ Note Pad Program မွန်လာမှာပါ။ Note Pad Programs ထဲ ဝင်ရောက်ပဲကို Figure 3.1 မှာ ပြထားပါတယ်။



Figure 3.1 To enter the Note Pad <Text Editer> In windows XP

က Note Pad Program ထဲ ရောက်သွားပြီ ဆိုကြတဲ့ပါ။ Listing 3.1 အနေနဲ့ ရေးသားပြပေးထားပါတယ်။ လေ့လာ ကြည့်ပါ။

## Listing 3.1 Using the JFrame Class in JAVA Swing Package

```
1. import javax.swing.*; // to call the swing Java Components
2. import java.out.*; // out components
3. public class test // this Class_Name is equal in this File_Name
4. {
5.     public static void main (string args [ ] ) {
6.         JFrame f = new JFrame ( ) ; // create object in JFrame
7.         f.setTitle ("JAVA Programming in Windows Platform") ; // show text in Title Bar
8.         f.setSize (300,200); // windows size
9.         f.setDefault Close Operation (JFrame.EXIT _ON_CLOSE) ;
10.        f.show ( ) ; // to call show method ( ) ;
11.    }
12. }
```

က အခါ Listing 3.1 မှာ ရေးသားခဲ့တဲ့ code-line တွေကို Note Pad မှာ ရိုက်သွင်းရမှာနေ၍။ အဲဒီလို Note Pad မှာ ရိုက်သွင်းထားပဲကို Figure 3.2 အနေနဲ့ဖော်ပြုပေးထားပါတယ်။ လျော့လှကြည့်ပါ။

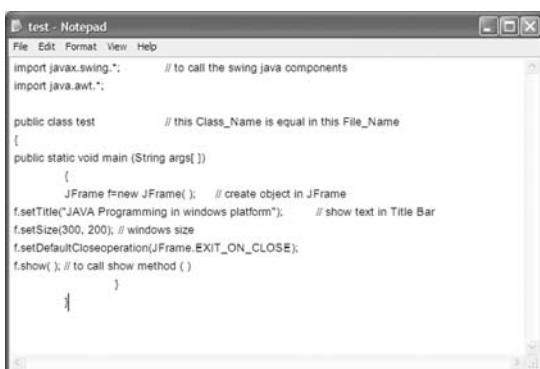


Figure 3.2 Create the JAVA Program code-line In Text Editor (Note Pad)

କି ଆଉ code-line ଦେଇ ରେ:ବା:ପ୍ରି:ଟୁ Note Pad କିମ୍ବା save ଲ୍ୟାପଟ୍ରି:ଟୋକୁ ଯିତିଃଶବ୍ଦିଃପ୍ରିମାଯି। ଅତିକିଳି ଯିତିଃଶବ୍ଦିଃଟୁ ଅପିଥିବା ବାମକ୍ଷ Program Application ଦେଇଗି ଯିତିଃଶବ୍ଦିଃ ବାଲିଖିମି: My Documents ହିଁ କେବଳ ଡ୍ରାଇଵ C: ଅତିଥିବା ଡ୍ରାଇଵ

D: အစရိတ္ထဲအထဲသူးပြီး save လုပ်လို မရနိုင်ပါဘူး။ C: \j2sdk  
1.4.1\_01\ Bin ဆိတဲ့ Path လမ်းကြောင်းအတိုင်း သွားရောက်ပြီး  
တော့ save လုပ် သိမ်းဆည်းရဖူးပါ။ ကဲ ဒီတော့ File ကနေ  
Save ကို တောင်းပါ။ အခု လက်ရှိ code-line တွေ ရှိက်သွင်း  
ထားတဲ့ Note Pad program ရဲ့ File menu က Save ကို  
တောင်းခိုင်းတာနေ့။ ဒီအခါမှာ Save Dialog Box ကျလာမှုပါ။  
အဲဒီ Save Dialog Box ထဲမှာပါဝင်တဲ့ Save In နေရာမှာရှိတဲ့  
Drop List Buttom ကို ရွေးပြီးတော့ Drive C: ကို ရွေးပါ။  
Drive C:\ ပွင့်လာရင် j2sdk 1.4 .1\_01 ကို ပြန်ရွေးပြီးတော့  
Double-Click ခေါက်ပါ။ ကဲ j2sdk 1.4.1\_01 ပွင့်လာရင်  
အဲဒီထဲကမဲ Bin ဆိတဲ့ folder ကို ရှာပြီးတော့ Double -Click  
ထပ်ခေါက်လိုက်ပါလိုး။ ဒီအခါမှာ Bin folder ထဲကိုရောက်ပြီ  
ဆိတဲ့မူ ကျွန်ုတ်တို့ code-line တွေ ရေးသားခဲ့တဲ့ file ကို  
test.java ထိ File Name ပေးလိုက်ပါ။

အေဒီလိုမီး ပင်မ Drive C: ထဲသွားပြီး C:\j2sdk 1.4.1\_01\ Bin နိုတဲ့ path ထဲမှာ သွားပြီးတော့ Save လုပ် သိမ်းဆည်းထား ပဲကို Figure 3.3 မှာ ပြပေးထားပါတယ်။

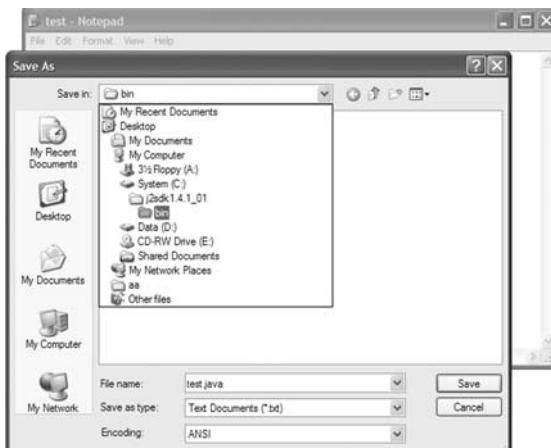


Figure 3.3 To Save In the C:\j2sdk 1.4.1\_01\bin,  
CLASSPATH Folder

အကျယ်ရင်းပြရရင်တော့ save လုပ်ခါနီးမှာ Drive C: ထဲကို  
အရင်ဝင်ပါ။ အဲဒီကု j2sdk1.4.1\_01 ကိုထွေရင် D-Click  
(Doucle Click) ပေါက်ပါ။ အဲဒီအခါမှာ ပေါ်လာသူ Directory  
Folder ထွေထဲကမှ Bin ဆိုတဲ့ Folder ကို (2) ချက် D-Click  
ပေါက်ရမှာပါ။ အဲဒီ path လမ်းကြောင်းထဲကို ရောက်တော့မှ test  
.java လို့ အမဟုပ်ပေးပြီးတော့ save လုပ်ရမှာပါ။

## ကွန်ပူးတာသင်ခန်းစာ

**Note :** အဲဒီလို save လုပ်ရမှာ ဝင်ရောက်ရမယ့် Drive path ပုံကို Figure 3.3 မှာ ဖော်ပြုပေးခဲ့ပါတယ်။ အဲဒီလို ဝင်ရောက်ပြီးတော့ save လုပ်ရမှာ save လုပ်တဲ့ File\_Name ဟာ Program code-line ရေးစဉ်က ဖန်တီးခဲ့တဲ့ Class\_Name နဲ့ တူရပါမယ်။ အဲဒီလို တူညီမှသာယျာင် Run လုပ်တဲ့အခါမှာ error လာမဖြားဖြစ်ပါတယ်။ ဒါကြောင့် save လုပ်တဲ့အခါမှာ ကိုယ် program ရေးစဉ်က တည်ဆောက်ခဲ့တဲ့ Class\_Name နဲ့တူအောင် သတိပြုပြီးတော့ save လုပ်ရမှာပါ။ နောက်ထပ် အရေးကြီးတဲ့ အချက်ကတော့ save လုပ်ရာ ကျွန်တော်တို့ Install လုပ်ခဲ့ j2sdk1.4.1\_01 program folder ထဲက Bin ဆိုတဲ့ folder ထဲမှာ သွားပြီး save လုပ်ရမှာနော်။ အဲဒီလို save လုပ်မှသာယျာင် Run & Test လုပ်လို့ရနိုင်မှာ ဖြစ်ပါတယ်။ အခြားသော Text Editor တွေကို အသုံးပြု ရေးသားပြီးတော့ save လုပ်ပြီး Run မယ်ဆိုရင်လည်း ရနိုင်ပေမယ့် လောလောဆယ် ထယ် စောင် လေးလာခါစာ programmer တစ်ယောက် အဖို့ အခုန်ညွှေးလမ်းဟာ အဆင်အပြဆုံးလို့ ယူဆပါတယ်။ ဒါကြောင့် ကျွန်တော် ပြောသလိုပါပဲ Note Pad မှာ စာသား အနေနဲ့ code-line တွေကို ရိုက်သွင်းပြီး Figure 3.3 မှာ ပြုခဲ့သလို စနစ်တကျနဲ့ သေသေချာချာ save လုပ်ပါ။

က ပေါ်ပြီးပြုလို ဆိုကြားပါ။ Code-Line တွေ ရေးသားထားတဲ့ Note-Pad File ကို မဖွင့်သေးဘူး၊ Command prompt <DOS> ကို Start → Program → Accessories → Command Prompt ကို 1-Click ခေါက်ပြီးတော့ ဖွင့်လိုက်ပါ။ အဲဒီအခါ Command Prompt ပွင့်လာရင် အောက်ပါအတိုင်း Command -line ကို ရိုက်သွင်းပါ။ ပေါ်နေတဲ့ Prompt က C:\ Document and settings\Administrator>လို့ ပေါ်နေမှာပါ။

CD . . . ↵

ဒါဆိုရင် Drive letter path ဟာ C:\ Documents and Settings > လို့ ပြောင်းလဲသွားပါပြီ။ ဒီအခါမှာ

CD . . . ↵

ကို ထပ်ရိုက်သွင်းပြီးတော့ ထပ်ပြီး Folder Path ကို ဖြုတ်ချလိုက်ပါ။ အဲဒီဆိုရင်တော့ C: \> ဆိုတဲ့ prompt ပေါ်လာမှာ။ အဲဒီတော့မှ

CD j2sdk1.4.1\_01 ↵

လို့ ရိုက်သွင်းပြီးတော့ Enter key ↵ ကို နိုင်ပါ။ အဲဒီအခါ C:

\j2sdk 1.4.1\_01 >လို့ ပေါ်လာပါပြီ။ ဒါဟာ Java 2 Software Development Kit တဲ့ ရောက်လာပါပြီ။ အဲဒီတော့ ကျွန်တော် တို့ code-line တွေ ရေးပြီးတော့ save လုပ်ရတာဟာ အဲဒီ j2sdk 1.4.1\_01 ထဲက Bin ဆိုတဲ့ Folder ထဲမှာနော်။ က အဲဒီ Bin folder တဲ့ ဆက်ဝင်ရအောင်နော်။ C: \ j2sdk 1.4.1\_01 > ဆိုတဲ့ prompt ပေါ်နေချိန်မှာ

CD Bin လို့ ရိုက်သွင်းပြီး Enter key ↵ ကို နိုင်ပါ။

ဒါဆိုရင် command-Line windows ထဲက Prompt သက်တာဟာ C: \j2sdk 1.4.1\_01\Bin ဆိုပြီးတော့ Prompt ပေါ်လာပါပြီ။ က ကျွန်တော်တို့ရေးခဲ့တဲ့ program ကို အရင် Compile လုပ်ကြည့်ကြရအောင်။ Compile လုပ်နိုင်ပို့အတွက် အောက်ပါ Command -Line ကို ရိုက်သွင်းပါ။

javac  
\_\_\_\_\_ |  
to call java compiler      test.java    ↵  
\_\_\_\_\_ |  
to call the File\_Name include  
code -time

က ဒီအခါ စာဖတ်သူဟာ ကျွန်တော် ရေးသားခဲ့တဲ့ code-line တွေကို အမှားအယွင်းမရှိ ရိုက်သွင်းပို့ခဲ့ရင် error လာမပြပါဘူး။ မူလ prompt ဖြစ်တဲ့ C:\j2sdk 1.4.1\_01 \ Bin > ဆိုတဲ့ prompt သာယျာင် ပြန်ပေါ်လာမှာပါ။ ဒါဆိုရင် စာဖတ်သူ ရေးသားခဲ့တဲ့ text.java ဆိုတဲ့ program ကို compile လုပ်ခဲ့ပါပြီ။

အကယ်၍သာ စာဖတ်သူဟာ Code-Line ရေးစဉ်က ;(semi-colon) ကျွန်တာ { brace အဖွင့်/အပိတ်တွေ ကျွန်တာ အစရိုတဲ့ အမှားအယွင်းတွေ ရှိခဲ့မယ်ဆိုရင်တော့-

အမှားအယွင်းရှိတဲ့ code-line ကို ʌ သက်တာနဲ့ ဗျားပြီးတော့ 1-error(သို့မဟုတ်) 2-error အစရိုတဲ့ compiler error တွေ တက်လာမှာပါ။ ဒါဆိုရင်တော့ စာဖတ်သူဟာ compiler ပြုပေးတဲ့ error တက်တဲ့နေရာကို မှတ်ထားပြီးတော့ code-line တွေ ရေးထားတဲ့ test.java ဆိုတဲ့ Note Pad ထဲကို သွားပြီးတော့ ပြန်လည် စစ်ဆေး ပြုပြင်ရမှာပါ။ ပြီးရင် error တက်တာကို သေသေချာချာစာစာစြုံးရင် File → Save As လုပ်ပြီးတော့ test.java ဆိုတဲ့ File ကို Save As လုပ်ရပြန်မှာပါ။

က compiler က ပြောတဲ့ အမှားကို ပြင်ပြီးပြီ။ Save As လည်း ထပ်လုပ်ပြီးပြုဆိုရင်တော့ Command-Line Windows ထဲက C:\j2sdk 1.4.1\_01 \ Bin > ဆိုတဲ့ prompt ပေါ်နေချိန်မှာ

## ကွန်ပျူးတာသင်ခန်းစာ

javac test.java ←  
ဆိုပြီးတော့ ပြန်ပြီး compileလုပ်ပါ။

ကဲ Compile လုပ်တာ အမှားအယုင်းမရှိဘဲ Compile လုပ်နိုင်ပြီ ဆိုပါစို့။ ဘယ်လို Compile error မှ တက်မလာဘူး၏ C:\j2sdk 1.4.1\_01\Bin > ဆိုတဲ့ prompt ကိုပဲ ပြန်ဖြေပေးယူပါ။ ဒါဆိုရင်တော့ ကျွေးနောတဲ့ ရေးခဲ့တဲ့ test.java ဆိုတဲ့ program ကို compile လုပ်ပြီးသွားပါပြီ။ အဲဒီလို compile လုပ်ပြီးသွားတာ ကို လွှေလာနိုင်ပြီ။ Figure 3.4 မှာ ဖော်ပြုပေးထားပါတယ်။ အပေါ် မှာ ရေးသားခဲ့တဲ့ အဆင့်ဆင့် C:\j2sdk 1.4.1\_01\Bin ထဲကို ဝင်ရောက်ပုံ အဆင့်ဆင့်လဲ အပါအဝင် ဖြစ်ပါတယ်။ လေ့လာကြည့်ပါ။

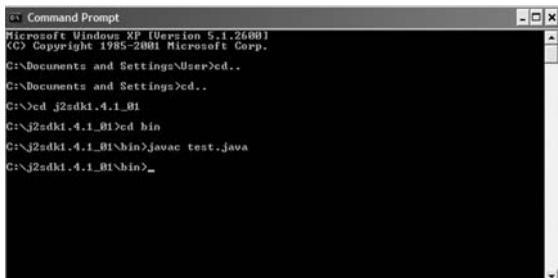


Figure 3.4 Compile the program

Note : ဒီနေရာမှာ C & C + + programmer သမားတွေအနိုင်၊  
မျက်စီရွဲပါသွားနိုင်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ C &  
C++ programming language မှာကတော့ code-line ရေးပြီး Run လုပ်တာနဲ့ error မရှိရင် output  
ဟာ တန်းပြီး ထွက်လာတာပါ။ ဒီနေရာမှာ Java  
Language မှာကတော့ complier လုပ်ရမှာ javac  
ဆိုတဲ့ keyword ကို ရှုံးကံပြီးတော့ compile လုပ်ရပါ  
တယ်။ compile error ကင်းစင်သွားပြီးတော့ မှန်သွားပြီ  
ဆိုရင်တောင်မှ Output ထွက်မလာသေးပါဘူး။ Run  
လုပ်ပြီးမှ Output ကို ခေါ်ယူကြည့်ရမှာပါ။ ဒါကို C  
& C + + programmer သမားတွေအနိုင် အထူးတလည်

ଗ୍ରାମ୍ ଲେଲାଦେଖିବା ତଥା ||  
କି ତିଟା ? compile ଲାଗିପାଇବା : test.java ଛି ତା  
program କି Run ଲାଗିବା ହେବାକିମ୍ବାନାହାନି ॥

C:\j2sdk 1.4.1\_01\Bin>ဆိတဲ့ Promptပါနေချိန်မှာပဲ  
java test ↵

လို့ ရိုက်သွင်းပြီးတော့ Run လုပ်လိုက်ပါ။ အဲဒီလို ရိုက်သွင်းပြီး  
တော့ Run လုပ်ပုံကို Figure 3.5 မှာ ရေးပြေးထားပါတယ်။  
သေသေချာချာ လေ့လာကြည့်ပါ။

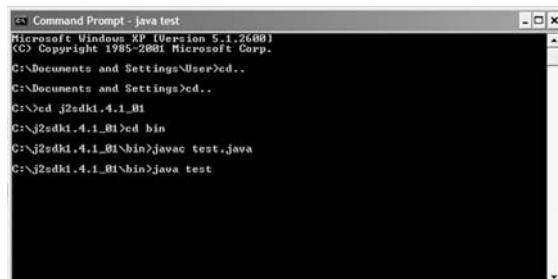


Figure 3.5 Run the JAVA Program in the command-line windows

က အခေါ်ပြခဲ့တဲ့ Figure 3.5 ဟာ Run လုပ်စီ command-line ကို ရိုက်သွင်းပေးတဲ့ ပုံပါပဲ။ က အခု command-line ကို ရိုက်သွင်းပြီးတော့ Enter key ↵ ကို နိုပ်လိုက်ပြီ ဆိုကြပါစိုး။ အဒီ program ရဲ့ output windows ဖြစ်တဲ့ JFrame Class ကို အခြေပြုထားတဲ့ Output ဟာ Computer Screen ရဲ့ ဘယ်ဘက် အပေါ်ထောင့်မှာ အောက်ပါအတိုင်း လာပြပေးမှာပါ။ က စာဖတ်သူကိုယ်တိုင် Run လုပ်ပြီး လက်ထွေ လေ့လာကြည့်ပါ။



Figure 3.6 To Show the Output Windows in the upper-left corner

က အချက်ဖြစ်ထားတဲ့ JFrame ဟာ test.java ဆိတဲ့ program ကို Run လုပ်ရကာနေ ရရှိလာတဲ့ Output အနေနဲ့ ပြပေးတဲ့ Windows ပဲ ဖြစ်ပါတယ်။ ဒီ program ဟာ sample အနေနဲ့ရေးပြထားတာပါ။ အရေးသင်ခန်းစာတွေဖြစ်တဲ့ Chapter [1] & Chapter [2] တွေကိုသာ ကြောက်ညာကိုလေ့လာခဲ့မယ် ဆိုရင် အခုရေးပြခဲ့တဲ့ Sample program ဖြစ်တဲ့ Listing 3.1 ကို အလွယ်တကူ နားလည်သောပေါက်မှာပါ။ ကျွန်တော့ဘက် ကတော့ ထုံးစာတိုင်း Analysis လုပ်ပြီးတော့ ရှင်းပြပေးထားပါတယ်။ လေ့လာလိုက်ပါ။

#### Analysis

၁။ ဒီ program မှာ ရေးထားတဲ့ လိုင်းနံပါတ် (၁) က import javax.swing.\*; ဆိတဲ့ Java ရဲ့ swing components တွေကို ခေါ်ယူအသုံးချမှာဖြစ်တဲ့အတွက် package file ကို ခေါ်ယူလိုက်တာပါပဲ။ C & C + + programmer တွေရဲ့ စကားနဲ့ပြောရမယ်ဆိုရင်တော့ Header File တွေကို ခေါ်ယူ လိုက်တာပဲပဲ။ အဲဒီလိုပါပဲ။ လိုင်းနံပါတ် (၂) မှာ ကြညာ ထားတဲ့ import java.awt.\*; ဆိတဲ့ package ဟာလဲ awt components တွေကို ခေါ်ယူ အသုံးချချင်လို့ ခေါ်ယူထားတဲ့ package ပါပဲ။

Note : JAVA ရဲ့ Awt components တွေဟာ swing package components တွေနဲ့ ဆက်နှံယ်နေတဲ့အတွက် Declare လုပ်တဲ့အခါမှာ package File နှစ်ခုစာလုံးကို ကြညာပေးရတာပါ။ လိုအပ်လာရင် အသုံးချချင်ဖို့ အတွက်

```
import javax.swing.event.*;
import java.awt.event.*;
လို့ ခေါ်တဲ့ Event Listener တွေကိုပါ ထပ်ဖြည့်ပြီးတော့
Declare လုပ်ရမှာပါ။ အဲဒီလို Events တွေကို ဖြည့်စွက်
အသုံးချတာကို ရှေ့လာမယ့် chapter [3] ရဲ့ index
တွေမှာ သင်ကြား ပို့ချပေးမှာပါ။
```

J။ လိုင်းနံပါတ် (၃) ကတော့ test ဆိတဲ့ primary class ကို Declare လုပ်တာပါပဲ။ အထူးသတိပြုရမှာကတော့ ဒီ class ရဲ့ Name ဟာ save လုပ်ရမယ့် File\_Name နဲ့ အတူတူပဲ ဖြစ်ရပါမယ်။ ဒါဟာ အရေးကြီးဆုံး အချက်ပါ။ ဒီ class ကို နိုင်မြှင့်ပြီးတော့ Object ကို ဖန်တီးမှာကိုး။ အဲဒါကြောင့် ပါ။ Class ရဲ့ open brace ကို ဖွင့်ထားပြီးတော့ main () function ကို လိုင်းနံပါတ် (၅) မှာ ဖန်တီးထားပါတယ်။ အဲဒီ main () function ထဲမှာ JFrame Class ကို object ဖြစ် ဖန်တီးခေါ်ယူထားပါတယ်။ JFrame Class ကို ခေါ်ယူ အသုံးချချင်တဲ့အတွက်လဲ အပေါ်က Swing package နဲ့ awt package တွေကို import လုပ် ခေါ်ယူထားတာပဲ။ က JFrame Class ကို Object f အဖြစ် ဖန်တီးပြီးပြုဆိုရင် JFram Class (object f အဖြစ် ဖန်တီး ခေါ်ယူပြီးသားပါ) ထဲမှာ ပါဝင်တဲ့ setTitle () method ကို ခေါ်ယူ အသုံးချပါတယ်။ Object f ရဲ့ member ဖြစ်တာကြောင့် F.setTitle () ဆိုပြီးတော့ ခေါ်ယူပါတယ်။

```
F.setTitle ("JAVA Programming in Windows
Platform");
```

မိမိနှစ်သက်ရာ စာသား Text ကို Title Bar မှာ ပါဝင်စေရန် ရေးသားနိုင်ပါတယ်။

၂။ က လိုင်းနံပါတ် (၈) မှာတော့ ကျွန်တော်တို့ အခု ဖန်တီးနေ တဲ့ windows ရဲ့ Default Frame Size ကို ကိန်းသေ ကြညာပေးထားပါတယ်။ Default Size လို့ ပြောခြင်းက ကျွန်တော်တို့ အခု ဖန်တီးလိုက်တဲ့ program ကို ခေါ်ယူ Run လိုက်တိုင်း ဖော်ပြုပေးမယ့် Windows Frame Size ပဲ ဖြစ်ပါတယ်။ အဲဒီ set size () method ဟာလဲ JFrame Class ရဲ့ member method ဖြစ်တာကြောင့် Object\_Name F.လို့ ခံပြီးမှ ခေါ်ယူထားတာပါ။

```
F.setSize (300, 200);
```

မိမိနှစ်သက်သော အချယ်အစား width & Height ကို ရေးသားထည့်သွင်း၍ ဖန်တီးနိုင်သည်။

## ကွန်ပူးတာသင်ခန်းစာ

က ဒါဆိုရင်တော့ `f.setSize(300,200);` ဆိုပြီး ခေါ်ယူဖော်ထားတာကို စာဖတ်သူဘက်က နားလည် သဘောပေါက်သွားပြီးနော်။

၄။ လိုင်းနံပါတ် (၉) မှာကတော့ `f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);` method ကို ခေါ်ယူပြီးတော့ အသုံးချထားပါတယ်။ အရေးကြီးတဲ့ အသုံးပြုရမယ့် အချက်ကတော့ Method () ကို ခေါ်ယူအသုံးချတဲ့အချိန်မှာ argument အနေနဲ့ ပါဝင်တဲ့ `JFrame.EXIT_ON_CLOSE` ဆိုတဲ့ parameter arguments ကို အထူးကရပြု လေ့လာရမှာပါပဲ။ `JFrame` Class ကို F ဆိုတဲ့ object အဖြစ် ဖော်တိုးပြီးပြီ ဆိုပေမယ့် parameter arguments အနေနဲ့ ရေးသား အသုံးချရင်တော့ တိုက်ရှိကို မူလ `Class_Name` ဂိုပ် ခေါ်ယူရပါတယ်။ က လိုင်းနံပါတ် (၁၀) ကို ဆက်လက် လေ့လာကြည့်ရင်တော့ `f.show()` ဆိုပြီး `JFrame` Class ရဲ့ `show()` method ကို ခေါ်ယူအသုံးချထားပါတယ်။ ဒီ အချက်ကတော့ ရေးထားတဲ့ Windows Frame ကို Showလုပ်နိုင်ဖို့အတွက် ခေါ်ယူထားတာဖြစ်လို့ သိပ်ပြီးတော့ complex ဖြစ်စရာ မရှိပါဘူး။ အဲဒီနောက် လိုင်းနံပါတ် (၁၁) မှာကတော့ `main()` function ရဲ့ close brace } ကို ပိတ်ထားပြီးတော့ လိုင်းနံပါတ် (၁၂) မှာကတော့ primary class ဖြစ်တဲ့ `test` class ရဲ့ close brace ကို ပိတ်ထားပါတယ်။

က အချသင်ကြားပို့ချုံးတဲ့ listing 3.1 က sample program လေ့လာပြီးရင် စိတ်ကြိုက်အရွယ်အစား Set Title အနေနဲ့ မိမိနှစ်သက်ရာ program GUI Windows တွေကို အလွယ်တကူ ဖန်တီးနိုင်ကြပါတယ်။ အလွယ်တကူ Compareလုပ်ပြီး လေ့လာနိုင်အောင် Sample code\_line နဲ့ Note Pad မှာ ရေးသားထားပုံရယ်၊ Program Runလုပ်နေပုံရယ်ကို Figure 3.7 မှာ ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါပြီး။

```
test - Notepad
File Edit Format View Help
import javax.swing.*; // to call the swing java components
import java.awt.*;

public class test // this Class_Name is equal in this File_Name
{
public static void main (String args[])
{
    JFrame f=new JFrame(); // create object in JFrame
f.setTitle("JAVA Programming in windows platform"); // show text in Title Bar
f.setSize(300, 200); // windows size
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
f.show(); // to call show method ()
}
}
```

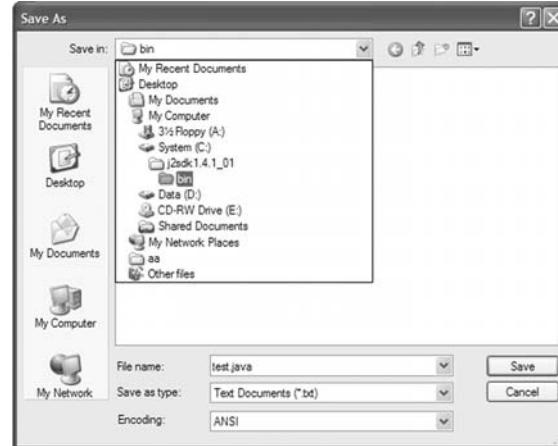


Figure 3.7 Write the program code-line In Note Pad, Save In and then Compile & Run it

အခြေခံကျွဲ့ JAVA Swing Component နဲ့ JAVA Awt Components တွေကို အခြေခံ တည်ဆောက်တဲ့အကြောင်းကို သဘောတရား နားလည်သွားပြီဆိုရင်တော့ ရှေ့သင်ခန်းစာအနေနဲ့ 3.2 Using the JAVA Swing Components ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် ပို့ချုပ်ထားပါတယ်။ စာဖတ်သူအနေနဲ့ သေသေချာချာ လေ့လာကြည့်ပါ။ အခြေခံကျွဲ့ တည်ဆောက်နည်းမောက်တတ်ကျွမ်း နားလည်လာပါလိမ့်မယ်။ က ဆက်လက်လေ့လာကြပါစို့။



### 3.2 USING THE JAVA SWING COMPONENTS AND AWT COMPONENTS

ဂျွဲန်တော် အချုပ်းဖြမယ့် အကြောင်းအရဟာ GUI Windows တွေကို ဖန်တီးရမှာ အလွန်အရေးပါတဲ့ အချက်တွေပါပဲ။ ဘာကြောင့်လဲဆိုတော့ Windows Interface ကို တည်ဆောက်ရမှာ (မပါမဖြစ်ပေါ့) တစ်ခုမဟုတ်တစ်ခု ထည့်သွင်း အသုံးပြုရတဲ့ Package file နှစ်ခု ရှိပါတယ်။ အဒီတွေကတော့ စာဖတ်သူသိရှိပြီးလောက်တဲ့ Java Language ရဲ့ အရင် Version 1.4 မှာ အထူး စတင်ကျော်ကြားလာတဲ့ -

```
import javax.swing.*;  
လို့ ခေါ်ယူ အသုံးပြုရတဲ့ Swing Components ရယ်။  
Java ရဲ့ ယင် Version အဟောင်းတွေမှာ Container  
Classes လို့ လူသိမှား ကျော်ကြားလှတဲ့ AWT components  
Package File -
```

```
သူကို ခေါ်ယူရမှာ  
import java.awt.*;  
လို့ ခေါ်ယူကြရပါတယ်။  
ဒီတော့ Java Programmer အများစုံဟာ GUI windows နဲ့ Windows Interface တွေကို တည်ဆောက်မယ်ဆိုရင် Swing Components နဲ့ AWT Components တွေကို ယုံ့တွဲပြီးတော့ အသုံးချကြပါတယ်။ တစ်ခါတစ်ရုံမှာ AWT Components Package File အပိုင်းကို လုံးဝဖြော်ထားကြလေ့လည်း ရှိပါတယ်။
```

```
အချို့သော Programmer တွေကလဲ AWT Component တွေကို အသုံးမပြုကြပေမယ့်  
import java.awt.*;  
import java.awt.event.*;  
ဒီပြီးတော့ program ထိပ်မှာ Declare & import လုပ်ထား  
တတ်ကြပါတယ်။
```

Note : ဥပမာပေးရရင်တော့ ဂျွဲန်တော် အခု အသုံးချနေတဲ့ သင်ကြားပေးတဲ့ program တွေမှာ Java Swing Components နဲ့ Java AWT Components တွေကိုပါ အသုံးချချင်လို့ package file နှစ်ခုစလုံးကို Declare & Import လုပ်ပြီးတော့ ခေါ်ယူထားတာပါ။ ဒါပေမယ့်



JAVA 21 days ထဲမှာတော့ Java ရဲ့ Swing Components အပိုင်းတွေကိုပဲ အဓိက အသုံးချချင်တဲ့အတွက် Declare & Import လုပ်တဲ့အခါမှာ import javax.swing.\*; ဆိုပြီးမှ Swing Package file ကိုပဲ ခေါ်ယူပါတယ်။ JAVA 21 days ဟာ sam publication က ထုတ်ဝေပြီး ဒီနေ့တော် ကျောင်းသားတော်တော်များများ အထူးပြု လေ့လာနေတဲ့ စာအုပ်ပါပဲ။ အဲဒီစာအုပ်ရဲ့ Chapter [2], Days -9 မှာ သွားကြည့်ရင် တွေ့မြင်နိုင်ပါတယ်။ လေ့လာကြည့်ပါ။

ဂျွဲန်တော်ကတော့ Swing Components နဲ့ AWT components နှစ်မျိုးစလုံးကို ရောဖွေပြီးတော့ အသုံးချချင်ပါတယ်။ ဒီတော့ Components တွေအကြောင်း အရင် လေ့လာကြည့်ကြရအောင်လား။

#### ◆ AWT Components <container classes>

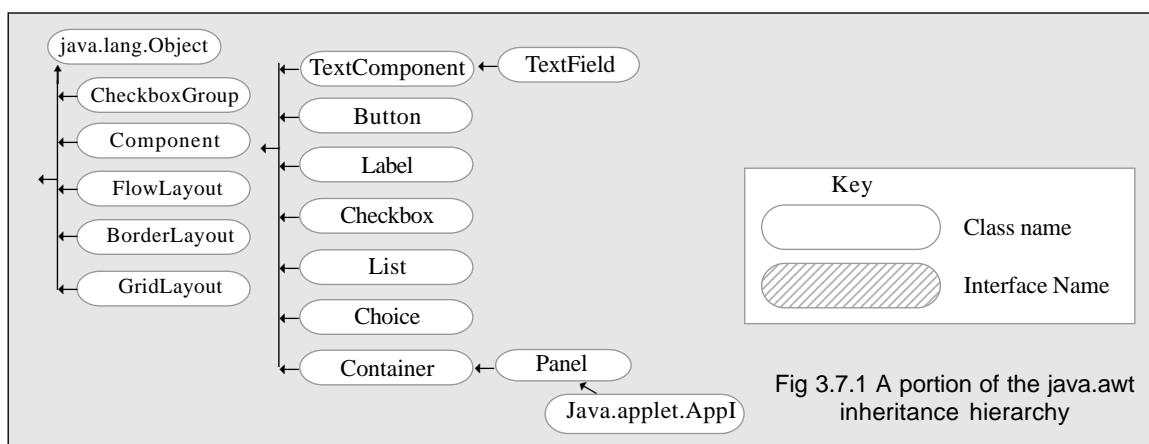
- AWT components ဆိုတာကတော့ JAVA Language မှာ GUI လိုပေါ်တဲ့ Graphical User Interface တွေ တည်ဆောက်ဖို့ပါပဲ။
- Graphical User Interface ဆိုတာတွေကတော့ menus, buttons, lists, drop down lists, frames, Dialogs Box အစုံးစုံ components တွေဟာ java.awt package အတွင်းမှာ ပါဝင်ပါတယ်။
- AWT Components ရဲ့ User-Interface ဟာ Interface သွေ့ပြင်အရ Platform မျိုးစုံနဲ့ သင့်တော်ပြီးတော့ Applet & Application type တွေအနေနဲ့ excuted လုပ်ပေးနိုင်ပါတယ်။

ဒီအချက်တွေဟာ AWT components နဲ့ပါတ်တပ်တာတွေကို အကြမ်းအားဖြင့် ရှင်းပြတာပဲ ဖြစ်ပါတယ်။ အဲဒီ awt components တွေကို လေ့လာကြည့်နိုင်ဖို့အတွက် Basic GUI Components ဆိုပြီးတော့ AWT (container class) ထဲမှာပါဝင်တဲ့ Components Tools တွေကို ဖော်ပြပေးထားပါတယ်။ Table 3.1 အနေနဲ့ပဲ လေ့လာကြည့်ပါ။

Component	Description
Label	An area where uneditable text can be displayed.
Button	An area that triggers an event when clicked.
TextField	An area in which the user inputs data from the keyboard. The area can also display information.
Choice	A drop-down list of items from which the user can make a selection by clicking on an item in the list.
Checkbox	A boolean GUI component that is either selected or not selected.
List	An area where a list of items is displayed from which the user can make a selection by clicking once on any element in the list. Double-clicking an element in the list generates an action event.
Panel	A container in which components can be placed.

Table 3.1 Basic GUI components

က အခု အပေါ်မှာရေးပြောတဲ့ Table 3.1 ဟာ container class ထဲမှာပါဝင်တဲ့ components Tools တွေကို လေ့လာကြည့်နိုင်ပါပဲ။ ဒီတော့ အဲဒီ AWT components တွေရဲ့ object inheritance အဖြစ် တည်ရှိပုံကို ဆက်လက် လေ့လာကြည့်ကြရအောင်လား။



က အခုဖော်ပြောတဲ့ Figure 3.7.1 ဟာ java.awt package ရဲ့ components တွေဟာ object hierarchy (Inheritance) အဖြစ် တည်ရှိနေပုံကို ဖြင့်ရတာပါပဲ။ က ဒါဆိုရင်တော့ အဲဒီ out package ထဲမှာပါဝင်တဲ့ Label, Button, TextField, Choice, Checkbox, List, Panel အစရှိတဲ့ components တစ်ခုချင်းစိတ် အထူးပြု လေ့လာကြရအောင်လား။ ဒါမှသာလျှင် Program တွေ ဖုန်းတော်သားတဲ့အခါမှာ လိုအပ်တဲ့ features တွေကို အသုံးချဖိုင်မှာဖြစ်တယ်။

#### ◆ Class TextField constructors and methods

အခု ရှင်းပြုမှာကတော့ TextField အကြောင်းပဲ ဖြစ်ပါတယ်။ အခိုက ရှင်းပြုချင်တာက component TextField ကို ခေါ်ယူ အသုံးချင် ဘယ်လိုစွမ်းဆောင်ရည်ရှိမလဲ။ ပြီးတော့ arguments & parameter အနေနဲ့ ဘယ်လို ထည့်သွင်းရေးသားလို့ရမယ်။ အစရှိသဖြင့်ပေါ်နေ၏။ သိရှိနားလည်အောင်လို့ TextField Class အကြောင်းကို Figure အနေနဲ့ ရေးသားပြီးတော့ ဖော်ပြပေးထား

ပါတယ်။ အရင်ဆုံး လေ့လာလိုက်ပါပြီး။

**Note :** ရေးသားထားတာကတော့ မူရင်း: JAVA Developer Guide မှာ ပါဝင်တဲ့ Figure အတိုင်း စာသားအတိုင်း ဖော်ပြပေးထားပါတယ်။ Program ရေးပြီးတော့ Listing အနေနဲ့ ရေးသားတဲ့အခါကျမှုသာလျှင် TextField class အကြောင်းကို အလွယ်တကူ နားလည် သဘောပေါက် အောင် ရှင်းပြပေးပါမယ်။ အခုရေးသားထားတဲ့ TextField Class အကြောင်းကို ဖော်ပြပေးထားတဲ့ Table 3.2 ကိုတော့ သေသေချာချာ လေ့လာပါ။ တကယ်လို့ စာဖော်သူဟာ Java Certified ဖြစ်နိုင် ရည်ရွယ်တယ်ဆို ရင် အခု ပို့ချေပေးတဲ့ Table တွေဟာ Exam ကျရင် အထောက်အကူ အများကြီး ရောနိုင်မှာပါ။

TextField ကို အသုံးပြုထားတဲ့ ဥပမာပြ program ကို listing 3.2 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ အဲဒီ Listing ကျရင် TextField ရဲ့ features ကို ဖြန်မှာလိုပြည့်စုံအောင် ရေးသားလို့ချပေးထားပါတယ်။

Class TextField constructors and methods
<pre>public TextField( )     Constructs a TextField object. public TextField(int columns)           //number of columns     Constructs an empty TextField object with the specified number of columns. public TextField(String s)              //text displayed in TextField     Constructs a TextField object displaying s. public TextField(String s,             //text displayed in TextField int columns)                         //number of columns     Constructs a TextField object displaying s in the specified number of columns. public void setEchoChar(char c)        // character to echo     Sets the character to be displayed as the user types in the TextField. This is particularly     useful for disguising a typed in password. public void setEditable(boolean b)     // true=editable     Sets the editability of the TextField, true is editable, false is uneditable.</pre>

Table 3.2 Class TextField constructors and methods

အချက်ပြထားတဲ့ Table 3.2 ဟာ TextField Class အတွင်းက constructor and method တွေအကြောင်းကိုပဲ အပြည့်စုံမှုးဖော်ပြုပေးထားတဲ့ Table ပါပဲ။ အသုံးပြုရမယ့် အနေ အထား ပါဝင်တဲ့ Theory သဘောတရားတွေကို ဖြန့်မှုလိုက် တော့ ရှုံးလာမယ့်သင်ခန်းစာ 3.2 ရဲ့ sub-Index ကျရင် ပြည့်စုံ

အောင် တတ်နိုင်သမျှ ရေးသားပေးထားပါတယ်။ ဒီတော့ အခါ Table 3.2 ကိုတော့ မျက်စိတ် စွဲနေအောင်၊ အလွတ်ရအောင် လေ့လာထားလိုက်ပါတီ။

က ဆက်လက်ပြီးတော့ Lable Class နဲ့ ပတ်သက်တဲ့ Table 3.3 ကို ဆက်လက်လေ့လာကြရအောင်လား။

Class Label constructors and methods
<pre>public Label( )     Constructs an empty Label - text is not displayed. public Label(String s)           //label text     Constructs a Label that displays the text s with default left-justified alignment. public Label(String s,           //label text int alignment)                 //alignment     Constructs a Label that displays the text s with the specified alignment. Possible alignments     are Label.LEFT, Label.CENTER and Label.RIGHT public String getText( )     Returns the Label text. public void setText(String s)    //read-only string to display     Sets the Label text public void setAlignment(int alignment) //alignment     Sets the Label alignment to Label.LEFT, Label.CENTER or Label.RIGHT</pre>

Table 3.3 Class label constructors and methods

#### ◆ Class Label constructor and methods

အချက်ပြနဲ့တဲ့ Table 3.3 ကတော့ label class ကို ခေါ်ယူ အသုံးပြုရင် ပါဝင် ထည့်သွင်းထားတဲ့ အသုံးပြုလိုရတဲ့ method() တွေ ရယ် ပါဝင်ထည့်သွင်းလို့ရမယ့် arguments and parameter တွေအကြောင်းပါပဲ။ မှတ်မိအောင် သေသေချာချာ လေ့လာထားပါ။

Note : Label ကို ခေါ်ယူ အသုံးပြုထားတဲ့ Listing ကတော့ Listing 3.3 မှာ စာဖတ်သူ လေ့လာရမှာပါပဲ။ Label Box ကို အသုံးပြထားပြီးတော့ အဲဒီ label class ရဲ့ features တွေအကြောင်းကို ဖြန့်မှုလို ရင်းပြပေးထားပါတယ်။ အခုကတော့ Table 3.3 ဟာ label class ရဲ့ constructor and methods တွေအကြောင်းကို မူရင်း JAVA Developer Guide ထဲကအတိုင်း ဖော်ပြပေးထားတာပါပဲ။ Label Box တွေကို ထည့်သွင်း တပ်ဆင်

နဲ့ Label() method တင်မကာဘဲ Label Box ပေါ်မှာ ပါဝင်ရမယ့် ထည့်သွင်းထားတဲ့ စာသားတွေကို Alignment အနေနဲ့ ဘယ်ဘက်ကပ်၊ ညာဘက်ကပ် အစရိုး သဖြင့် Alignment ဖြို့ဖို့ စာသားတွေ ထည့်သွင်းဖို့ getText() method အစရိုးတာတွေကိုပါ စာဖတ်သူ တွေမြင်ရမှာပါ။ အဲဒီ method တွေ အားလုံးကိုပါ ပြည့်စုံအောင် သေသေချာချာ လေ့လာပါ။

#### ◆ Text Area Class Constructor

TextArea Class နဲ့ ပတ်သက်ပြီးတော့ အသုံးပြုလိုရမယ့် method() တွေရယ် argument (or) parameter type တွေကို ဘယ်လိုလက်ခံအသုံးပြုမလဲဆိုတာကို လေ့လာနိုင်ဖို့အတွက် Table 3.4 အနေနဲ့ ရေးသားဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

<b>TextArea class constructors</b>
<pre>public TextArea()     Constructs a TextArea with default size(normally controlled by the layout of     the container to which it is attached). public TextArea(int rows, int columns)     Constructs a TextArea with the specified number of rows and columns. public TextArea(String s)     Constructs a TextArea containing the text s. public TextArea(String s,int rows,int columns)     Constructs a TextArea containing the text s and the specified number of rows and     columns. public TextArea(String s,int rows,int columns,int scrollbars)     Constructs a TextArea containing text s and the specified number of rows and columns.     The scrollbars argument specifies which scrollbars should be visible using one     of the TextArea constants: SCROLLBARS_BOTH,SCROLLBARS_HORIZONTAL_ONLY,     SCROLLBARS_NONE or SCROLLBARS_VERTICAL_ONLY.</pre>

Table 3.4 TextArea class constructors

Note : TextArea နဲ့ပတ်သက်တဲ့ Program အနေနဲ့ Listing 3.5 အနေနဲ့ ရေးသား ပို့ချေပြေးထားပါတယ်။ ဒါကြောင် Table 3.4 အနေနဲ့ပဲ သက်ဆိုင်ရာ method() တွေကိုလဲ လေးလာလိုက်ပါဉိုး။

◆ Checkbox and Checkbox Group constructors  
Application & Applet type Programs တွေကို ရေး

သား ဖုန်တီးရာမှာ အမိက မပါမဖြစ်တဲ့ checkbox တွေကို ဖုန်တီးရေးသားချင်ရင် အသုံးပြု ခေါ်ယူရမယ့် Class ကတော့ Checkbox Class ပါပဲ။

ဒီတော့ Checkbox Class နဲ့ပတ်သက်တဲ့ Table 3.5 အနေနဲ့ ရေးသားဖော်ပြပေးထားပါတယ်။ ဒီတော့ ဘယ်လို method() တွေ ပါဝင်တယ်ဆိုတာကို လေးလာလိုက်ပါဉိုး။

<b>Checkbox and CheckboxGroup constructors</b>
<pre>public Checkbox(String s)     Constructs a Checkbox object with label s. The Checkbox object is initially     unchecked. public Checkbox{     String s                      //radio button label     CheckboxGroup c,              //CheckboxGroup owning radio button     boolean state)               //state of radio button     Constructs a single radio button with the label s and the booleanstate.The radio     button is added to the existing CheckboxGroup C. If c is null,this constructor     creates a normal Checkbox object with the specified state. public CheckboxGroup()     Constructs a CheckboxGroup object.</pre>

Table 3.5 Checkbox and Checkbox Group constructors.

Note : Checkbox နဲ့ ပတ်သက်တဲ့ program အနေနဲ့တော့ Listing 3.6 အနေနဲ့ ရေးသားဖော်ပြပေးထားပါတယ်။ ကဲဒဲ့ဆိုရင် တော့ ကျော်တော်ရှင်းပြပေးတာဟာ AWT Components ရဲ့ အမိက ပင်မ Class တွေအကြောင်းကတော့ ပို့ချေပြီး ဖြစ်သွားပါပြီ။ ကျော်တဲ့ components class တွေဖြစ်တဲ့ Button, Choice, List, Radio Button/ Option Button အစရိတ်တဲ့ Class တွေကိုတော့ သက်ဆိုင်ရာ Listing တွေနဲ့အတူ program ထံလေးတွေ ရေးပြပြီးမှ ရှင်းပြပေးပါတော့မယ်။

အခု လက်ရှိသင်ခန်းစာကို အထပ်ထပ်အခါခါ လေးလာပြီးလို့ ကဲပြား မြားနားနေတဲ့ Class Method() တွေကို နားလည် သဘောပေါက်သွားပြီးရင်တော့ အခုလက်ရှိ Chapter Index 3.2 Using the JAVA Swing Components & AWT components ဆိုတဲ့ သင်ခန်းစာရဲ့ sub-Index အဖြစ် သက်ဆိုင်ရာ components Tools တစ်ခုချင်းစီအလုံကဲ ပို့ချေပြေးထားပါတယ်။ ဆက်လက် လေးလာကြပါစိုး။

KYAW ZAYAR LAY [ IT ENGINEERING ]



### 3.2.1 USING THE JAVA SWING COMPONENTS AND AWT COMPONENTS

#### ◆ Text Field (OR) JTextField

Java ရဲ့ awt components တွေထိ ခေါ်ယူ အသုံးပြုရမှာ JTextField ကို ခေါ်ယူအသုံးပြုတဲ့အကြောင်းကို ရှင်းပြုပါမယ်။ အချို့သော Java Programmer တွေကတော့ JTextField ကို JTextField , Label Box ကို JLabel အစရှိသဖြင့် ခေါ်ပေါ်ကြပါတယ်။ အဲဒါတွေကတော့ Swing components ရဲ့ အခေါ်အနေနဲ့ပါပဲ။

Note: Visual Basic Programming Language မှာ ပါဝင်တဲ့ Control ToolBox ထဲက Tools တွေအတိုင်း Java Language မှာလည်း Tools တွေပါဝင်ပါတယ်။ အခေါ်အနေ Naming အချို့ပဲ ကွဲပြားတာပါ။ ဥပမာပြာရရင်တော့ JTextField လို့ Java Language မှာ ခေါ်ဝေါ်ပေမယ့် Visual Basic Language မှာတော့ TextBox လို့ ခေါ်ဝေါ်သလိုပါပဲ။

TextField ဒါမှမဟုတ် JTextField ဆိုတာကတော့ Visual Basic Language က TextBox အတိုင်းပါပဲ။ Visual Basic Language မှာတော့ Coding လိုပေါ်တဲ့ code တွေ မရေးခေါင်မှာ Tool Box ထဲကနေ Drug လုပ် ဆွဲချုပြုးတော့ ဒါမှမဟုတ် D-Click ခေါက်ပြီးတော့ Form Windows ထဲမှာ အရင် တည်ဆောက်ရပါတယ်။

#### ◆ TextField (OR) JTextField ဆိုတာဟာ

- User (အသုံးပြုသွောက) User Information တွေကို keyboard ကနေ Input လုပ် ထည့်သွင်းနိုင်ဖို့ပါပဲ။
- ခေါ်ယူအသုံးပြုချင်ရင်  
import javax.swing.\*;  
import java.awt.\*;
- အစရှိတဲ့ Java Package File တွေကို အရင်ဆုံး Import လုပ်ပြီးတော့ ခေါ်ယူထားရပါမယ်။
- JPanel Class ကို extends လုပ်ရပါမယ်။ JTextField

(OR) JTextField, JLabel (OR) Label အစရှိတဲ့ Component Class တွေဟာ JPanel Class အောက်မှာ တည်ရှိလိုပါပဲ။ AWT container Class မှာလည်း မှားဝင်လိုပါပဲ။

- မိမိဖန်တီးမယ့် Class \_Name နဲ့အတူ constructor () method ကို ရေးသားပြီးတော့ အဲဒီ constructor ထဲမှာ Components တွေကို တပ်ဆင် ရေးသား ထည့်သွင်းပါတယ်။
- File ကို Compile & Run လုပ်ဖို့ Save လုပ်တဲ့အခါ အဲဒီ Class \_Name အတိုင်း Save လုပ်ရပါမယ်။ Class \_Name = File \_Name ရဲ့နောက်မှာ .JAVA (OR).java လိုပေါ်တဲ့ extension ကို ထည့်သွင်း save လုပ်ရပါမယ်။

Listing 3.2 အနေနဲ့ JTextField ကို အသုံးပြု လေ့ကျင့်နိုင်ပါ Program တဲ့ခုကို တည်ဆောက်ပြပါမယ်။ Program code \_line တွေကို ရေးပြထားပါတယ်။ အဲဒီ program code \_line တွေကို သွားရောက်ရေးသားရမယ့် Text Editor ကတော့ windows နဲ့အတူ Built-in ပါလာတဲ့ NotePad Program မှာ သွားပြီး ရေးသားရပါမယ်။

#### Listing 3.2 Testing the JAVA TextField

1. import javax.swing.\*; // Declare import the javax.swing package
2. import java.awt.\*; // Declare import the java. awt package
3. public class textfield extends JPanel
4. {
5. public textfield () // constructor method () in textfield class
6. {
7. setLayout (null) ;
8. JTextField tf\_1= new JTextField ("Enter Text" , 50) ;
9. JTextField tf\_2 = new JTextField ("Enter Number",20);
10. JTextField tf\_3 = new JTextField ("enter comments",200);

## ကွန်ပျူးတာသင်ခန်းစာ

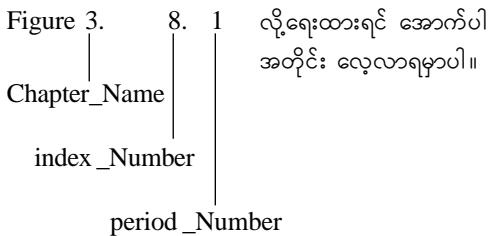
```

11.
12. tf 1. SetBounds (30,10,150,30); // x,y, width,
    Height
13. tf 2. SetBounds (30,50,150,30);
14. tf 3. SetBounds (30,90,150,100) ;
15.
16. add (tf 1);
17. add (tf 2);
18. add (tf 3);
19.
20. public static void main (string args [ ]) // write the
    main ( ) method
21. {
22. // to show the title bar set Text
23. JFrame f = new JFrame ("Testing the JAVA
    TextField" );
24.
25. textfield itf = new textfield () ; // create objects
26. f. setsize (300,200) ; // width and Height of Win-
    dows Frame size
27. f. get Content Pane ( ) .add (jtf) ;
28.
29. f. Set Default Close Operation ( JFrame .EXIT_
    ON_CLOSE ) ;
30. F. showe ( );
31. } // close brace } in main ( )method
32. } // close brace } in Primary class textfield

```

အခု အပေါ်မှာရေးပြုခဲ့တဲ့ code\_line တွေကို NotePad မှာ ရိုက်သွင်းပြီးတော့ Save လုပ်ပါ။ Save လုပ်ရာမှာ ပြီးခဲ့တဲ့ သင်ခန်းစာမှာ ပြောခဲ့သလို C:\j2sdk 1.4.1\_01\bin> ဆိတ္တာ path လမ်းကြောင်းအောက်မှာ သွားပြီးတော့ save လုပ်ပါမယ်။ ပြီးတော့ save လုပ်ရာမှာ File\_Name ကို Class \_ Name နဲ့ အတူတူ textfield.java လို့ အမည်ပေးရပါမယ်။ NotePad ထဲမှာ code\_line တွေ စာရိုက်ထားပဲကို Figure 3.8.1 အနေနဲ့ ရေးပြထားပါတယ်။

Note: ပြီးခဲ့တဲ့ သင်ခန်းစာ 3.1 မှာက Figure တွေကို ဖော်ပြရာ မှာ Caption တွေကို 3.7 အထိ နံပါတ်စဉ်ထိုးပြီး ဖော်ပြုခဲ့တာပါ။ ဒါပေမယ့် ဒီ listing 3.2 မှာတော့ Figure တွေကို 3.8.1 ကနေပြီးတော့ sub\_index လေးတွေ ခွဲရေးပြီးတော့ ဖော်ပြုပေးထားပါတယ်။ ဥပမာ ဆိုတဲ့ပါ၌။



Chapter\_Name ကတော့ ရောက်ရှိ သင်ကြားနေတဲ့ Chapter အခန်းကလွှာပါပဲ။ index\_Number ကတော့ ရောက်ခဲ့ listing သို့မဟုတ် ငါးချွန်တဲ့ သင်ခန်းစာအပိုင်းပါပဲ။ အဲဒီရဲ့ နောက်မှာ period\_Number ပါနေသရွှေ့ လက်ရှိ listing သို့မဟုတ် လက်ရှိသင်ခန်းစာရဲ့ သက်ဆိုင်တဲ့ Figure တွေလို့ပဲ မှတ်ယူပါ။ ကဲ ဒီတော့ Figure 3.8.1 ကို လေ့လာကြည့်လိုက်ပါတီး။ အဲဒီ NotePad File ကို textfield.java လို့ အမည်ပေးရမှာနော်။

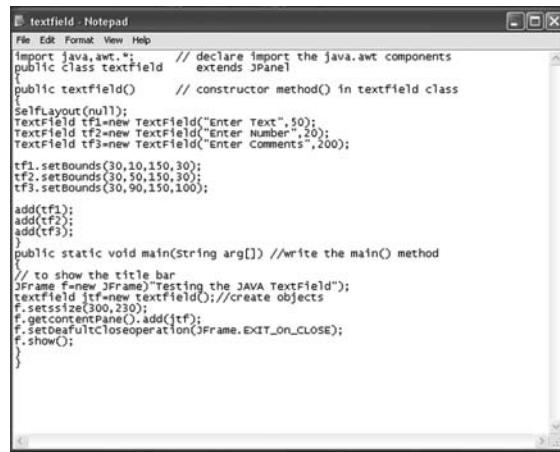


Figure 3.8.1 To show the Program Code\_Line in NotePad

ပြီးရင်တော့ command prompt <OR> Dos ကို သွားဖွို့ပါ။ အောက်မှာရေးပြထားတဲ့ command-line ကို ရိုက်သွင်းပြီးတော့ Enter key ကို နိုင်ပါ။

CD C:\j2sdk 1.4.1\_01\bin ↵

ဒီအခါမှာ Command prompt Interface မှာ

C:\j2sdk 1.4.1\_01\bin>

လို့ prompt ပေါ်နေချိန်မှာ အောက်ပါ command-line ကို ရိုက် သွင်းပြီးတော့ Compile လုပ်ပါ။

javactextfield .java ↵

အဲဒီလို့ compile လုပ်တဲ့အခါမှာ error တက်လာရင်လည်း NotePad File ထဲဝင်ပြီးတော့ ကျွန်တော်ရေးပြုခဲ့တဲ့ listing 3.2 က program code\_line တွေနဲ့ တိုက်ဆိုင်စစ်ဆေးပြီးတော့ ဖြစ်ပြင်ပါ။ ပြီးရင် Save As ကို File → Save As ကနေ တော်ပြီးတော့ Save As လုပ်ရပါမယ်။ အမည်တွေ၊ ဘာတွေ ပြောင်းစရာ မလိုဘူးနော်။ အဲဒီအောက်

javactextfield .java ↵

လို့ command\_line ကို ရိုက်သွင်းပြီးတော့ compile ပြန်လုပ်ပါ။ တကယ်လို့ error ထပ်တက်လာရင်လည်း အခုလိုပဲ NotePad

ထမှာ ပြပြင်ပြီးတော့ Save As လုပ်ပြီးတော့ Command Prompt <DOS>ထဲမှာပြန်ပြီးတော့ Compile လုပ်ရမှာပါပဲ။

Compile လုပ်ပဲနဲ့ error မရှိဘဲ အောင်မြင်သွားရင် C:\j2sdk 1.4.1\_01\bin >ဆိုတဲ့ prompt ပဲ ဖြန်ပေါ်လာမှာပါ။ ကဲ အခု ကျွန်တော် ရေးပြထားတဲ့အတိုင်း Compile လုပ်တာ အောင်မြင်သွားရင်တော့ အဲဒီ command prompt ထမှာ-

`java textField`

ဆိုတဲ့ text command ကို ရိုက်သွင်းပြီးတော့ Run လုပ်ပါ။ အဲဒီလို compile လုပ်ပြီးတော့ Run လုပ်ပဲကို Figure 3.8.2 မှာ ဖော်ပြပေးထားပါတယ်။

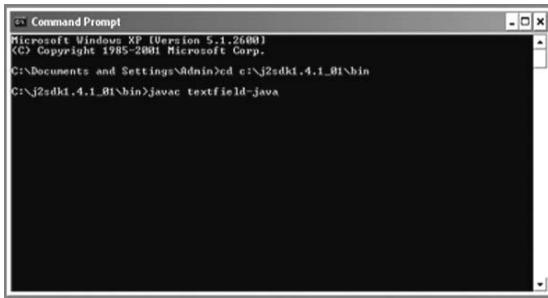


Figure 3.8.2 Compile & Run of the textField .java program

အခုလို compile လုပ်ပြီးတော့ Run လိုက်ရင် စာဖတ်သူရဲ့ Computer Desktop screen ရဲ့ ဘယ်ဘက် အပေါ်ထောင် Upper left corner မှာ output console windows ဟာ လာပြပေးမှာပါ။ Output Windows ဖုစ်ကို Figure 3.8.3 အနေနဲ့ ဖော်ပြပေးထားပါတယ်။

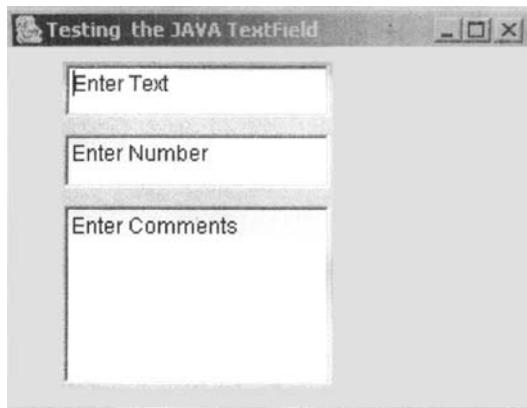


Figure 3.8.3 Run & Show of the Output Windows, textField .java programs

ကဲအခု listing 3.2 ကို စာဖတ်သူ လက်တွေ့ compile လုပ်၊ Run လုပ် လေ့လာပြီးရင်တော့ Program code-line ထွေကို Analysis လုပ်ပြီးတော့ လေ့လာကြတာပေါ့။ ဒါမှသာလျင် program code တွေ ရေးသားပဲတွေကို အမှန်တကယ် နားလည်တတ်မောက်မှာပါပဲ။

#### Analysis

၁။ လိုင်းနံပါတ် (၁) နဲ့ လိုင်းနံပါတ် (၂) မှာ-

`import javax.swing.*;`

`import java.awt.*;`

ဆိုတဲ့ package file နှစ်ခုကို import လုပ် ခေါ်ယူထားပါတယ်။ လိုင်းနံပါတ် (၃) မှာမဲ့ public class textField extends JPanel ဆိုပြီးတော့ textField ဆိုတဲ့ primary class ကို ဖန်တီး ရေးသားပါတယ်။ textField class ကို ရေးသားရာမှာ JPanel Class ကို လုမ်းပြီးတော့ extends လုပ်ထားပါတယ်။

**Note :** extends လုပ်ပဲနဲ့ Interfaces တို့အကြောင်းကို chapter [2]မှာ အကျယ်တောင် ရှင်းလင်းပို့ချ ပေးနဲ့ပြီးပါပြီ။ အဲဒီသင်ခန်းစာတွေကိုသာ နားမလည်ခဲ့ရင် အခုအချိန်မှာ ရှင်းပြရ တော်တော်ခက်ခဲမှာပါ။ ဒါကြောင့် နားမလည်သေးရင် ပြန်လှန် လေ့လာစေချင်ပါတယ်။

လိုင်းနံပါတ် (၄) ကတော့ textField ဆိုတဲ့ primary class ရဲ့ အဖွင့် open brace { ပါပဲ။ လိုင်းနံပါတ် (၅) မှာတော့ textField () ဆိုတဲ့ constructor method ကို ရေးသားပါတယ်။ အဲဒီ constructor method ထဲမှာပဲ Textfield နဲ့ ပတ်သက်တာတွေ ကို ရေးသားမှာ ဖြစ်ပါတယ်။

၂။ လိုင်းနံပါတ် (၆) မှာတော့ setLayout() method ကို ခေါ်ယူအသုံးပြုထားပါတယ်။ SetLayout() method ရဲ့ parameter value တန်ဖိုးကို null လို့ ပေးထားပါတယ်။ SetLayout() method ကို ခေါ်ယူရခြင်းဟာ လိုင်းနံပါတ် (12), (13), (14)မှာ ခေါ်ယူထားတဲ့ SetBound() method ကို ခေါ်ယူအသုံးချချင်လို့ပါပဲ။ တစ်နည်းပြောရရင်တော့ SetBound() method မှာ arguments (4) ခု ပါဝင်ပါတယ်။ အဲဒီကို စွဲမြားပြရရင်တော့-

`SetBounds (int x, int y, int width, int Height)`

ဆိုပြီး စွဲမြားပြီးတော့ လေ့လာကြည့်နိုင်ပါတယ်။ ရှင်းသွားအောင် လေ့လာမယ်ဆိုရင်-

## ကွန်ပျူးတာသင်ခန်းစာ

```

SetBounds (int x, int y, int width, int height);
    ↑           ↑           ↑           ↑
textfield တည်ရှိမည့်      textfield တည်ရှိမည့်      textfield တည်ရှိမည့်      textfield တည်ရှိမည့်
x ဝင်ရှုံးမှတ်          y ဝင်ရှုံးမှတ်          width အလျားထုတ်          height အကျယ်အဝန်း
                           textfield ၏ အလျားထုတ်
                           အကျယ်အဝန်း
                           textfield ၏ အဖြစ်
                           ဒေါင်ဂိုက်အချုပ်အစား
                           အစရိတ်ဖြင့် ခွဲခြား လေ့လာလို့ ရပါတယ်။

```

၃။ လိုင်းနံပါတ် (12),(13),(14) မှာတော့-

```

tf1 .SetBounds (30,10,150,30);
tf2 .SetBounds (30,50, 150,30);
tf3 .SetBounds (30,90,150,100);

```

ဆိုပြီးတော့ textfield (3) ခုကို တပ်ဆင်ပါတယ်။ အပေါက် textfield () ခုဟာ အချုပ်အစား တူညီပါတယ်။ တတိယမြောက် textfield ကသာ အချုပ်အစား ကျန်းနှစ်ခုထက် ကြိုးတာပါ။ ဒါတော့ ခွဲခြားလေ့လာကြည့်ရင် -

```

textfield နှစ်ခု
    | tf 1. setBounds (30, 150, 30); - y
    |   ဝင်ရှုံးမှတ်နေရာ 10 အဖြစ်သတ်မှတ်
    | tf 2.setBounds (30, 50, 150, 30); - y
    |   ဝင်ရှုံးမှတ်နေရာ 50 အဖြစ်သတ်မှတ်

```

y ဝင်ရှုံးမှတ် ကွဲပြားသောကြောင့် သာ နေရာကွဲပြီး ဖော်ပြပေးတာပါ။ သေသေချာချာ လေ့လာကြည့်ပါ။ လိုင်းနံပါတ် (၁၄) မှာ တပ်ဆင်ထားတဲ့ tf3. setBounds (30,90,150,100); ဆိုတဲ့ textfield ကို လေ့လာကြည့်ရင် -

```

tf 3. SetBounds (30, 90, 150, 100);
    ↑           ↑           ↑           ↑
    y ဝင်ရှုံးမှတ်          height အချုပ်အစား: Height

```

ဆိုပြီးတော့ ကွဲပြားခြားနားတာကို တွေ့ဖြင့်ရမှာပါ။ ဒါဟာ textfield ရဲ့ အချုပ်အစား: Height ကို အကျယ်ချုံထားတာပါပဲ။

က ဒါဆိုရင် SetBounds () method တွေ အသုံးပြုထားတာကို တွေ့ရပြီးနော်။ လိုင်းနံပါတ် (၁၅) ကတော့ line space ကို blank ချိန်ထားတာပါပဲ။

၄။ တကယ်တော့ set Bounds () method ကို ခေါ်ယူ အသုံးပြုချင်လို့ လိုင်းနံပါတ် (၇) က SetLayout (null) method ကို ခေါ်ယူ အသုံးပြုထားတာပါ။ ဒါ မီ method သာမပါရင်တော့ setBounds () method ကို ခေါ်ယူအသုံးပြုပေမယ့် အလုပ်လုပ်မှာ မဟုတ်ပါဘူး။

ဒါအပြင် setBounds () method ကို ခေါ်ယူ အသုံးပြုထားပေမယ့် add () method ကို ခေါ်ယူ အသုံးမပြုခဲ့ရင် အလုပ်လုပ်မှာ မဟုတ်ပါဘူး။ ဒါကြောင့်လည်း လိုင်းနံပါတ် (16), (17), (18) မှာ

```

add (tf 1);
add (tf 2);
add (tf 3);

```

ဆိုပြီးတော့ ခေါ်ယူ အသုံးချထားပါတယ်။ ဒီအချက်တွေရဲ့ ဆက်စပ်ပုံကို စာဖတ်သူအနေနဲ့ သေသေချာချာ လေ့လာကြည့်ပါ။ code\_line ဖွဲ့စည်းထားပုံတွေဟာ အလွန် ကျစ်လှစ်ပါတယ်။ လိုင်းနံပါတ် (19) ကတော့ textfield () ဆိုတဲ့ constructor method ရဲ့ Close brace } ပါပဲ။ ဒီအထိရပြီးနော်။

၅။ လိုင်းနံပါတ် (20) မှာတော့ main () function ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (23) ကတော့ JFrame Class ကို Object\_Name f လို့ အမည်ပေးပြီးတော့ ခေါ်ယူ အသုံးချပါတယ်။ JFrame ရဲ့ titlebar မှာပါဝင်ရမယ့် စာသား Text တွေကို JFrame Class ရဲ့ constructor method () မှာ ထည့်ရေးထားပါတယ်။ ဒါကြောင့်

JFrame f = new JFrame ("Testing the JAVA TextField");  
ဆိုပြီး ရေးထားတာပါ။

title bar မှာ ပါဝင်စေချင်တဲ့ text တွေ ဖြစ်ပါတယ်။

က ပြီးတော့ JFrame Class ဟာ object f လို့ ဖြစ်သွားတဲ့ အတွက် အော့ဒီ JFrame ရဲ့ အချုပ်အစားကို ချိန်ညွှန်တဲ့အခါမှာ F. setSize (300,200); // width, Height

လို့ ခေါ်ယူ ချိန်ညွှန်ထားတာပါ။ အော့ဒီ code\_line ကို လိုင်းနံပါတ် (26) မှာ တွေ့မြင်ရမှာပါ။ လိုင်းနံပါတ် (25) ကတော့ textfield class ကို ခေါ်ယူပြီးတော့ jtf ဆိုပြီးတော့ object အဖြစ် ဖန်တီးထားပြန်ပါတယ်။ ဒါမှာသာ textfield တွေကို ခေါ်ယူအသုံးချလို့ ရရှိပါမှာကိုပဲ။

**Note:** textfield Class ဟာ မိမိကိုယ်တိုင် ဖန်တီးခဲ့တဲ့ class ဆိုတာကို သတိပြုမိဖော်ပါတယ်။ ဒါပေမယ JPanel Class ကို မို့မြင်းပြီးတော့ extendsလုပ်ထားပါတယ်။

၆။ လိုင်းနံပါတ် (27) ကတော့ JPanel ကို JFrame method ထဲကို ပေါင်းထည့်နိုအတွက် add () method ကို ခေါ်ယူ အသုံးချထားပါတယ်။ ဒါကြောင့် -

```
f.getContentPane().add(jtf);
```

လို့ ခေါ်ယူထားပါတယ်။ အဲဒီ JFrame ထဲမှာမူ add (jtf) ဆိုတာက textfield တွေကို ဖော်ပြုနိုအတွက် textfield Class ရဲ့ Object\_Name ကို ခေါ်ယူထားတာပါ။ လိုင်းနံပါတ် (29) က

```
f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
```

ဆိုတာကတော့ JFrame Class ကို ဝိတ်ပစ်နိုပါပဲ။ အဲဒီ နောက်မှာတော့ f.show () method နဲ့ Output windows ကို ခေါ်ယူ လိုက်တာပါပဲ။ အပေါ်က code\_line တွေကို ဘယ်လိုရေးခဲ့ရေးခဲ့ အခုရေးပြုခဲ့တဲ့ f.show ( ) method သာ ကျွန်ုတ်ရင် program ဟာ compileလုပ်နိုင်ပေမယ့် Run လုပ်တဲ့အခါမှာ ဘာမျှ လာပြပေးနိုင်မှာ မဟုတ်ပါဘူး။ ဒါကြောင့် ဒီအချက်ကို အထူးသတိပြုမိဖော်ပါတယ်။ လိုင်းနံပါတ် (31) က main () function ရဲ့ close brace } ပါပဲ။ လိုင်းနံပါတ် (32) ကတော့ primary class ဖြစ်တဲ့ textfield Class ရဲ့ close brace } ပါပဲ။ ဒါဆိုရင် အခါ

Program ကို Compile & Runလုပ်လိုက်ရင် Figure 3.8.4 မှာ ပြင်ရတဲ့ Output Windows ကို မြင်ရမှာပါဝဲ။

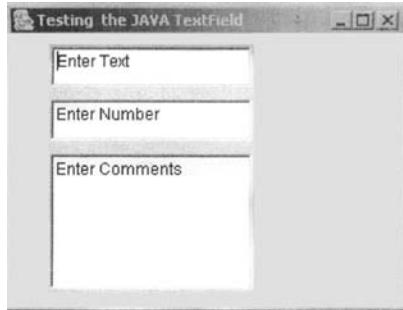


Figure 3.8.4 Run & Show of the Output Windows, textfield .java Program

က ဆက်လက်ပြီးတော့ JAVA ရဲ့ swing components တွေ အကြောင်းကို လက်ရှုသင်ခန်းစာ chapter [3] ရဲ့ အောက် မှာပဲ sub-Index တွေ ခွဲပြီးတော့ ပို့ချပေးထားပါတယ်။ အမှန် တကယ်ကတော့ JAVA Swing Components တွေကို အသုံးပြုရတာဟာ ပြီးခဲ့တဲ့ Chapter [1] and [2] က သင်ခန်းစာတွေ ဖြစ်တဲ့ program code\_line တွေ structure ဖွဲ့စည်းထဲတွေ Class, Interface အစရိတ် အခြေခံအယူအဆတွေလောက် လေးလာရတာ မရှုတေားပါဘူး။ အခြေခံအယူအဆတွေကို ပိုင်နိုင်မှသာ လျှင် JAVA Swing Components တွေနဲ့ တွေ့ဖော်ပြီးတော့ လိုချင်တဲ့ program တွေကို ဖန်တီးနိုင်မှာပါ။ ဒါကြောင့် ပြုပြန်လေ့လာစေခဲ့တယ်။ က JAVA Swing Components တွေကို ဆက်လက်လေ့လာကြပါစို့။

KYAW ZAYAR LAY [ IT ENGINEERING ]

## သင့် အိတ်ကာင်တဲ့က သို့ပေါ်ပွဲသွေးထွေ

အိတ်ကာင်ထဲထည့်ပြီး သွားလေရာသို့ ဆောင်သွားနိုင်တဲ့ hard drive ဆုံးကလေးကို Western Digital က ထုတ်လုပ်ခဲ့ပါတယ်။ 'Passport Pocket USB 2.0' လို့ အမည်ပေးထားတဲ့ ငါးဦး hard drive အရွယ်က ဆေးလိပ်ဘူး သာသာပါ။ ငါးဦးသည် လက်ရှိ ခေတ်စားနေတဲ့ flash memory အမျိုးအစား မဟုတ်ဘဲ hard drive ဖြစ်တယ်ဆိုတာကိုတော့ သတိပြုစေလိုပါတယ်။ Western Digital ဆိုတာကလည်း ဟု့ဒု့ဒ်ဒေတွေထဲတော်လုပ်လာခဲ့တာ နှစ်ပေါင်း မနည်းတော့တဲ့ ကုမ္ပဏီပါ။ ဟု့ဒု့ဒ်များကို အချက်အစား သေးငယ်သည်ထက် သေးငယ်အောင် ပြုလုပ်နေသလို အချက်အလက် သို့လောင်နှင့် မြင့်မားအောင် လည်း ကြိုးစားနေတဲ့ ကုမ္ပဏီလည်းဖြစ်ပါတယ်။ ယခု အချက်အလက်သို့လောင် နှင့် 6 gigabytes ရှိတဲ့ 'Passport Pocket USB 2.0' ကို ဒေါ်လာ ၁၀၀ နှစ်းနဲ့ ရောင်းချုပါတယ်။ ငါးဦးဘွင်း 'WD Sync' ဆော့ပဲထည့်သွင်းထားပြီး ဂွန်ပူးတာထဲက ဖိုင်များကို စကားဂုဏ်ပုံစံမျိုးနဲ့ တစ်ခါတည်း သိမ်းဆည်းပေးနိုင်ပါတယ်။





### 3.2.2 USING THE JAVA SWING COMPONENTS AND AWT COMPONENTS

#### ◆ Label <OR> JLabel Box

Listing 3.3 Testing the JAVA Label Box // save file name is label.java

```

1. import javax.swing.*;
2. import java.awt.*;
3. public class label extends JPanel
4. {
5.     public label()
6.     {
7.         setLayout(null);
8.         Label L1 = new Label("label 1"); // create label 1
9.         Label L2 = new Label("label 2"); // create label 2
10.        Label L3 = new Label("label 3"); // create label 3
11.        L1.setBounds(30,20,70,20); // int x, int y, int width, int height
12.        L2.setBounds(30,50,70,20);
13.        L3.setBounds(30,80,70,20);
14.
15.        add(L1);
16.        add(L2);
17.        add(L3);
18.
19.    public static void main(String args[])
20.
21.    // to set the title bar text
22.    JFrame f = new JFrame("Testing the JAVA Label");
23.    Label Lbl = new Label();
24.    f.setSize(300,200); // width and Height of Frame
25.    f.getContentPane().add(LBL);
26.
27.    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
28.    f.show();
29.
30. }
```

ကဲ အခု ဒီ program မှာကတော့ Label Box ကို ဘယ်လို အသုံးပြုထားသလဲဆိုတာကို ရှင်းလင်းပြပါမယ်။ ကျေန်တော်

အပေါ်မှာရေးပြုခဲ့တဲ့ code\_line ထွေကို Figure 3.9.1 မှာ ပြထားတဲ့အတိုင်း NotePad Program ထဲကိုသွေးပြီးတော့ ရှိက်သွင်းလိုက်ပါပြီး။ Figure 3.9.1 ကို လေ့လာကြည့်ပါ။



Figure 3.9.1 To show the Program Code\_Line in NotePad

ပြီးရင်တော့ ထဲ့စံအတိုင်း C:\j2sdk 1.4.1-01\Bin > ထဲကို သွေးပြီးတော့ saveလုပ်ပါ။ Saveလုပ်ရမယ့် File\_Name ကတော့ Label.java လို့ ပေးလိုက်ပါ။

ကဲ ပြီးတော့ compile လုပ်တာနဲ့ Run လုပ်တာကိုတော့ အကျယ်တောင့် ရှင်းမပြတော့ပါဘူး။ ရှုံးသင်ခန်းစာတွေမှာ ပို့ချွေသားမို့ Figure 3.9.2 ကိုသာ လေ့လာလိုက်ပါ။

```

C:\ Command Prompt
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Admin>cd c:\j2sdk1.4.1_01\bin

C:\j2sdk1.4.1_01\bin>javac label.java

C:\j2sdk1.4.1_01\bin>java label 1
Label 1
  
```

Figure 3.9.2 Compile & Run of the Label.java Programs

ကဲ Compile & Run လုပ်လိုက်ရင်တော့ Label Box <3> ခုကို use လုပ်ထားတဲ့ Output Windows ကို စာဖတ်သူဟာ အခါလို တွေ့မြင်ရမှာပါ။ Title Barမှာလည်း Testing the JAVA label Box လို့ ပါတယ်နော်။

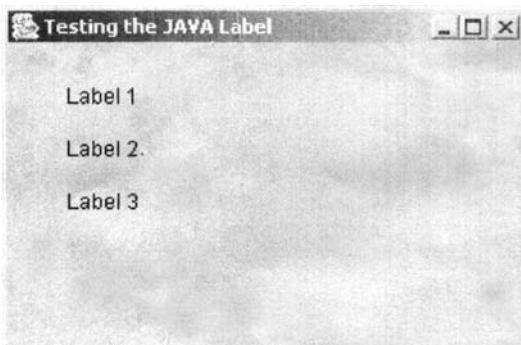


Figure 3.9.3 Run & Show of the Output Windows, Label. java Program

ကဲ code\_line တွေရေးပဲ့၊ Compile & Run လုပ်ပဲ့တွေကို လေ့လာပြီးရင်တော့ code\_line တွေကို Analysis လုပ်ကြတာ ပေါ့။ Analysis လုပ်ရမှာ အရှေ့သင်ခန်းစာ 3.2 က Testing the JAVA Textfield Program နဲ့ ဆင်တူတာလို့ လေ့လာရတာ လွယ်ကူသလိုဖြစ်ခဲ့တဲ့ သင်ခန်းစာကို ပြန်လေ့လာရာလည်း ရောက်ပါတယ်။ ဒါကြောင့် လေ့လာကြည့်လိုက်ပါဦး။

#### Analysis

၁။ java program တွေရဲ့ ထုံးစံအတိုင်းပေါ့။ Package file တွေကို Declare လုပ်ပါတယ်။ အဲဒေါက်ပိုင်း လိုင်းနံပါတ် (၃) မှာမူ primary class ဖြစ်တဲ့ label ဆိုတဲ့ class ကို ဖန်တီးပါတယ်။ ဖန်တီးရာမှာ JPanel Class ကို extends လုပ်ထားတာနော်။ ကဲ အဲဒီ Class ရဲ့ constructor method ( ) ကို လိုင်းနံပါတ် (၅) မှာ ရေးသားပါတယ်။ အဓိကအချက်က ဒီ constructor method ဖုံးနဲ့ ရေးသားထားတဲ့ label () function ပဲဆိုတော့ အဲဒီ function () ကိုပဲ အဓိက လေ့လာကြတာပေါ့နော်။

**Note :** Constructor method ( ) ဆိုတာကတော့ Class\_Name နဲ့ function \_Name တူမေ့တာကို ပြောတာနော်။ ဒါကတော့ ခဏခဏ ရှင်းပြနေလို့ နားလည်သော ပေါက်နေလောက်ပါပြီ။

၂။ ကဲ constructor method ( ) ထဲက set layout (null) method ကတော့ setBounds ( ) method ကို ခေါ်ယူ အသုံးပြုချင်လို့ (null) arguments ထည့်ပြီးတော့ အသုံး ချထားပါတယ်။ အဲဒီ null arguments သာ မပါရင်တော့ setBounds ( ) method ကို အသုံးချလို့ ရနိုင်မှာ မဟုတ်ပါဘူး။ လိုင်းနံပါတ် (7),(8),(9) မှာကတော့ Label Box (3) ခုကို create လုပ် ဖန်တီးပါတယ်။ အဲဒါဟာ constructor method ( ) ထဲမှာတင် ဖန်တီးနေတာနော်။ လိုင်းနံပါတ် (11),(12),(13) မှာကတော့ label Box တစ်ခုချင်းစို့ တည်နေရာတို့ X,Y ဝင်ရှိနဲ့ ညီပြီးတော့ ခေါ်ယူပါတယ်။ ဒါကြောင့်

L1 .Set Bounds (30,(**20**), 70, 20); // int x, int y, int width, int height

L2. Set Bounds (30,(**50**), 70, 20);

L3. Set Bounds (30,(**80**), 70, 20); ဆိုပြီး ခေါ်ယူထားတာပါ။

y ဝင်ရှိုး တည်နေရာတွေ  
ကွဲပြားခြားနားတာပါ။

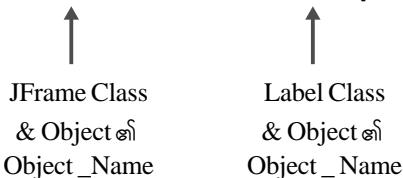
၃။ လိုင်းနံပါတ် (15),(16),(17) မှာတော့ label Box တွေကို ခေါ်ယူထားတာပါ။ ဒါကြောင့် add () method အနေနဲ့-  
add (L1);  
add (L2);  
add (L3);  
လို့ ရေးသားထားတာပေါ့။ ကဲ လိုင်းနံပါတ် (18) ကတော့ label () ဆိုတဲ့ constructor function ရဲ့ close brace } ပါပဲ။ ဒီအထိ စာဖတ်သူအနေနဲ့ သေသေချုချု ဖတ်ရှုကြည့်ပြီးတော့ လေ့လာကြည့်ပါ။ လက်နဲ့ Program code\_line တွေကို လက်နဲ့ ချေရေးပြီးတော့ လေ့လာကြည့်ပါ။

၄။ လိုင်းနံပါတ် (19) မှာ ရေးသားထားတဲ့ main () function ကို စတင်လေ့လာကြရအောင်လား။ အဲဒီ main () function ထဲမှာ JFrame class ကို Object \_Name f လို့ အမည်ပေးပြီးတော့ object အဖြစ် ဖန်တီးခေါ်ယူထားပါတယ်။ အဲဒီ JFrame class ရဲ့ constructor function မှာပါတဲ့ String arguments နေရာမှာ "Testing the JAVA label" ဆိုတဲ့ ပါဝင်စေချင်တဲ့ စာသားကို ရေးသားထားပါတယ်။ အဲဒါကြောင့်

## ကွန်ပူးတာသင်ခန်းစာ

JFrame f = new JFrame ("Testing the JAVA Label");  
 လို့ ရေးသားထားတာပါ။ ဒါကြောင့် windows JFrame မှာ ပါဝင်ရမယ့် Text စာသားကို ရေးသားထားပဲ ရှင်းပြီနော်။  
 အဲဒီ နောက် လိုင်းနံပါတ် (23) မှာတော့ ကျွန်ုပ်တော်တို့အပေါ်  
 မှာ ဖန်တီးခဲ့တဲ့ label class ကို Object\_Name အနေနဲ့  
 Lbl လို့ အမည်ပေးပြီးတော့ ခေါ်ယူလိုက်ပါတယ်။ အဲဒီ  
 နောက် လိုင်းနံပါတ် (24) မှာတော့ JFrame ရဲ့ size ကို  
 (300,200) လို့ widthနဲ့ height အဆုံးအစားကို တွက်ချက်  
 ထည့်သွင်းပါတယ်။ ဒါကို နားလည်ပြီးလောက်ပါပြီ။ အဲဒီ  
 နောက် လိုင်းနံပါတ် (25) မှာတော့ get contentPane  
 () method နဲ့ add () method ကို တွဲဖော်ပြီးတော့ အသုံး  
 ချဖိတယ်။ ဒါကြောင့်

f.get contentPane () .add (lbl); လို့ ခေါ်ယူထားတာပါ။



ပြီးတော့ JFrame Windows ကို Class & Exit လုပ်တဲ့  
 အခါမှာ Windows Frame ပိတ်သွားအောင်လို့ လိုင်းနံပါတ်  
 (27) မှာ-

f.set Default Close Operation (JFrame.EXIT\_  
 ON\_CLOSE);

လို့ ရေးသား ခေါ်ယူထားတာပါ။

အဲဒီနောက် Program Run လုပ်ရင် Output လာပြနိုင်  
 အောင်လို့-

f. show ();

ဆိုပြီးတော့ show () method ကို ခေါ်ယူ အသုံးချထားပါ

တယ်။ show () method ဟာ JFrame Class ထဲမှာ  
 ပါဝင်တာပါ။ ဒီတော့ Class ကို object သို့ ပြောင်းလဲတဲ့  
 လိုက်စားလှယ် varibale\_Name f ကိုရှုက .(dot) ခံပြီး  
 တော့ ခေါ်ယူရတာပါ။

၅။ လိုင်းနံပါတ် (29) ကတော့ main () function ရဲ့ close  
 brace } ပါပဲ။ အဲဒီအောက်ကတော့ Primary Class label  
 ရဲ့ close brace } ပဲ ဖြစ်ပါတယ်။ ဒါဆိုရင်တော့ စာဖတ်သူ  
 ဟာ ဒီ program code\_line အလုပ်လုပ်သွားပဲကို နားလည်  
 သဘောပေါက်သွားမှာပါ။

အဲဒီ program ကို label.java လို့ အမည်ပေးပြီး save  
 လုပ် Compile & Run လုပ်လိုက်ရင် အောက်မှာ ဖော်ပြထားတဲ့  
 Figure 3.9.4 ၏ Output windows ကို တွေ့မြင်ရမှာပါပဲ။

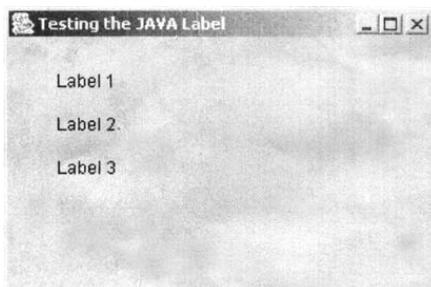


Figure 3.9.4 Run & Show of the Output Windows,  
 Label .java Program

ကဲ ဆက်လက်ပြီးတော့ Listing 3.2 နဲ့ Listing 3.4 ထိုကို  
 ပေါင်းစပ် ရေးသားပြီးတော့.textLabel.java ဆိုတဲ့ program  
 ကို ဖန်တီး ရေးသားထားပါတယ်။ ဒီ program မှာ textfield နဲ့  
 Label Box တွေကို ပေါင်းစပ်အသုံးချေပေးထားပါတယ်။ Listing  
 3.3. အနေနဲ့ပေါ့။ စာဖတ်သူအနေနဲ့ လေ့လာလိုက်ပါဉိုး။

KYAW ZAYAR LAY [ IT ENGINEERING ]

ကျေးဇူး ကိုပို့အားကိုယ်ကိုးစာကြည့်တိုင်းလှုပ်ပြီး ကျေးဇူးပေါင်း (၅၄၇၀၇) အတွက် စာအုပ်၊ စာတော်များနှင့်  
 ငွေသားအလူရှင် ငွေလာက်ခံရနိုက်စာရင်း

စဉ်	လူပါန်းဖွံ့ဖြိုးလုပ်သည့်အကြောင်း	အလူရှင်းရေ	ငွေသားလူပါန်းရရှိနှု	စာအုပ်၊ စာတော်နှင့်တန်ဖိုး	
				အပ်ရေ	တန်ဖိုး (ကျပ်)
(၁)	ပထားမှ သတ္တမအကြိမ်အထိ အငွေမအကြိမ်ရရှိမှု	၄၉၀	၄၈၇၄၀၀၀	၂၆၇၇၄၉၉	၈၆၆၄၄၈၂
(၂)	နုဝင်းမှု (ပုဂ္ဂိုလ်မှု) (၂၆-၃-၂၀၀၆)	၁၀၀	၄၆၇၀၀၀၀	၉၁၁၈၅	၂၆၈၆၇၅၀
(၃)	ဒေသမ (ရန်ကုန်) (၁၀-၄-၂၀၀၆)	၁၇	၁၀၀၀၀၀၀	၁၇၇၈၈	၂၉၀၉၄၀၀
(၄)	မန္တလေး (သာကြိုး) (၂၄-၄-၂၀၀၆)	၁၇၅	၇၇၀၀၀၀၀	၁၉၇၃၇	၂၅၅၈၅၈၀
	ပေါင်း	၇၇၅	၆၃၇၀၉၀၀	၁၉၇၀၇၂၇	၂၇၅၈၇၉၉၂



# Java Developer Guide

## 3 .2 .3 Using The JAVA Swing Components And AWT Components

### Combine Label Box and TextField (OR) JLabel and JTextField

အရှေ့သင်ခန်းစာမှာ ရေးသား ပိုချုခဲ့တဲ့ Label Box နဲ့ JTextField တွေကို ပေါင်းစပ်ပြီးတော့ အသုံးချ ကြည့်ကြအောင်လား။ ဒီတော့ listing 3.4 အနေနဲ့ ရေးပြထားတဲ့ program ကို လေးလာကြည့်ကြအောင်လား။

Listing 3.4 Combine Use of Label Box and TextField

```

1. import javax.swing.*;
2. import java.awt.*;
3. public class textlabel extends JPanel
4. {
5.     public textlabel()
6.     {
7.         SetLayout (null);
8.
9.         TextField tf1 = new TextField ("Enter Text",50);
10.        TextField tf2 =new TextField ("Enter Number",20);
11.        TextField tf3 = new TextField ("Enter comments",200);
12.        tf1 .SetBounds (80,10,150,30);
13.        tf2 .SetBounds (80,50,150,30);
14.        tf3 .SetBounds (80,90,150,100);
15.        add (tf 1);
16.        add (tf 2);
17.        add (tf 3);
18.
19.        Label L1= new label ("label 1");
20.        Label L2 = new label ("label 2");

```

```

21. Label L3 = new label ("label 3");
22. L1 .SetBounds (30,10,70,20);
23. L2 . SetBounds (30,50,70,20);
24. L3. Set Bounds (30,90,70,20);
25. add (L1);
26. add (L2);
27. add (L3);
28. }
29. public static void main (string args [ ])
30. {
31.
32. JFrame f = new JFrame ("Testing the JAVA TextField
and label");
33. text label tl = new textlabel ();
34. f.setSize (350,250); // width and Height of Frame
Windows
35. f.get ContentPane () . add (tl) ;
36. f.set Default Close Operation (JFrame .EXIT_ON_
CLOSE);
37. f.show () ;
38. }
39. }
```

ကဲ အခုရေးပြထားတဲ့ program code\_line တွေကို Notepadမှာသွားပြီးတော့ ရိုက်သွင်းရပါမယ်။ ပြီးရင်တော့ အဲဒီ File ကို textlabel.java လို့ အမည်ပေးရပါမယ်။ အဲဒီလို့ NotePad မှာ ရိုက်သွင်းထားပဲကို Figure 3.10.1 မှာ ပြပေးထားပါတယ်။

```

textlabel - Notepad
File Edit Format View Help
import javax.swing.*;
import java.awt.*;
public class text_label extends JPanel
{
    public textlabel()
    {
        setLayout(null);
        JTextField tf1=new JTextField("Enter Text",50);
        JTextField tf2=new JTextField("Enter Number",20);
        JTextField tf3=new JTextField("Enter Comments",200);
        tf1.setBounds(80,10,150,30);
        tf2.setBounds(80,50,150,30);
        tf3.setBounds(80,90,150,100);
        add(tf1);
        add(tf2);
        add(tf3);
        Label l1=new Label("Label 1");
        Label l2=new Label("Label 2");
        Label l3=new Label("Label 3");
        l1.setBounds(30,10,70,20);
        l2.setBounds(30,50,70,20);
        l3.setBounds(30,90,70,20);
        add(l1);
        add(l2);
        add(l3);
    }
    public static void main(String args[])
    {
        JFrame f=new JFrame("Testing the JAVA TextField and Label");
        textlabel t1=new textlabel();
        f.setSize(350,250);
        f.getContentPane().add(t1);
        f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        f.show();
    }
}

```

Figure 3.10.1 To show the Program code\_line in NotePad

ကဲ Program code\_line တွေ ရေးသွင်းပြီးတဲ့ file ကို C:\j2sdk 1.4.1\_01\Bin ထဲကို ဝင်ရောက်ပြီးတော့ saveလုပ်ပါ။ Save လုပ်ရမယ့် File\_Name က textlabel.java နော်။ မမေ့ပါနဲ့။

ကဲ Program ကို save လုပ်ပြီးရင် Compile လုပ်ကြရအောင်။ Compile လုပ်ဖို့အတွက် DOS <OR> command prompt ထဲကိုဝင်ပါ။ Save လုပ်ထားတဲ့ Directory ရှိရာကို သွားဖို့

CD C:\j2sdk 1.4.1\_01\Bin ↵

ဆိတဲ့ command line ကို ရှိက်သွေးပြီးတော့ Directory ကို ပြောင်းပါ။ ပြီးရင်တော့ အောက်ပါ command နဲ့ program ကို compileလုပ်ပါ။

javac textlabel.java ↵

compile လုပ်ပြီးရင်တော့ –

java textlabel ↵

ဆိပြီး Run လုပ်ပါ။ အဲဒီလိုမျိုး Compile & Run လုပ်ပုံကို Figure 3.9.2 အနေနဲ့ ပြပေးထားပါတယ်။

Note: ကွန်တော်ရှင်းပြချင်တာကတော့ Compile & Run လုပ်ရမယ့် အဓိပ္ပာတဲ့ ကွန်တော်ပြောသလို တန်းပြီး Output က ထွက်လာမှာ မဟုတ်ဘူးနော်။ စာဖတ်သူရဲ့ Program code\_line မှာ case sensitive ဖြစ်တဲ့ အတွက် Error တွေ တက်လာမှာပါ။ Compile မပြောတဲ့ error တွေ၊ ဥပမာ- ဘယ်နေရမှာ brace } ကျွန်တယ်။

ဘယ်နေရာ စာလုံးအကြီး၊ အသေးမတူတာ။ ;semi-colon ကျွန်တာ အစရှိတဲ့ Error တွေကို ရှင်းလိုက်၊ ပြန်ပြင်လိုက်၊ Save As ပြန်လည်ပြီးတော့ Dos <OR> Command line Interface ထဲမှာပြန်ပြီး Compile ပြန်လည်ရမှာပါ။ အဲဒီလိုမျိုး Error အမျိုးမျိုးနဲ့ ရင်ဆိုင် တွေပြီးကာမှ ပြည့်စုံ မှန်ကန်တဲ့ Output Windows ကို ပြင်တွေရမှာပါ။ ဒီအဆင့်တွေကိုတော့ စာဖတ်သူရဲ့ ဘက်ကပ် ကြိုးစားအားထုတ် လေ့လာစေချင်ပါတယ်။

ကဲ Compile & Run လုပ်ပုံကို Figure 3.10.2 အနေနဲ့ ပြထားပါတယ်။ လေ့လာကြည့်ပါ။

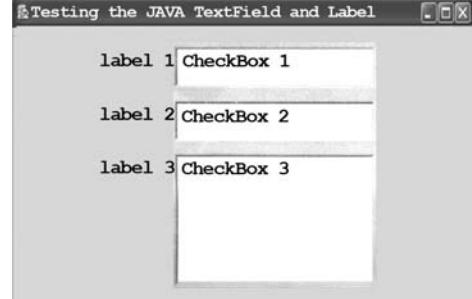
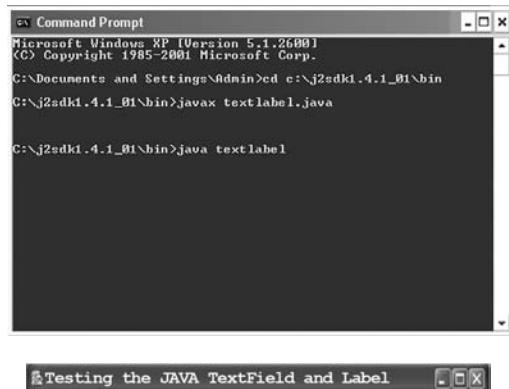


Figure 3.10.2 Compile, Run & Show of the Output Windows, textlabel.java Program

အခါ ကွန်တော် ရှင်းလင်းပြတဲ့ အကြောင်းအရာတွေကို သေချာချာ နားလည်အောင် ဂရိုစိုက်ပါ။ ဘယ် Programming Language မှာမဆို error ဆိတဲ့ အမြှုနေရမှာပါ။ အဲဒီ error တွေကို ရှင်းလင်းရင်းနဲ့ ထက်မြေက်တဲ့ အတွေအကြုံတွေကို ရရှိလာ မှာပါပဲ။

ကဲ ဆက်လက်ပြီးတော့ textfield.java program ကို Trace လိုက်ပြီးတော့ Analysis လုပ်ကြည့်ကြရအောင်လား။ TextField and Label Box နှစ်ခုကို ယုံ့ချုံတဲ့ အသုံးပြန်လည်းကို နားလည် သဘောပေါက်သွားပါလိမ့်မယ်။

### Analysis

၁။ ဒါ program ကို ရှင်းပြုမယ်ဆိုရင် ရဲ့သင်ခန်းစာ listing 3.1 နဲ့ listing 3.2 ထို့ထို့ ကျေည်က် နားလည့်ခဲ့မယ်ဆိုရင် တော့ ဘာနဲ့ သိပ်ရှင်းပြုစရေတောင် မရှိပါဘူး။ ထုံးစံအတိုင်း java package တွေကို import လုပ် ခေါ်ယူပါတယ်။ ပြီးတော့ JPanel Class ကို extends လုပ်ပြီးတော့ text-label ဆိုတဲ့ Primary Class ကို ဖန်တီးပါတယ်။ အဲဒါ textlabel class ရဲ့ constructor function () ကို လိုင်းနံပါတ် (၅) မှာ ရေးသားထားတာကို တွေ့ဖြင့်ရမှာပေါ့။ Label Box နဲ့ TextField တွေကို ထပ်ဆင် ခေါ်ယူ အသုံးပြု ဖို့အတွက် setLayout (null); method ကို လိုင်းနံပါတ် (၇) မှာ ရေးသားထားပါတယ်။ က ပြီးတော့ လိုင်းနံပါတ် (၉), (၁၀), (၁၁) မှာကတော့ TextField (၃) ခုကို ဖန်တီးရှိ အတွက် class ကို object အဖြစ် ဖန်တီးပါတယ်။ TextField Class ကို object အဖြစ် ဖန်တီးတဲ့အခါမှာ tf1, tf2, tf3 အစရိတဲ့ Object\_Name တွေပေးပါတယ်။ ပြီးတော့ TextField (၃) ခုကို SetBounds () method နဲ့ ထည့်သွင်း တပ်ဆင်ပါတယ်။ လိုင်းနံပါတ် (၁၅), (၁၆), (၁၇) မှာ တော့ add () method နဲ့ JFrame Windows ပေါက်ဖြည့်သွင်း တပ်ဆင်ထားလိုက်ပါတယ်။ ဒါကြောင့်-  
add (tf1);  
add (tf2);  
add (tf3);  
ဆိုပြီး ရေးသားထားတာပေါ့နော်။ အခု ရေးပြုခဲ့တဲ့ add () method ဟာ JFrame Class ရဲ့ JFrame Windows ပေါက်ဖြည့်သွင်းပေးပါတယ်။  
၂။ လိုင်းနံပါတ် (၁၉),(၂၀),(၂၁) မှာတော့ JPanel Class ထဲက extends လုပ်ထားတဲ့ label Class ကို လှမ်းပြီးတော့ ခေါ်ယူပြီးတော့ object အဖြစ် ဖန်တီးပါတယ်။ Object\_Name ကို  
Label L1= new label ("label 1");  
Label L2 = new label ("label 2 ");  
Label L3 = new label ("label 3");  
လို့ object (၃) ခု ဖန်တီးပါတယ်။ ဒါဟာ အရှေ့က Listing 3.2 တွေနဲ့ကလိုပါပဲ။ အလွန်ရှိရှင်းပါတယ်။  
လိုင်းနံပါတ် (၂၂), (၂၃), (၂၄) မှာတော့ Object\_Name .dot ခံပြီးတော့ SetBounds () method ကို အသုံးပြုပါတယ်။ ဒါကြောင့်မို့-  
L1.setBounds (30,10,70,20);  
L2.setBounds (30,50,70,20);  
L3.setBounds (30,90,70,20);

လို့ ရေးသားထားတာပေါ့နော်။ အဲဒီနောက် JFrame Windows ပေါက်မှာ တပ်ဆင်နိုင်ပါ / ပါဝင်နိုင်ပါ add () method ရဲ့ parameter arguments ထဲမှာ ထည့်သွင်းပြီးတော့ အသုံးပြုပါတယ်။ ဒါဟာ လိုင်းနံပါတ် (25),(26), (27) မှာ ရေးသားထားတာပါ။  
အဲဒီနောက် textlabel ( ) constructor function ရဲ့ close brace } ကို လိုင်းနံပါတ် (28) မှာ လာပိတ်ထားပါတယ်။  
၃။ main () function ကိုတော့ လိုင်းနံပါတ် (29) မှာ စတင်ဖန်တီး ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (32) မှာမှာ JFrame Class ကို object အဖြစ် ဖန်တီးပြီးတော့ ခေါ်ယူထားပါတယ်။ Title Bar မှာ ပါဝင်ရမယ့် စားသားအတွက်ရေး Testing the JAVA TextField and Label လို့ ရေးသားထားပါတယ်။  
အဲဒီနောက် အပေါ်မှာပါဝင်ပြီး ရေးသားခဲ့တဲ့ constructor function ဖြစ်တဲ့ textlabel ကို လုပ်ပြီးတော့ ခေါ်ယူပါတယ်။ Object\_Name အဖြစ် t1 လို့ အမည်ပေးထားပါတယ်။ လိုင်းနံပါတ် (34) မှာတော့ JFrame Windows ရဲ့ size ကို- f.setSize (350,250);

ဆိုပြီးတော့ ရေးသားဖန်တီးပါတယ်။ လိုင်းနံပါတ် (35),(36),(37) မှာ ရေးသားထားတဲ့ Code\_Line တွေကတော့ အထွေအထူး ရှင်းပြုရာ မလိုပါဘူး။

**Listing 3.1 & 3.2**မှာ ရှင်းပြုခဲ့ပြီးသားမို့ နားလည် သဘောပါက်ပြီးနောက်ပါ။

ဒါဟာ ပြီးခဲ့တဲ့ TextField နဲ့ Label Box နဲ့ ပတ်သက်တဲ့ program နှစ်ခုကို ပေါင်းစပ် ရေးသားပြီးတော့ အသုံးချကြည့်တဲ့ program ပဲဖြစ်ပါတယ်။ ဒါတော့ ဒီ textlabel.java Program ကို လေ့လာကြည့်ရင် Swing Components နဲ့ awt components တွေကို ခေါ်ယူအသုံးချက် မပါမဖြစ် ပါဝင်ရမယ့် method () တွေ လုပ်ဆင်ပါတယ်။ လို့ ထည့်သွင်းရမယ့် arguments parameter တွေအကြောင်းကို အခြေခံကျကျနားလည် သဘောပါက်သွားမှာပါ။

### 3.2.4 Using the JAVA Swing Components & AWT components

#### JCheckbox (OR) Checkbox

ကျွေးနဲ့တော်တို့ အရှေ့မှာ ဖန်တီးခဲ့တဲ့ Demo Program ကလေးတွေဟာ အဓိကပင်မ program တွေ ဖန်တီး တည်ဆောက်ရာမှာ အလွန်အရေးပါတဲ့ Tools လေးတွေကို အသုံးချ ဖန်တီးထားတာပါ။ အခု ဖန်တီးလေ့လာမယ့် JCheck Box နဲ့ ပတ်

## ကွန်ပါတာသင်ခန်းစာ

သက်ပြီးတော့ သက်ဆိုင်တဲ့ အယားများကို Table 3.5 CheckBox and CheckBox Group constructor ဆိုပြီးတော့ ရေးသား ပို့ချခြုံသားပါ။ အဲဒီ Table 3.5 ကို လေ့လာကြည့်ရင် လက်ရှိ အသုံးဖြစ်ပါ။ JCheckBox နဲ့ ပတ်သက်ပြီးတော့ အကြမ်းဖျင်း အားဖြင့်တော့ သဘောပါက်ထားမှာပါ။ ဒါကြောင့် listing 3.6 အနေဖြင့် ရေးသားပို့ချပေးခဲ့တဲ့ program ကို လေ့လာကြည့်ပါမယ်။

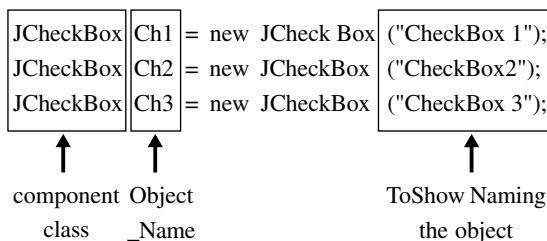
**Listing 3.6 Using the JCheckBox (OR) CheckBox**

```
1. import javax.swing.* ;
2. import java.awt.*;
3. public class checkbox extends JPanel
4. {
5.     public Checkbox () // Checkbox Class constructor
function
6.     {
7.         setLayout (null);
8.         JCheckBox Ch1 = new JCheckBox
("CheckBox1");
9.         JCheckBox Ch2 = new JCheckBox ("CheckBox
2");
10.        JCheckBox Ch3 = new JCheckBox ("Check-Box3");
11.
12.        Ch1.setBounds (60,30,100,30);
13.        Ch2.setBounds (60,60,100,30);
14.        Ch3.setBounds (60,90,100,30);
15.
16.        add (Ch1);
17.        add (Ch2) ;
18.        add (Ch3) ;
19.    }
20.    public static void main (String args [ ] ) {
21.        JFrame f = new JFrame ("Testing the JAVA
CheckBox" );
22.        Checkbox chb = new Checkbox ( ) ;
23.        f.setSize (300,200);
24.        f.getContentPane ( ) .add (chb);
25.        f.setDefaultCloseOperation (JFrame.
EXIT_ON_CLOSE);
26.        f.show ( ) ;
27.    } // Close brace} for main ( ) function
28. } // Close brace } for Checkbox Class
```

### Analysis

၁။ ဒီ program သိလေးမာတော့ သိပ်ပြီး အထွေအထူး ရှင်းပြ စရာ မလိပါဘူး။ လိုင်းနံပါတ် (၁) နဲ့ (၂) မှာ SWING & AWT component တွေကို အသုံးချဖို့အတွက် package file တွေကို import လုပ်၊ Declare လုပ် သော်ပါတယ်။ ပြီးတော့ CheckBox ဆိုတဲ့ primary class ကို ဖန်တီး ပါတယ်။ ဖန်တီးရာမှာ JPanel Class ကို extends

လုပ်ထားတာနော်။ လိုင်းနံပါတ် (၅) မှာတော့ Checkbox Class ရဲ့ constructor function( )ကို ရေးသားထားပါတယ်။ Brace { တွေကိုတော့ သတိပြုပါ။ ဒြီးတော့ လိုင်းနံပါတ် (၇) မှာ setLayout() methodကို ခေါ်ယူ အသုံးချထားပါတယ်။ ဒြီးတော့ Swing component ဖြစ်တဲ့ JCheckBox ကို ခေါ်ယူအသုံးချမှိုင်းဖို့အတွက် လိုင်းနံပါတ် (၈)၊ (၉)၊ (၁၀) မှာ အောက်ပါအတိုင်း ရေးသားထားတာပါပဲ။



ဒါဆိုရင်တော့ CheckBox (3) ခုကို ကြော်တပ်ဆင်ထားပဲ  
ကို စာဖတ်သူ နားလည်သော့ပေါက်သွားမှာပါ။ ကဲ ဒီတော့  
checkbox သုံးခုကို Declare လုပ်ပြီးသွားပြီခိုရင်တော့ လိုင်း  
နံပါတ် (၁၂)၊ (၁၃)၊ (၁၄) မှာ SetBounds () method နဲ့  
ကြော် အသုံးချဖြိုး Form-Windows ပေါ်မှာ ထည့်သွင်း တင်  
ဆင်ပါတယ်။ Object\_Name () .dot အနေနဲ့ ကြော်ရာမှာ  
အပေါ်မှာ Declare & Name လုပ်ခဲ့တဲ့ Object\_Name တွေကို  
ယူယူစွဲပြီး သက်ဆိုင်ရာအလိုက် အသုံးချပါတယ်။

Ch1. SetBounds (60,30,100,30);

## Ch2. SetBounds (60,60,100,30);

### Ch3. SetBounds (60,90,100,30);

က ဒီအထိ နားလည် သဘောပေါက်ပြီဖော်။ ဒါဆုံးရှင်တော့  
 add () method ကို ခေါ်ယူ အသုံးချထားတဲ့ လိုင်းနံပါတ် (၁၆)  
 (၁၇) (၁၈) က code\_line တွေကို စာဖတ်သူ အလှယ်တက္ကာ  
 နားလည် သဘောပေါက်သွားမှာပါ။ ဂျာနှစ်အနေနဲ့ကတော့  
 အကျယ်ချုပြုးတော့ မရင်းပြတော့ပါဘူး။ လိုင်းနံပါတ် (၁၉) က  
 တော့ constructor.function Checkbox ()၊ close brace }  
 ပါပဲ။ ဒီအထိ နားလည်သဘောပေါက်သွားရင် main () function  
 ကို ဆက်လက် လေ့လာကြတာပေါ့။

## ကွန်ပူးတာသင်ခန်းစာ

```
JFrame f = new JFrame ("Testing the JAVA
CheckBox"); လို့ ရေးသားထားတာပါပဲ။ အဲဒီ စာသားဟာ
JFrame Windows ရဲ့ Title Text အနေနဲ့ ပါဝင်လာမှာပါပဲ။
ပြီးတော့ လိုင်းနံပါတ် (၂၂) မှာမူ CheckBox Class ကို
Object_Name Chb လို့ နာမည်ပေးပြီးတော့မူ ခေါ်ယူလိုက်တာ
ပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၂၃) မှာမူ JFrame Class ထဲက
Frame Windows တစ်ခုကို လုမ်းပြီးခေါ်ယူအသုံးချပါတယ်။
ဒါကြောင့်
```

```
f.setSize(300,200);
```

လို့ ရေးသားထားတာပါပဲ။

အဲဒီနောက် အပေါ်မှာ ဖုန်းတီး ရေးသားခဲ့တဲ့ checkbox
class ကို add ( ) method နဲ့ ထည့်သွင်း အသုံးချပါတယ်။
အသုံးချရမှာ JFrame Class ထဲက get Content Pane ( )
method နဲ့ ယဉ်တွဲပြီးတော့ အသုံးချကာပါပဲ။ ဒါကြောင့်-

```
f.getContentPane ( ).add (Chb);
```

ဆိုပြီး ရေးသားပေးထားတာပါပဲ။ ပြီးတော့ ကျွန်တော်တို့အပေါ်က
လေ့လာပြီးခဲ့တဲ့ သင်ခန်းစာတွေမှာ အကြောင်းကြိုင် အသုံးချုပြီးခဲ့တဲ့
SetDefault Close Operation ( ) method ကို ထည့်သွင်း အသုံးချပါတယ်။
အကြောင်းကြိုင် အသုံးပြုခဲ့ပြီးတဲ့ method ဖြစ်တာ
ကြောင့် ရှင်းပြစ်ရာ မလိုတော့ပါဘူး။ ရေးရှိးရေးစဉ် code\_line
လိုပဲ ဖြစ်နေပါပြီ။

**Note:** ရှုံးလာမယ့် သင်ခန်းစာ Buftons & Standard Dialog Box အုပ်စုတွေအကြောင်းကို ရေးသားပို့ချုပဲတဲ့အခါမှာ
အခု ရေးသားထားတဲ့ SetDefault Close Operation (JFrame.EXIT\_ON\_CLOSE); ဆိုတဲ့ code\_line ကို
ထည့်သွင်း အသုံးမပြုတဲ့ ရေးသား ပို့ချေပေးထားပါတယ်။
အဲဒီ သင်ခန်းစာတွေမှာ program ကို Run & Compile
လုပ်ပြီးရင် အခု ရေးသားထားတဲ့ program တွေလိုမျိုး
လွယ်လွယ်ကူကူ EXIT & Close လုပ်လိုရမရ လေ့
လာကြည့်ပါ။ ထူးခြားချက်ကို တွေ့ရပါလိမယ်။

ရှုံးလာမယ့် သင်ခန်းစာတွေဖြစ်တဲ့ 3.2.5 Standard Dialog
Box of JAVA components ဆိုတဲ့ သင်ခန်းစာမှာ အသုံးချု
ထည့်သွင်းပေးထားခြင်း မရှိပါဘူး။ အဲဒီအခါမှာ ဘာတွေ ထူးခြား
နေသလဲဆိုတာကို တွေ့ရပါလိမယ်။ Run နေတဲ့ program ကို
ရပ်ချင်ရင်/ close လုပ်ချင်ရင် command prompt (DOS)
windows ကိုပါ အပြီးပိတ်ပစ်မှုသာလျှင် program Run နေတာ
ရပ်သွားမှာပါ။ အဲဒီဟာ အခု အသုံးပြုရေးသားတဲ့-
Set Default Close Operation (JFrame.EXIT\_ON\_-

CLOSE);

ဆိုတဲ့ code\_line ကြောင့်ပါပဲ။ ဒီအချက်လေးကို အထူး

သတိပြုပြီးတော့ မှတ်သားထားပါ။ Standard Dialog Box
အကြောင်း သင်ခန်းစာတွေပြီးရင် ပြန်လည် သတိပေးပြီးတော့
ရှင်းပြုပါမယ်။

က code\_line တွေကို အထက်ပါအတိုင်း Line -By -
Line Analysis လုပ်ပြီးတော့ လေ့လာကြည့်မယ်ဆိုရင်တော့ အရင့်
အရင့် sample program တွေအတိုင်း အလွယ်တကူနဲ့ နားလည်
သဘောပါက်နိုင်မှာပါ။ လိုင်းနံပါတ် (၂၆) မှာတော့ program
ကို Run လိုက်ရင် Output windows ကိုလာပြီး ဖော်ပြပေးနိုင်
ဖို့အတွက် f.show ( ) ဆိုပြီးမှ Show ( ) method နဲ့ ခေါ်ယူ
အသုံးချထားပါတယ်။

လိုင်းနံပါတ် (၂၇) ကတော့ main () function ရဲ့ close
brace } ပါပဲ။ လိုင်းနံပါတ် (၂၈) ကတော့ primary class
checkbox ရဲ့ Close brace } ပါပဲ။ ဒါဆိုရင်တော့ program
code\_line တွေကို Analysis လုပ်တာ ပြီးသွားပါပြီ။ က ဆက်
လက်ပြီးတော့ လေ့လာကြရအောင်လား။

အခြေပြုခဲ့တဲ့ code\_line တွေကို Notepad ထဲမှာ သွားပြီး
တော့ ရိုက်သွင်းပြီးရင် C:\j2sdk 1.4.1\_01\Bin>ဆိုတဲ့ path
လမ်းကြောင်းထဲမှာ သွားရောက်ပြီးတော့ save လုပ်ပါ။ Save
လုပ်ရင် ပေးရမယ့် File\_Name ဟာ Checkbox.java လို့ အမည်
ပေးရပါမယ်။ Extension အနေနဲ့ .java ဆိုတဲ့ File\_Extension
ကို လုံးဝ မမေ့ပါနဲ့နော်။

CheckBox နဲ့ ပတ်သက်တဲ့ Code\_Line တွေကို လေ့လာ
ကြည့်ရအောင်။ အဲဒီ program ဟာ Java ရဲ့ Swing components
နဲ့ Awt Components တွေကို အခြေပြုပြီးတော့
ဖုန်းတီးထားတဲ့ code\_line တွေပါပဲ။ အသေးစိတ် ရှင်းပြထားတဲ့
အချက်တွေကိုတော့ Analysis မှာ လေ့လာလို့ရပါတယ်။

က Listing 3.6 ကို C:\j2sdk 1.4.1\_01\Bin>ဆိုတဲ့
CLASS-PATH အောက်မှာ Checkbox.java လို့ save
လုပ်ပြီးရင်

Start → Program →Accessories →command
prompt ကို သွားဖွဲ့ပါ။ Command line Interface ဖွင့်လာ
မှာပါ။ ဒီအခါမှာ အောက်ပါ command line ကို ရိုက်သွင်းပြီး
တော့ Directory path ကို change လုပ်လိုက်ပါပြီး။

CD C:\j2sdk 1.4.1\_01\Bin ↵

အဲဒီ command-line ကို ရိုက်သွင်းပြီးတော့ Enter key ↵
ကို နှိပ်လိုက်ရင် Java program.code.line ကို Run & Execute
လုပ်ဖို့ အသင့်ဖြစ်နေပါပြီ။ က ဒီတော့

javac checkbox.java ↵

လို့ program ကို compile လုပ်လိုက်ပါ။ အဲဒီအခါကျရင် Error
ကင်းတယ် ဆိုရင်ပေါ့။

C:\j2sdk 1.4.1\_01\Bin > ↵

ဆိတဲ့ promptပဲ ပေါ်လာမှာဝါ။ Error ရှုရင်တော့ Line-Number နဲ့တော့ အကုန်မှန်သွားရင်တော့ ဒီ command line နဲ့ပဲ Program ကို Run လုပ်ထိုက်ပါ။

java check box ↵

ကဲ ဒီအခါမှ Figure 3.11 ကို မြင်ရတဲ့ Output Console Windows မှာလာပြီး ဖော်ပြုပေးမှာပါပဲ။ ဒါဟာ Java ရဲ့ checkbox components & features ကို လက်တွေ လေ့လာကြည့် တာပါပဲ။



Figure 3.11 Testing the Java Checkbox

ဒီ program ငယ်လေးကို လေ့လာကြည့်ရင် Java Language နဲ့ checkbox တွေကို ဖန်တီးတဲ့အကြောင်းပါပဲ။ ဒါကို အထပ်ထပ်အခါခါ လေ့လာပါ။ ကဲ ဆက်လက်ပြီးတော့ Buttons တွေအကြောင်းကို လေ့လာကြည့်ကြရအောင်လား။ Listing 3.7 က code-line တွေကို Notepad မှာ အရင်ဆုံး ရေးသားလိုက်ပါ ဦး။ ပြီးမဲ့ လေ့လာကြတာပေါ့။

### Using the JButtons (OR) Buttons .JButtons (OR) Buttons

Listing 3.7 Demonstrate the Message Box, Using the Buttons

```

1. import javax.swing.*;
2. public class Buttons extends .JFrame {
3. JButton abort = new JButton ("Abort");
4. JButton retry = new JButton ("Retry");
5. JButton fail = new JButton ("Fail");
6. public Buttons () {
7. Super ("Buttons");
8. setsize (300,100);
9. SetDefault. Close Operation (JFrame.EXIT_ON_CLOSE) ;
10. JPanel Pane = new JPanel () ;
11. pane.add (abort);
12. pane.add (retry);
13. pane.add (fail) ;
14. Set Content Pane (Pane);

```

```

15. }
16. public static void main (String args [ ]) {
17. Buttons rb = new Buttons ( );
18. rb.show ( ) ;
19. }
20. }

အခါရေးပြခဲ့တဲ့ code_line တွေဟာ JButtons နဲ့ ပတ်သက်
ပြီးတော့ ရေးသားထားတဲ့ Program Code_Line တွေပါပဲ။ ဒီ
code-line တွေကို Analysis လုပ်ကြည့်ကြရအောင်လား။
Analysis
၁။ ဒီ program ရဲ့ လိုင်းနံပါတ် (၁) မှာ import.java.swing.
* ;components တွေကို ခေါ်ယူအသုံးချဖို့အတွက် import
လုပ် ခေါ်ယူပြီးတော့ Declare လုပ်ပါတယ်။ ပြီးတော့
primary class Buttons ကို ဖန်တီးပါတယ်။ ဖန်တီးရာမှာ
JFrame Class ကို extends လုပ်ပါတယ်။ ဒါကို တွေ့ရ
မှာပါ။ လိုင်းနံပါတ် (၂) မှာ JButton Class ကို ခေါ်ယူရာ
မှာ abort ဆိုပြီး New- Object အနေနဲ့ ဖန်တီးပါတယ်။
ဖန်တီးရာမှာ objectSetText အနေနဲ့ Abort လို့ အမည်
ပေးထားတာပါပဲ။ ဒါကြောင့် "Abort" ဆိတဲ့ စာသားကို
Double quote " " ထဲမှာ ရေးထားတာပါ။ လိုင်းနံပါတ်
(၃) မှာ-

```

JButton abort = new JButton ("Abort");

↑  
objectSetText

```

ဆိုပြီး ရေးသားထားတာပါပဲ။ ဒါဟာ JPanel ထဲက
JButton Class ကို လုမ်းပြီးခေါ်ယူထားတာနော်။ ကဲ အဲဒီ
အတိုင်း Retry/ Fail စတု ခလုတ် Button တွေအတွက်
လည်း ဖန်တီးပါတယ်။ ဒါကြောင့် လိုင်းနံပါတ် (၃) (၄)
(၅)က လိုင်းပုံစံတွေဟာ code_line ပုံစံတွေ ဖြစ်နေတာ
ပါပဲ။ Button_SetText(Button မှာ ပါဝင်ရမယ့် စာသား
အတွက်သာ) ပြောင်းလဲသွားတာ ဖြစ်ပါတယ်။
၂။ Buttons တွေ ပါဝင်တဲ့ Frame_size အတွက် setSize
(300,100); ဆိုပြီးတော့ Width and Height အတွက်
ဖန်တီးပေးထားပါတယ်။ ပြီးတော့ setDefault () method
ကိုလည်း ခေါ်ယူအသုံးချတာကို တွေ့ရမှာပါ။ ဒါဟာ ရှေ့က
program တွေမှာ တွေ့မြင်ပြီးသားနေမှာပါ။ အခါ
Button Class ထဲမှာတင်ပဲ JPanel Class ကို Pane လို့ Object
_Name ပေးပြီးတော့ ခေါ်ယူ အသုံးချပြန်ပါတယ်။

```

Note: JPanel Class & JFrame Class တွေအကြောင်းကို
ရှေ့သင်ခန်းစာတွေမှာ တွေ့မြင်ရပြီးသားဖြစ်ပါတယ်။
နားမလည်သေးရင် အရင်သင်ခန်းစာအဟောင်းတွေကို
ပြန်လုန်လေ့လာနိုင်ပါတယ်။

JPanel Class ကို Pane လို့ Object\_Name ပေးထား  
တာဆိုတော့ JPanel Class ထဲမှာ ပါဝင်တဲ့ add ()  
method ကို ခေါ်ယူအသုံးချရမှာ-

```
Pane.add (abort);
Pane.add (retry);
Pane.add (fail);
```

ဆိုပြီးမှ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ Button တွေကို ခေါ်ယူ  
ထည့်သွင်းပါတယ်။ အဲဒီနောက် set contentPane  
( ) method နဲ့ object ကို Pass By Value လုပ်ပါ  
တယ်။ လိုင်းနံပါတ် (၁၅) မှာတော့ Buttons () con-  
structor function ရဲ့ close brace} ကို ပိတ်ပေးထား  
ပါတယ်။ ဒါ class ရဲ့ ထူးခြားချက်ကတော့ လိုင်းနံပါတ်  
(၆) က constructor function ဖြစ်တဲ့ Buttons ()  
function ထဲမှာ Inheritance Technique ကို ခေါ်ယူ  
အသုံးချထားတာကို တွေ့ရှုပါပဲ။ ဒါကို လိုင်းနံပါတ်  
(၇) မှာ ရေးသားထားတဲ့ -

```
Super ("Buttons");
```

ဆိုတဲ့ code\_line ကို ရေးသားထားတာတွေရင် သဘော  
ပါက်သွားမှာပါ။ အဲဒီ main ( ) function ရဲ့ အတွင်းမှာ  
Buttons Class ကို rb လို့ object\_Name ပေးပြီးတော့  
ဖန်တီးခေါ်ယူပါတယ်။ အဲဒီ rb.show () ဆိုပြီးမှ show  
( ) method နဲ့ ခေါ်ယူလိုက်တာပါပဲ။ အမှန်ကတော့  
Buttons Class ကို object အဖြစ် ရေးဖွဲ့စုံမှာ  
ကတည်းက constructor .function ဖြစ်တဲ့ Bustons  
( ) function ဟာ ဝင်လာပြီးတော့ အလုပ်လုပ်နေပါပြီ။  
ဒါကို show () method နဲ့ ခေါ်ယူလိုက်တဲ့ အခါကျတော့  
Buttons တွေ ပါဝင် ဖွဲ့စည်းထားတဲ့ Output\_Form  
ကို တွေ့မြင်ရတာပါပဲ။

ကဲ ဒီတော့ Analysis လုပ် လေ့လာပြီးရင်တော့ အဲဒီ file  
ကို Notepad ထဲမှာ ရုံက်သွင်းထားတဲ့ပုံကို အောက်ပါ Figure  
3.11.1 အနေနဲ့ လေ့လာလိုက်ပါပြီး။

```
buttons - Notepad
File Edit Format View Help
import javax.swing.*;
public class Buttons extends JFrame{
    JButton abort = new JButton("Abort");
    JButton retry = new JButton("Retry");
    JButton fail = new JButton("Fail");
    public Buttons(){
        super("Buttons");
        setLayout(null);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        pane.add(abort);
        pane.add(retry);
        pane.add(fail);
        setContentPane(pane);
    }
    public static void main(String args[]){
        Buttons b = new Buttons();
        b.show();
    }
}
```

Figure 3.11.1 Type the Buttons. java program in Notepad

ကဲ Notepad က textfile ကို လေ့လာပြီးရင် အဲဒီ file ကို  
system CLASSPATH'sဖြစ်တဲ့ C:\j2sdk 1.4.1\_01\Bin >  
ဆိုတဲ့ Directory ထဲကို ဝင်ရောက်ပြီးတော့ save လုပ်ပါ။ ကဲ  
ပြီးသွားရင်တော့ command prompt ထဲကို ဝင်ရောက်ပြီးတော့  
C:\j2sdk 1.4.1\_01\Bin > ဆိုတဲ့ prompt ပေါ်နေချိန်မှာ  
အောက်ပါ command နဲ့ compile လုပ်ပါ။

```
javac Buttons.java ↵
```

တကယ်လို့ compile လုပ်စဉ်မှာ Error ရှိခဲ့ရင် သက်ဆိုင်ရာ  
line\_No အလိုက် Error တွေကို ရှင်းပြီးရင်တော့ အောက်ပါ  
command နဲ့ Run လုပ်ပါ။

```
java Button ↵
```

အဲဒီလို့ Run လိုက်ရင်တော့ output windows ကို တွေ့မြင်  
ရမှာပါပဲ။ အဲဒီ output windows ကို တွေ့မြင်ရဖို့အတွက် com-  
pile & Run လုပ်ပုံကိုရေး၊ output ထွက်လာပုံကို ပြင်နိုင်ပါ။  
Figure 3.11.2 အနေနဲ့ ရေးပြထားတာကိုပါ လေ့လာလိုက်ပါပြီး။

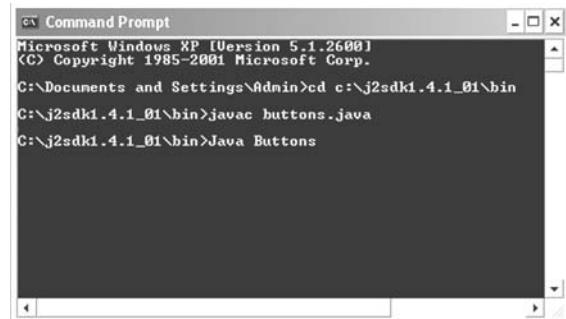


Figure 3.11.2 Demonstrate the compile & Run of Buttons.java

ကဲ အခုလိုမျိုး Buttons တွေအကြောင်းကို နားလည်သော  
ပါက်သွားရင်တော့ ဆက်လက်ပြီး JAVA ရဲ့ standard Dia-  
log Box တွေအကြောင်းကို ဆက်လက်လေ့လာကြတာပေါ့။  
ဒါကြောင့် Standard Dialog Box တွေအကြောင်းကို လေ့လာ  
လိုက်ပါပြီး။



### → 3.2.5 STANDARD DIALOG BOX OF JAVA COMPONENTS

#### Standard Dialog Box From Standard Dialog Boxes Group

Java Language ရဲ့ Button တွေအကြောင်းကို ရှင်းပြေား  
တဲ့အခါမှာ ဆက်နွယ်နေတဲ့ Standard Dialog Box တွေ  
အကြောင်းကို သင်ကြား ဂို့ချပေးချင်ပါတယ်။ အခုလက်ရှိ  
ကျွန်တော်တို့အသုံးပြုနေတဲ့ JOptionPane Class ဟာဆိုရင်  
အမျိုးမျိုးသော standard Dialog Box တွေကို ဖန်တီးပေးနိုင်  
ပါတယ်။ စာဖတ်သူအနေနဲ့ သံသယမရှိဘဲနဲ့ JOptionPane  
class ကို အသုံးချဖြိုးတော့ Standard Dialog Box အမျိုးမျိုးကို  
ဖန်တီးရယူနိုင်ပါတယ်။ ဥပမာပြောရရင်တော့-

သင့်ရဲ့ computer system မှာ system Crashes ဖြစ်  
သွားခဲ့ရင် Operation System ကနေဖြိုးတော့ Dialog Box  
တစ်ခု ပေါ်လာဖြိုးတော့ သတိပေးလာမှာပါ။ ဒါဟာ Message  
Box ရဲ့ သဘောတရားပါပဲ။

တကယ်လို့ စာဖတ်သူဟာ Deleting Files လုပ်တယ်ဆိုကြ  
ပါနဲ့ Dialog Box တစ်ခု ပေါ်လာဖြိုးတော့ -

To make sure that you really want to do that?  
မင်းအမှန်တကယ် ဖြစ်ပျက်မှုလားလို့မေးတဲ့ Dialog Box  
ပေါ်လာပါတယ်။ ဒါဟာ Confirm Message Dialog Box  
ရဲ့ သဘောတရားပါပဲ။

ဒါဟာ အခြေခံကျကျ အကြမ်းဖျင်းအားဖြင့် ရှင်းပြတာပါ။  
အပေါ်မှာ ရှင်းပြခဲ့တဲ့ Message Box တွေရဲ့ သဘောတရားကို  
ဘယ်လို့ အမျိုးအစား operating system (windows, Mac,  
Linux) ကိုမဆို၊ သုံးစွဲသူကိုမဆို တွေ့ဖြင့်ဖြိုးသားဖြစ်တဲ့ သဘော  
တရားတွေမှိုပါပဲ။

တကယ်လို့ Standard Dialog Box ကို အပ်စွဲဖြိုးတော့  
လေ့လာမယ်ဆိုရင် အောက်မှာ ရေးသားပို့ချပေးထားတဲ့ Table  
3.6 ကအတိုင်း တွေ့ရမှာပါ။

Type	Description
confirmDialog	Asks a question, with buttons for Yes, No, and Cancel responses
InputDialog	Prompts for text input
MessageDialog	Displays a message
OptionDialog	Comprises all three of the other dialog box types

Table 3.6 Types of the standard Dialog Boxes

Notes : Each of these dialog Boxes has its own method in the JOptionPane Class included

ကဲ ဒီတော့ အခု တွေ့ဖြင့်ရတဲ့ Table .3.6 ထဲက Dialog  
Box တစ်ခုချင်းစီကို သေသေချာချာ အပ်စွဲဖြိုးတော့ လေ့လာကြရ<sup>1</sup>  
အောင်လား။

#### Confirm Dialog Boxes

အလွယ်ကူဆုံးသော နည်းလမ်းနဲ့ Yes/No/Cancel ဆိုတဲ့  
Buttons တွေပါဝင်တဲ့ Dialog Box ကို confirm - Dialog  
Box ကို ဖန်တီးနိုင်ပါတယ်။ အသုံးချမယ့် method ကတော့ -  
ShowConfirmDialog (components , Object);

ဆိတဲ့ method ပါပဲ။ Components ဆိတဲ့ argument ကတော့ Container Class ကို အသုံးပြုရမှာပါ။ Container class ကို အထူးပြု မို့ဖြစ်းပြီးတော့ (Parent Class အနေနဲ့ မို့ဖြစ်းတာပါ) Dialog Box ကို ဖန်တီးတာပါပဲ။ ဒါဟာ အသုံးပြုနေသူ user ကို determine လုပ်နိုင်ဖို့အတွက် display လုပ်ပေးတာပါပဲ။ ဒါကို သေသေချာချာ မှတ်ထားနော်။

တကယ်လို့ အခု အသုံးချေနေတဲ့ Container Class (component argument) နေရာမှာ - null ဆိတဲ့ statement ကို အသုံးချု ထည့်သွင်းလိုက်ရင် ဒါမှမဟုတ် Container (component) ဟာ JFrame Class လဲ၊ ထောက်ပံ့မှုကရော ကင်းလွတ် နေမယ်ဆိုရင်တော့ Dialog Box ဟာ စာဖတ်သူရဲ့ screen အလယ်တည့်တည့်မှာ လာပြီးတော့ ပြေားပါလိမ့်မယ်။

Notes : ဒီနည်းလမ်းကိုပဲ အသုံးချြိုးတော့ လက်ရှိ standard Dialog Box တွေမှာ အသုံးပြု သင်္ကြား ပို့ချေပေးထားပါတယ်။ ဒါကြောင့် ရှင်းပြတာကို နားလည် အောင် ဂရုစိုက်ပါ။

ကဲ- ဆက်လက်ပြီးတော့ ခုတိယ arguments ဖြစ်တဲ့ Object ဆိတဲ့ arguments ကို လေ့လာကြရအောင်လား။ အဲဒီ object ဆိတဲ့ arguments နေရာမှာ့-

Dialog Box မှာ ပါဝင်စေချင်တဲ့ (ပြောချင်တဲ့) string စာသား components ဒါမှမဟုတ် Icon Object တွေကို ထည့်သွင်း ဖန်တီးနိုင်ပါတယ်။

အဲဒီ second argument နေရာမှာ ထည့်သွင်းလိုက်တဲ့ String, Text ဟာသုတေသန Dialog Box မှာ ပါဝင်ပြီးတော့ လာပြပေးမှာပါ။ ကဲ အဲဒီတော့ အောက်မှာ ရေးသားပို့ချေပေးထားတဲ့ example ကို အသုံးပြုပြီးတော့ Confirm Dialog Box တစ်ခု ကို ဖန်တီး ရေးသားပြီးတော့ လေ့လာကြည့်ကြရအောင်။ Listing 3.8 ကို လေ့လာကြည့်ကြရအောင်လား။

#### Listing 3.8 Demonstrate the Confirm Dialog Box

```

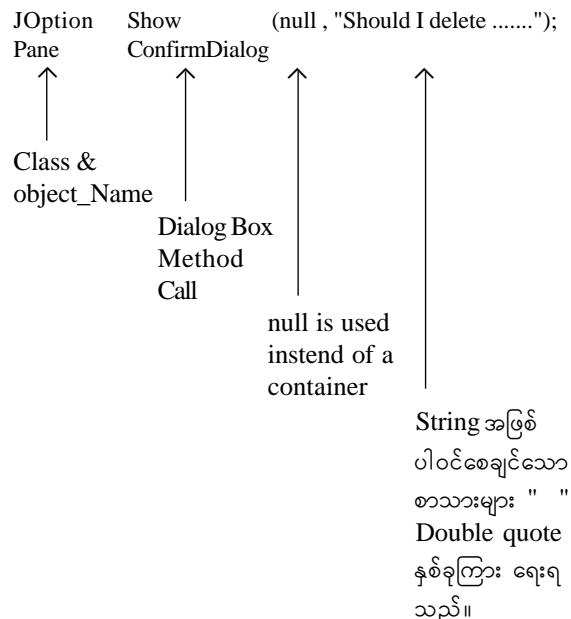
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class dialogs extends JFrame{
5. public static void main (String args [ ]){
6. int response;
7. response = JOptionPane.show ConfirmDialog
(null,
8. "Should I delete all of your irreplaceable per-
sonal files?");
9. }
10. }
```

ကဲ code\_line တွေကို ရေးပြီးသွားပြီဆိုတော့ Analysis လုပ်ပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။ သေသေချာချာ ရှင်းပြပေးထားပါတယ်။

#### Analysis

၁။ GUI components တွေကို တည်ဆောက်ရမှာ အရေးပါတဲ့ packages files တွေဖြစ်တဲ့ swing နဲ့ awt အစဉ်တဲ့ package file နှစ်ခုကို import လုပ် ခေါ်ယူပါတယ်။ လိုင်းနံပါတ် (၄) မှာမူ JFrame Class ကို မို့ဖြစ်းပြီး extends လုပ်ကာ dialog 1 ဆိတဲ့ class ကို ဖန်တီး ရေးသားပါတယ်။ အဲဒီ Class ထဲမှာမူ main () function ကို ရေးဖွံ့ဖြိုးပါတယ်။ ဒီတော့ Dialog 1 ဆိတဲ့ primary class ကို ဖန်တီး ရေးသားပါကို နားလည် သဘောပါက်သွားပြီနော်။

၂။ လိုင်းနံပါတ် (၆) မှာတော့ int response ဆိုပြီးမူ return value ပြန်ပေးရမယ် value အတွက် int constant value ကို Declare လုပ်ပါတယ်။ ဒါကို နားလည် သဘောပါက်ပြနော်။ အဲဒီနားလည် variable \_Name ဖြစ်တဲ့ response ကို အသုံးပြုပြီးတော့မှ-  
response = JOptionPane .Show ConfirmDialog  
(null,  
"Should I delete all of your irreplaceable personal  
files?"); ဆိုပြီး ရေးသားထားပါတယ်။ JOptionPane Class ထဲက-



က အခ ရေးပြီးတော့ ရှင်းပြခဲ့တဲ့ show ConfirmDialog () method ကို အခြေခံပြီးတော့မှ ရေးသားထားတာကို နားလည် သဘောပါက်ပြီးနော်။ စာဖတ်သုဖြစ်စေချင်တဲ့ String တွေပါဝင် တဲ့ confirm Dialog Box တွေကိုလည်း ရေးသားဖန်တီးလို့ရနိုင် တာပေါ့နော်။

က ဒီတော့ အပေါ်မှာ ရေးသားခဲ့တဲ့ JAVA Program code\_line တွေကို Notepad Program ထဲမှာ သွားရောက်ပြီး တော့ ရေးသားလိုက်ပါ။ ပြီးရင်တော့ File\_Name ကို dialog1.java လို့ အမည်ပေးပြီးတော့မှ save လုပ်ပါ။ အဲဒီလို Code\_Line တွေ ရေးသားပြီးတော့ save လုပ်ထားပုံကို Figure 3. အနေနဲ့ ရေးသားပေးထားပါတယ်။ လေ့လာကြည့်လိုက်ပါပြီး။

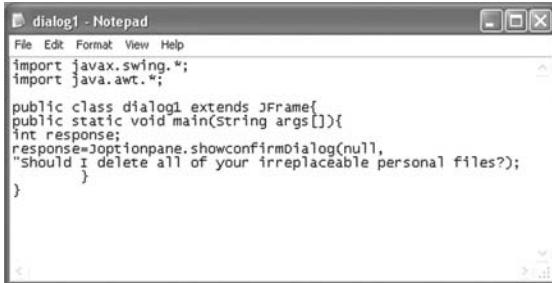
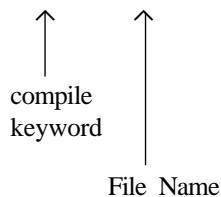


Figure 3.12 Write the program code\_line in the Notepad

က အခ ဖွေ့ဖြိုးတွေ့မြင်ရတဲ့ Figure 3.12 ဟာဆိုရင်တော့ JAVA code\_line တွေကိုရေးသားပြီး save လုပ်ထားတဲ့ Notepad ရဲ့ပုံပါပဲ။ က ဒီတော့ ဆက်လက်ပြီး လေ့လာကြတာပေါ့နော်။ က ဆက်လက်ပြီးတော့ Java code\_line တွေကို ရေးသားပြီး တော့ save လုပ်ပြီးရင်တော့ command Prompt (DOS) ထဲကို ဝင်ပြီးတော့ အောက်ပါအတိုင်း JAVA ရဲ့ CLASS\_PATH'S ထဲကို ဝင်ပြီးတော့ compile လုပ်ပါ။

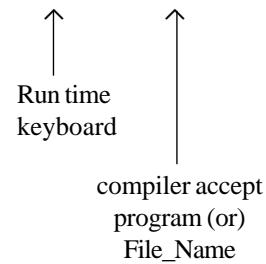
C:\j2sdk1.4.1\_01\Bin > javac dialog1.java



အဲဒီအတိုင်း Compile လုပ်လိုက်ရင် error မရှိဘူးဘဲ ဆိုကြပါစိုး။ error ရှိရင်လည်း NotePad file ထဲကို ဝင်ရောက်ပြီးတော့ အမှားပြင်/ နောက်ထပ် Save As လုပ်ပြီးတော့ ပြန်ပြီးတော့ compile လုပ်ပါ။

Compile လုပ်တာ Error မရှိတော့ဘဲ compiler ကနေ လက်ခံလိုက်ပြီး ဆိုကြပါစိုး။ အောက်ပါအတိုင်း Program ကို Run လုပ်ပါ။

C:\j2sdk1.4.1\_01\Bin > javac dialog1



အဲဒီအခါကျရင်တော့ Figure 3.12.1 ရဲ့ မြင်ရတဲ့အတိုင်း command line Interface (OR) DOS Windows ရဲ့ အလယ် တည်တည်မှာ အောက်ပါအတိုင်း Confirm Dialog Box ဟာ လာရောက်ဖော်ပြပြီး Run လုပ်နေခြားပါ။

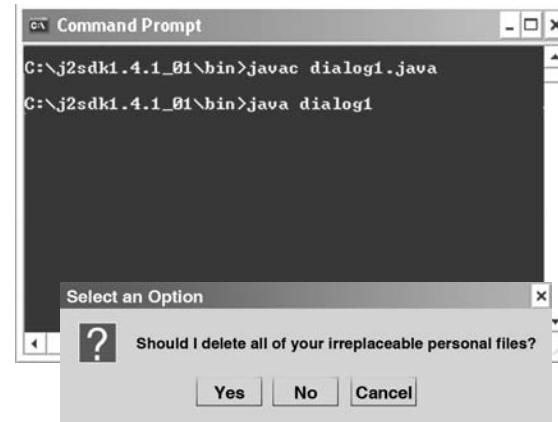


Figure 3.12.1 Compile & Run of the dialog1.java [show Confirm Dialog Box]

က အခရေးပြထားတဲ့ code\_line တွေရယ်။ File တွေကို Save လုပ်ပြီးတော့ Compile & Run လုပ်ပုံတွေရယ်ကို သေသေ ချာချာ တွက်ချက်လေ့လာပါ။ Analysis ကိုလည်း သတိပြု လေ့လာပါ။ ဒါဆိုရင်တော့ ရှုံးလာမယ့် standard Dialog Box အပ်စုစင်တွေဖြစ်တဲ့ Input Dialog Box, Message Box, Option Dialog Box အစုရိတာတွေကို နားလည် သဘော ပေါက်သွားပါလိမ့်မယ်။ က ဆက်လက်ပြီးတော့ လေ့လာကြည့်ပါဦးစိုး။

### 3.2.6 SUB-TITLE OF CONFIRM DIALOG BOX

က ဆက်လက်ပြီးတော့ Confirm Dialog Box ကို အခြား  
နည်းလမ်းတစ်ခုနဲ့ ရေးသား ဖန်တီးကြည့်ကြရအောင်လား။  
Confirm Dialog Box ကို အခြားနည်းလမ်းတစ်ခုနဲ့ ရေးသား  
မယ်ဆိုရင် ဘာတွေလိုအပ်မလဲ။ အဲဒီအသုံးမျမှတ်  
Method မှာ ကွဲပြားခြားနားမှုတွေ ရှိနိုင်သလား။ အောက်မှာ ရေးပြထားပါ  
တယ်။ လေ့လာကြည့်ပါ။

Show Confirm Dialog (Component, object, string, int, int, int);

အရှင်သင်ခန်းစာများ ပို့ချလေးခဲ့တဲ့ Sample program မှာ  
လိုပါပဲ။ အရှင် arguments နှစ်ခုဟာ အရင်လိုပါပဲ။ ဒါဟာ  
အခြားသော confirmDialog Box များမှာလိုပါပဲ။ အသုံးချမဆယ့်  
နည်းပညာပါ။ က အသုံးပါဝင်လာတဲ့ arguments အနေနဲ့က  
(၃) ခရီးတယ်နော်။ အဲဒါတွေကို ရှင်းသွားအောင် လေ့လာကြည့်ကြ  
ရအောင်လား။

Show Confirm (Component, Object, String, int, int);  
 Dialog  
 မူလသင်ခန်းစာတွင်  
 ရင်းလင်းပေးခြား  
 listing 3.8              New\_argument 1  
 New\_argument 2  
 New\_argument 3

ကဲ ဒီတော့ New\_arguments တွေကို လေ့လာကြရအောင်  
လူး။



drawing ලැබුවයින් (ඇමගහුත්) determine ලැබුවයින්  
 තු arguments එකිනෙකුව හා ගිණු නො යොමු කිරීමේදී පෙන්වනු ලබයි  
 මෙයින් අනුමත වූ ඇත්තා නො යොමු කිරීමේදී පෙන්වනු ලබයි  
 මෙයින් අනුමත වූ ඇත්තා නො යොමු කිරීමේදී පෙන්වනු ලබයි

Listing 3.9 မှာ ရေးသားထားတဲ့ Sample program code\_line တွေကို အရင်ဆုံး လျှောက်ပြုလိုက်ကြရအောင်။

Listing 3.9 Demonstrate in ShowConfirmDialog in  
any other method

- ```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class dialog 2 extends J Frame {
5. public static void main (string args [ ]) {
6. int response ;
7. response = J Option>Show Confirm Dialog (null,
8. "Error reading file .Want to try again? ";
9. "File Input Error";
10. JOption Pane. YES _NO_ OPTION,
11. JOptionPane.ERROR_MESSAGE);
12.
13. }
14. }
```

Note : show ConfirmDialog (Components, Object);  
          show ConfirmDialog (Component, Object,  
          String, int, int);

## Analysis

၁။ ဒီ program ကို လေ့လာကြည့်ကြရအောင်။ Swing Components နဲ့ AWT components တွေကို အသုံးချဖိုင် ဖို့အတွက် package file နှစ်ခုကို Declare လုပ်ပါတယ်။ ပြီးတော့ လိပ်ငါးနံပါတ် (c)မှာမူ dialog 2 ဆိတ် Class ကို

## ကွန်ပူးတာသင်ခန်းစာ

ဖန်တီးပါတယ်။ အဲဒီ Class ကို ဖန်တီးရမှာ `JFrame` Class ကို extends လုပ်ထားတာဖော်။ ဦးတော့ `main()` function ကို ရေးဖွဲ့ထားပါတယ်။ အဲဒီ `main()` function ထဲမှာမူ int data type ကိုအသုံးချဖြီးတော့ `int response;` ဆိုပြီး variable ကို Declare လုပ်ပါတယ်။ အဲဒီလိုင်းနံပါတ် (၇) မှာ ရေးသား Declare လုပ်ခဲ့တဲ့ `response` ဆိုတဲ့ variable ဟာ `show ConfirmDialog()` method ကို ကိုယ်စားပြုခဲ့တဲ့/ကိုယ်စားပြုမယ့် variable ပဲ ဖြစ်ပါတယ်။

JII ဒါကြောင့် လိုင်းနံပါတ် (၇) မှာ-  
`response = JOptionPane.show ConfirmDialog(null;".....လို့ ရေးသားထားတာဟာ variable_Name ဟာ ကိုယ်စားပြု ရေးဖွဲ့ထားတာကိုပါပဲ။ Double quote " " ထွေ ကြားကနေ ရေးဖွဲ့ထားတဲ့ စာသား String တွေက တော့ Message Box မှာ ပါဝင်စေချင်တဲ့ စာသားပါပဲ။ ဒါကြောင့် "Error reading file .Want to try again?" ဆိုတဲ့ String ဟာ Message ပေးရမယ့် String ပါပဲ။ အဲဒီနဲ့ရောက်မှာ ပါဝင်တဲ့ String ဖြစ်တဲ့ လိုင်းနံပါတ် (၉) မှာ ရေးသားထားတဲ့ "File Input Error", ကတော့ Message Box ရဲ့ Title Bar မှာ ပါဝင်စေချင်တဲ့ စာသား String ပါပဲ။ က ဒီအထိ ရှင်းပြုနေ၏ လိုင်းနံပါတ် (၁၀) မှာ ရေးသားထားတဲ့ JOptionPane.YES_NO_OPTION, ကတော့ Message Box မှာ ပါဝင်ရမယ့် Option Button တွေကို တပ်ဆင်ပေးနိုဖြစ်ပါတယ်။ Class Constant အနဲ့ YES_NO_OPTION ကို ပေးထားတာပါပဲ။ က ဒီအထိ နားလည်းသဘောပေါက်ပြုနေ၏။ ဆက်လက် လေ့လာကြရ အောင်။`

က ဆက်လက်လေ့လာကြည့်ရင်တော့ လိုင်းနံပါတ် (၁၁) မှာ ရေးသားထားတဲ့

`JOptionPane. ← Show ConfirmDialog ()  
ERROR_MESSAGE);` function ၏ Close Brace ဖြစ် ပါတယ်။

ဆိုပြီး ရေးသားထားတာကတော့ `MESSAGE` ဟာ ဘယ်လို အမျိုးအစား `MESSAGE` လဲဆိုတာ သိနိုင်ဖို့အတွက် `determine` လုပ်ပေးမယ့် argument ပါပဲ။ ဒါအဲရင်တော့ ရေးသားထားတဲ့ `program code_line` ကို နားလည်းသဘောပေါက်သွားမှာပါ။ လိုင်းနံပါတ် (၁၂) ကတော့ `Blank space line` ချုပ်ထားပါ။

Note : Dialog Box နဲ့ ပတ်သက်တဲ့ argument (OR) parameter တွေကို ရေးဖွဲ့ရမှာ arguments တွေကို comma (,) ခံထားပြီးတော့ တစ်လိုင်းစီခွဲပြီးတော့ ရေး

သားလို့ရပါတယ်။ အခိုက်အချက်ကတော့ comma, ခံပြီး ရေးရတာပါပဲ။ အဆုံးမှာ `showConfirmDialog` ( )method ရဲ့ အပိတ် bracket ( ) ကို ပါရပါမယ်။ လိုင်းနံပါတ် (၁၃) မှာတော့ `main()` function ရဲ့ close brace } ပါပဲ။ လိုင်းနံပါတ် (၁၄) မှာတော့ primary class dialog2 ရဲ့ close brace } ပါပဲ။

က အဲဒီအတိုင်း Analysis တွေကို လေ့လာလိုက်ရင်တော့ `ERROR_MESSAGE` အတွက် ရေးသားရတဲ့ program ဟာ အလွန်လွယ်ကူတယ်ဆိုတာကို တွေ့ဖြင့်ရမှာပါ။ ဒါအဲရင် Note pad ထဲမှာ သွားရောက်ပြီးတော့ ရေးသားထားပုံကို စာဖတ်သူ အနေနဲ့ ရေးသား save လုပ်ထားပုံကို Figure 3.13 အနေနဲ့ ရေးသားပြီးတော့ ဖော်ပြပေးထားပါတယ်။

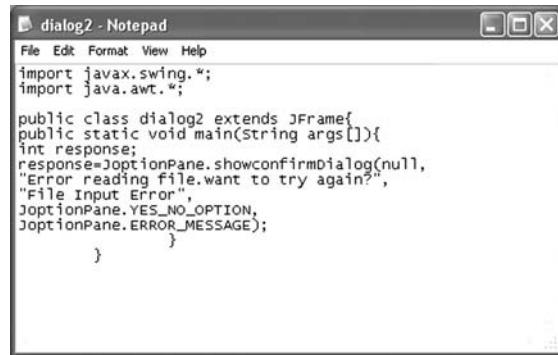


Figure 3.13 Write the Program code\_line in the Notepad.

က အခါ ဖော်ပြပေးထားတဲ့ Notepad ကို လေ့လာပြီးရင် တော့ အဲဒီ dialog2.java ဆိုတဲ့ File ကို compile လုပ်ကြရ အောင်လား။ ဒီတော့ အရှင်စုံး command line windows (DOS) ထဲကို သွားပါ။ အဲဒီထဲရောက်ရင် အောက်ပါအတိုင်း File\_path's လမ်းကြောင်းကိုပြောင်းပါ။

CD C:\j2sdk 1.4.1\_01\Bin ↵

ဒါအဲရင်တော့ computer Directory လမ်းကြောင်းဟာ အောက်ပါ prompt ပုံစံအတိုင်း ပြောင်းလဲသွားမှာပါ။

C:\j2sdk 13.4.1\_01\Bin ↵

က ဒီတော့ အဲဒီလို prompt ပေါ်နေချိန်မှာ အောက်ပါ အတိုင်း command-line ကို ရှိကိုသွင်းပါ။

javac dialog2.java ↵

အဲဒီအတိုင်း compile လုပ်လိုက်ရင်တော့ dialog 2.java ဆိုတဲ့ code\_file ကို compile လုပ်မှာပါ။ အဲဒီလို compile လုပ်လိုက်လို့ Error ရှိရင်တော့ ပြန်ဖြင့်/save လုပ်ပြီးတော့ compile ပြန်လုပ်ပါ။ တကယ်လို့ Error မရှိတော့ဘူးဆိုကြပါစိုး။ အောက်ပါအတိုင်း Run ရန်အတွက် command-line ကို ရိုက်သွင်းပါ။

java dialog 2 ↵

ဒါဆိုရင်တော့ စာဖတ်သူရေးသားထဲ့ program code-line ရဲ့ Output ဟာ အောက်မှာဖော်ပြထားတဲ့ Figure 3.အတိုင်း စာဖတ်သူရဲ့ Computer Screen အလယ်တည့်တည့်မှာ Output အနေနဲ့ ဖော်ပြပေးမှာပါပဲ။

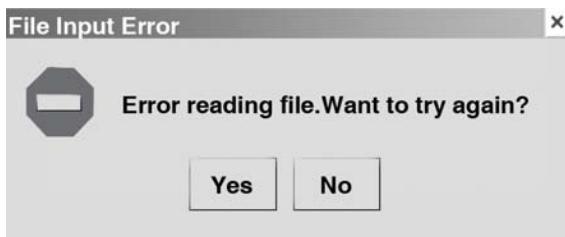
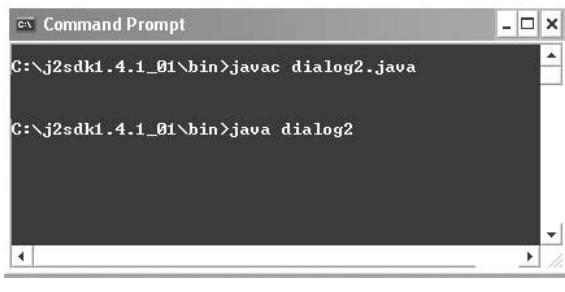


Figure 3.13.1 Compile & Run the dialog2.java [Show ConfirmDialog Box]

ကဲ ဖော်ပြထားတဲ့ Figure 3.13.1 ကို လေ့လာကြည့်ခြင်း အားဖြင့် ConfirmDialog Box အမျိုးအစား အပ်စွာဝင် Dialog Box (2) မျိုးကို ရေးသား လေ့ကျင့်တဲ့ နည်းလမ်းပါပဲ။

ဆက်လက်ပြီးတော့ Standard Dialog Box အပ်စွာဝင်အမြားနည်းလမ်းကို လေ့လာ လေ့ကျင့်ကြတာပေါ့။ သေသေ ချာချာ လေ့လာကြည့်ပါ။ အလွန် အသုံးဝင်တဲ့ နည်းလမ်းတွေပါပဲ။

## ဘီးကာလေးကို စီးပြီး

စက်ဘီးစီးရင်း တေးသီချင်းနားထောင်ဖို့ဆိုတာ အလွန် အန္တရာယ်များပါတယ်။ တေးဖွဲ့စက်ကို လက်နဲ့ ကိုင်ထားလို့ လည်းမဖြစ်၊ အိတ်ကာင်ထဲ ထည့်ထားရင်လည်း

အဆင်မပြုတတ်ပါ။ ဒါပေမယ့် Hammacher Schlemmer ကုမ္ပဏီက Bicycle MP3 player လို့ အမည်ပေးထားတဲ့ MP3 စက်ကယ်ကလေးကို မိတ်ဆက် ထုတ်လုပ်လိုက်ပါတယ်။ စက်ဘီးရဲ့လက်ကိုင်မှာ တပ်ဆင်ထားနိုင်ပြီး တေးဂါတာ အီအစဉ်များကို 1GB ထထိ ထည့်ထားနိုင်ပါတယ်။ စပိကာလည်းပါလို့ သင့်အနေနဲ့ နားကြပ်ကို သုံးစရာ မလိုတော့ပါ။ သူကို ထည့်ထားတဲ့ ကာဗာအိတ်ကလည်း မိုးဒက်၊ ရေဒက်ခံနိုင်ပြီး စွမ်းအင်အတွက် စက်ဘီးလည်းရာမှာ တပ်ဆင်နိုင်တဲ့ ခိုင်နမိုလို ကိုရိုယာကို သုံးနိုင်ပါတယ်။ ဒါမှာဟုတ် AAA အမျိုးအစား စတ်ခဲ င လုံး ထည့်ပြီး ဖွင့်နိုင်ပါတယ်။ နောက် မလိုအပ်ရင်လည်း ဖြေတွဲပြီး အိမ်ကွန်ပူးတာနဲ့ ချိတ်ဆက်သုံးနိုင်ပြီး SD/MMC cards များကိုလည်း တိုးချဲ့သုံးလို့ရပါတယ်။ အချိုအစားက ၈x၁၂၅x၄၀၅ လက်မဖြစ်ပြီး တန်ဖိုး ဒေါ်လာ ၁၃၀ လို့ ရေးထားပါတယ်။



KYAW ZAYAR LAY [ IT ENGINEERING ]



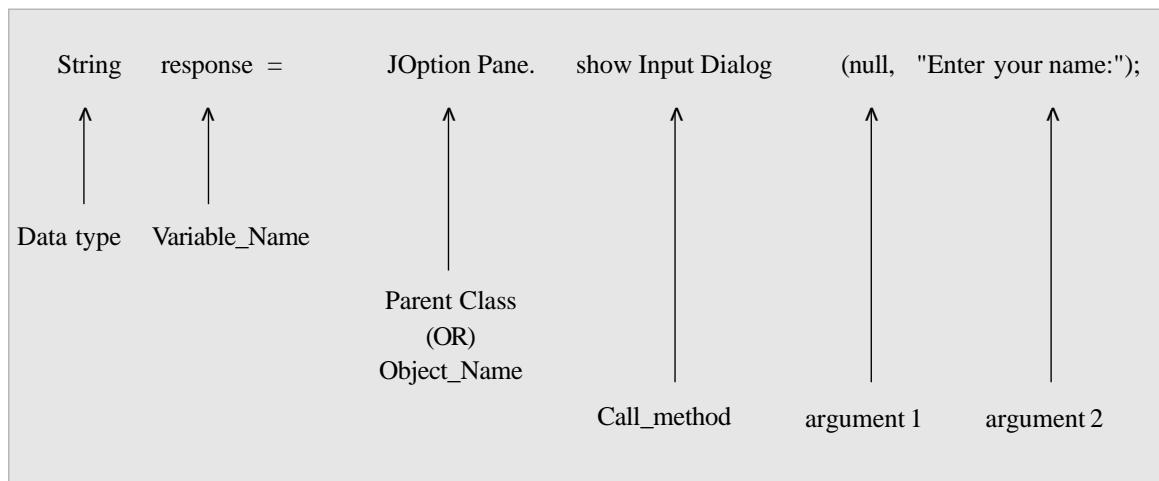
### 3.2.7 Show Input Dialog Box

ဒီတော့အလွယ်ရင်းပြရင် Input Dialog Box ဆိတာကတော့ user ကို question လာမေးတဲ့အခါမှာ text.field မှာ Input Data တွေကို store လုပ်ပါ။ response လုပ်တာပါပဲ။

Input Dialog Box ကို ခေါ်ယူအသုံးချမယ်ဆိုရင်

ShowInputDialog (Component, Object);

အထက်ပါ method ကို ခေါ်ယူအသုံးချရမှာ ဖြစ်ပါတယ်။ ဒီတော့ ShowInput Dialog Box ထဲမှာ ထည့်သွင်း အသုံးချ ထားတဲ့ arguments တွေကတော့ parent component and string, ဒါမှမဟုတ် components or icon တွေကို display လုပ်ပေးနိုင်ဖို့အတွက် ရေးသားဖော်ပြပေးထားတဲ့ component & arguments တွေပဲ ဖြစ်ပါတယ်။



ဒါနိုရင်တော့ InputDialog Box ကို ဖန်တီး၏ယူမယ့်အခြေခံနည်းလမ်းပါပဲ။ String Data type ကို အသုံးချဖြိုးတော့ response ဆိတ် variable ကို ကြည်းပါတယ်။ ပြီးတော့ JOptionPane Class ထဲကနေ အသုံးချမယ့်နည်းလမ်းဖြစ်တဲ့ Show Input Dialog ( ) method ကို ၇၅ယူအသုံးချပါတယ်။ Argument 1 နေရာမှာ ဘာမှ ထည့်သွေးမထားတဲ့အတွက်ကြောင့် showInputDialog Box ဟာ လူပြီးပေါ်လာရင် Screen

အလယ်တည့်တည့်မှု လာမြို့တော့မှ Output ကို show လုပ်လေးမှာပါ။ အနေအက်က argument 2 ကတော့ InputDialog Box မှာ ပါဝင်စေချင်တဲ့ ဘာသားပါပဲ။

ကဲ ဒီအထိ ရင်းပြေနော်။ အခြား method တစ်ခုအနေနဲ့ဆိုရင်  
တော့ showInputDialog() method ကို အောက်ပါနည်းလမ်း  
အနေနဲ့လည်း ပေါ်ယူ အသုံးချိန်ပါတယ်။



**Show Input Dialog**  
`(Component, Object, String, int);`

ဒီ method မှာပါဝင်တဲ့ First & Second arguments တွေဖြစ်တဲ့ Component, Object တွေကတော့ အပေါ်မှာ ရှင်းပြခဲ့တဲ့ method တွေပါပဲ။ ကဲ နောက်ထပ် ပါဝင်လာတဲ့ Third & Fourth arguments တွေကတော့

- The title to display in the dialog box title bar.
- One of five class constants describing the type

```
String response = JOptionPane . showInputDialog (null,
    "What is your IIP code?",
    "Enter ZIP code",
    JOptionPane.QUESTION_MESSAGE);
```

အထက်ပါ code\_line ကို လေ့လာကြည့်ရင် အောက်ပါ argument (4) ခုစုံတဲ့ ကို စနစ်တကျ အသုံးပြထားတာကို တွေ့ရ မှာပါ။ ဒါဆိုရင်တော့ အခြေခံ method တွေကို လေ့လာပြီးသွားပြီးဆိုတော့ ShowInputDialog () method ကို အသုံးချဖြီးတော့ program တစ်ပုဒ်လောက် ရေးကြည့်ကြရအောင်လား။

**Listing 3.10 Demonstrate the ShowInputDialog ( ) method**

- import javax.swing.\*;
- import java.awt.\*;
- 
- public class dialogs extends JFrame {
- public static void main (string args [ ]) {
- String response;
- response = JOptionPane.ShowInputDialog
- (null,
- "Enter your name:") ;
- }

of dialogbox: ERROR\_MESSAGE, INFORMATION\_MESSAGE, PLAIN\_MESSAGE, QUESTION\_MESSAGE, or WARNING\_MESSAGE.

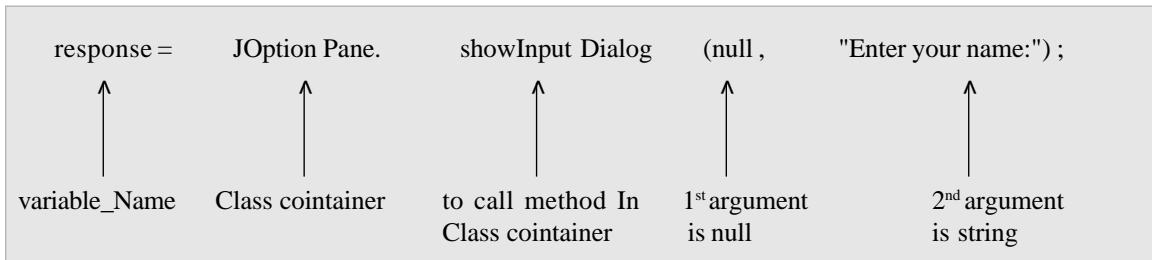
ဆိုပြီးတော့ JAVA Developer Guide Book မှ ရှင်းပြပေးထားပါတယ်။ ဒါတော့ အခုတင်ပြထားတဲ့ second-method ကို အခြေခြားပြီးတော့ လေ့လာကြည့်မယ်ဆိုရင် အောက်ပါ အခြေခံ code\_line အနေနဲ့ လေ့လာလို့ ရနိုင်ပါတယ်။

```
String response = JOptionPane . showInputDialog (null,           argument 1
    "What is your IIP code?",           argument 2
    "Enter ZIP code",           argument 3
    JOptionPane.QUESTION_MESSAGE);           argument 4
```

#### Analysis

၁။ လိုင်းနံပါတ် (၁) ကတော့ java ရဲ့ swing component ကို ခေါ်ယူထားတဲ့ package file ပါပဲ။ လိုင်းနံပါတ် (၂) ကတော့ out component ကို ခေါ်ယူအသုံးချဖြီးထားတဲ့ Declare လုပ်ထားတဲ့ package ပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၃) ကတော့ line-spacing ချန်ထားတာပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၄) မှာတော့ primary class dialogs ဆိုတာကို ဖော်တိုးပါတယ်။ Dialogs ဆိုတဲ့ primary Class ကို ဖော်တိုးရာမှာ JFrame class ကို extends လုပ်ထားတာ ပါပဲ။ ဒီအချက်ဟာ လုံးဝ မေချန်ထားလို့ မရတဲ့ အရေးကြီးတဲ့ အချက်တစ်ချက်ပါပဲ။ ကဲ အဲဒီနောက် လိုင်းနံပါတ် (၅) မှာမှ main () function ကို ရေးသားပါတယ်။

၂။ ကဲ main () function ထဲမှာ ရေးသားထားတာကို လေ့လာကြည့်ကြရအောင်။ String Date type ကို အသုံးပြုပြီးတော့ response ဆိုပြီး variable\_Name ကို Declare လုပ်ပါတယ်။ ဒီအခါမှာ response ဆိုတဲ့ variable\_Name ကို အသုံးပြုပြီးတော့ JOptionPane ထဲက ShowInputDialog () method ကို ခေါ်ယူပြီးတော့ အသုံးချပါတယ်။ ဒါဟာ အပေါ်မှာ ရှင်းပြခဲ့သလိုပါပဲ။



က အဲဒီလို လေ့လာလိုက်တော့လည်း ShowInputDialog() method ဟာ အလွန်လွယ်ကူသွားပါတယ်။ စာဖတ်သူ အနေနဲ့လည်း နောက်ပိုင်းမှာ အလွယ်တကူ လေ့လာလို ရသွားဖိုပါ။ " " Double quote နှစ်ခြားက string ဖြစ်တဲ့ Enter your name ဆိုတဲ့ နေရမှာ မိမိဖန်တီးမယ် program နဲ့လိုက်ဖက်တဲ့ စာသားကို ထည့်သွင်းရေးသားလို ရပါတယ်။ လိုင်းနံပါတ် (၉) ကတော့ main() function ရဲ့ close brace } ပါပဲ။ လိုင်းနံပါတ် (၁၀) ကတော့ primary class dialog3 ရဲ့ close brace } ပါပဲ။

က program code\_line တွေကို လေ့လာပြီးသွားရင်တော့ အဲဒီ code\_line ကို Notepad မှာသွားပြီး ရိုက်သွင်းပါ။ ပြီးတော့ အဲဒီ File\_Name ကို dialog3.java လို့ အမည်ပေးပါ။ အဲဒီ File ကို ရေးသားပြီးတော့ Save လုပ်ထားပုံကို Figure 3.14 အနေနဲ့ လေ့လာကြည့်ပါ။ သတိပြုရမှာကတော့ C:\j2sdk1.4.1\_01\Bin>ဆိုတဲ့ Class path အောက်မှာ save လုပ်ရမှာ နော်။ က Figure 3.14 ကို လေ့လာလိုက်ပါ၍။

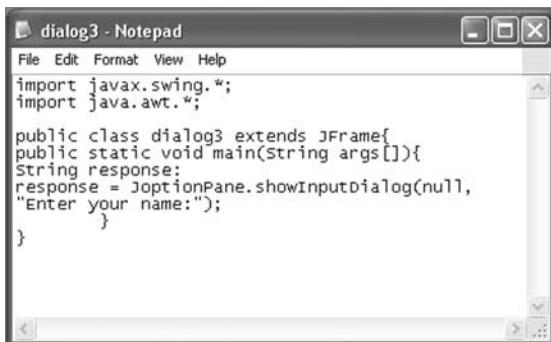


Figure 3.14 Write the sample program code\_line in the Notepad.

က Figure 3.14 က ရေးသားထားတဲ့ code\_line တွေကို လေ့လာပြီးရင်တော့ အဲဒီ File ကို Compile & Run လုပ်ပို့ရာ အကွက် command line windows (DOS) ထဲကိုသွားပါ။ Command Line Windows ထဲကိုရောက်ရင် အောက်ပါ command နဲ့ CLASSPATH'S ကိုပြောင်းပါ။

CD C:\j2sdk1.4.1\_01\Bin ↵

ကဲ့ဒါဆိုရင်တော့ CLASSPATH'S Directory ပြောင်းသွားပါ။ အဲဒီဆိုရင်တော့ အောက်ပါ command line နဲ့ program ကို compile လုပ်ပါ။

java c dialog3.java ↵

အဲဒီလို compile လုပ်ပြီးတော့ Error ကင်းသွားပြီး ဆိုကြပါ၌။ Error ရှိခဲ့ရင်လည်း ရှေ့သင်ခန်းစာတွေမှာလို မျိုး Error

ကို ရှာဖွေပြီးတော့ compile ပြန်လုပ်ပါ။ အဲဒီလိုပြန်ပြီး compile လုပ်လို မရှိတော့ဘူးဆိုရင် အောက်ပါအတိုင်း Run လုပ်ပါ။

java dialog3 ↵

အဲဒီ command ကို အသုံးပြုပြီးတော့ Run လိုက်တယ် ဆိုရင်တော့ dialog3 လို့ အမည်ပေးထားတဲ့ program code\_file ရဲ့ output ဟာ command line windows <DOS> ရဲ့ အလယ်တည့်တည့်မှာ [First argument ဖြစ်တဲ့ null argument ကြောင့်] လာပြီးတော့ Output ကို ဖော်ပြုပေးတာပါပဲ။ အဲဒီ Output ကို လေ့လာကြည့်လိုက်ရင် ကျွန်တော်တို့ ဖန်တီးရေးသားလိုက်တဲ့ /ShowInputDialog() method ကို အသုံးချထားတဲ့ Input Dialog Box ကို တွေ့ဖြင့်ရမှာပါပဲ။ က အဲဒီပုံကို Figure 3.14.1 အနေနဲ့ ရေးသား ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

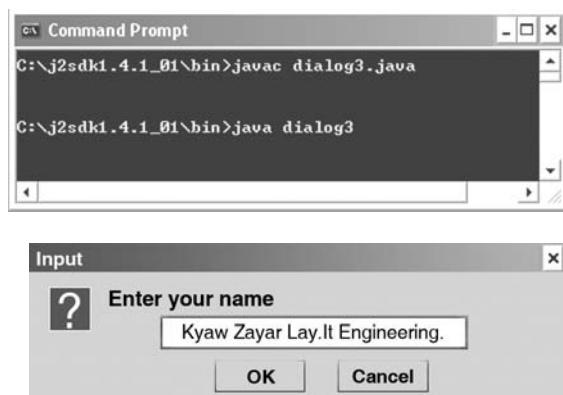


Figure 3.14.1 Compile & Run of the dialog 3.java [ShowInputDialog Box]

က Figure 3.14.1 မှာ ရေးပြထားတဲ့ program output ထွက်လာပုံကို လေ့လာပြီးရင်တော့ Analysis လုပ်ထားပုံနဲ့ နှင့် ယူဉ်ပြီးတော့ ပြန်လုန်လေ့လာသင့်ပါတယ်။ ကဲ့ဒါတော့ ဆက်လက် ပြီးတော့ Index.3.2.8 အနေနဲ့ ရေးသား ဖော်ပြုပေးထားတဲ့ သင်ခန်းစာကို လေ့လာကြရအောင်လား။ 3.2.8 ဟာ showInputDialog() method ကို Icon သင်္ကာတန်ဖြူပြီးတော့ OK\_BUTTON နဲ့ CANCEL\_BUTTON နှစ်ခုကို အသုံးချ ရေးသားထားတာကို လေ့လာကြည့်ကြပါမို့။ 3.2.8 Another way of show Input Dialog () method လို့ အမည်ပေးထားပါတယ်။ ဆက်လက် လေ့လာလိုက်ပါ၍။



### 3.2.8 ANOTHER WAY OF SHOWINPUTDIALOG ( ) METHOD

ShowInput Dialog ( )method နဲ့ ဝတ်သက်ပြီးတော့ program တစ်ပုဒ်ကို ရေးပြုပါပြီ။ ဆက်လက်ပြီးတော့ အခြား နည်းလမ်းတစ်ခုကို အသုံးပြုပါတယ်။ ShowInputDialog ( ) method ကို အသုံးချက်ထဲကြရအောင်လား။ ဒီတော့ ရှုံးသင်ခန်းစာများ ရှင်းပြုခဲ့တဲ့ ShowInputDialog ( ) အကြောင်းကို ပြန်တွေးကြည့်ရအောင်။

```
String response = JOptionPane. ShowInputDialog
                    (null,
                     "What is your ZIP code?",
                     "Enter ZIP Code",
                     JOptionPane.QUESTION_MESSAGE);
```

ဒီတော့ အဲဒီမှာပါတဲ့ arguments (4) ခုဖော်ရှာမှာ မိမိ ထည့်သွင်းလိုတဲ့ အကြောင်းအရာ /String & Icon အစရိတ္တာတွေကိုပါ ထည့်သွင်း ဖန်တီးလိုပါတယ်။ အဲဒီတော့ Listing 3.11 အနေနဲ့ ရေးသားထားတဲ့ program ကို လေ့လာကြည့်ကြရအောင်လား။

**Listing 3.11 Another way of ShowInputDialog ( ) method**

```
1. import javax. swing.*;
2. import java.awt.*;
3.
4. public class dialog4 extends JFrame {
5. public static void main (String args [ ] ) {
6. String response;
7. response = JOptionPane. ShowInputDialog
                    (null,
                     "What is your ZIP code?",
                     "Enter ZIP Code",
                     JOptionPane.QUESTION_MESSAGE);
11. }
12. }
```

#### Analysis

၁။ ဒီ sample program လေးတို့ ကျန်တော်ကတော့ Another way of ShowInputDialog ( ) method လို့ အမည်ပေးထားပါတယ်။ က လိုင်းနံပါတ် (၁) နဲ့ (၂)ကတော့ SWING & AWT Package file နှစ်ခုကို import လုပ် ခေါ်ယူလိုက်ပါတယ်။ လိုင်းနံပါတ်(၃) ကတော့ Blank Line\_Space ချိန်ထားပါတယ်။ လိုင်းနံပါတ် (၄) မှာကတော့ dialog 4

ဆိုတဲ့ primary class ကို ဖန်တီးပါတယ်။ အဲဒီ primary class ကို ဖန်တီးရာမှာ JFrame Class ကို extends လုပ်ဖန်တီးထားပါတယ်။ ပြီးတော့ main ( ) function ကို ရေးသားထားပါတယ်။ ဒီအခြေအနေအထိ ဆိုရင် စာဖတ်သူဟာ အလွယ်တကူ နားလည်သောပေါက်မှာပါ။ ဆက်လက် လေ့လာကြည့်ကြရအောင်။ လိုင်းနံပါတ် (၆) မှာတော့ String data type ကို အသုံးပြုပြီးတော့ response ဆိုပြီး Variable\_Name ကို Declare လုပ်ပါတယ်။ အခု အသုံးပြုမယ့် response ဆိုတဲ့ variable ရှိမှုသာလျှင် Equal လုပ်ပြီးတော့ JOptionPane Class ထဲက ShowInputDialog ( ) method ကို ခေါ်ယူမှုမိုလိုပါပဲ။ ဒီအထိ ရှင်းပြီးနော်။

JII လိုင်းနံပါတ် (၇) မှာတော့ အပေါ်မှာ ကျွန်ုပ်တော် ရှင်းပြုပေးခဲ့တဲ့ Sample code\_line လေးကို ရေးသား ဖော်ပြပေးထားပါတယ်။ ဒီတော့ အကျယ်ချွဲမရှင်းပြတော့ပါဘူး။ Code\_Line အနေနဲ့ပဲ ရှင်းပြပါမယ်။

7. response = JOptionPane. ShowInputDialog (null,
  8. "What is your ZIP Code?",
  9. "Enter ZIP code",
  10. "JOptionPane.QUESTION\_MESSAGE);
- စာဖတ်သူအနေနဲ့ သတိပြုရမှာကတော့ argument code\_line တွေကို ရေးသားရာမှာ comma (,) sign ခံပြီးတော့ ရေးသားလို့ ရပါတယ်။ အမိကအချက်ကတော့ Comma (,) sign နဲ့ Double quote " "နှစ်ခုကြား ဖြစ်ဖိုပါပဲ။ ဒီတော့ လိုင်းနံပါတ် (၁၀)က ရေးသားထားတဲ့ JOptionPane QUESTION\_MESSAGE ကြည့် ဒီ ShowInputDialog ( ) Box မှာ QUESTION သက်တဲ့ Icon လေးပါဝင်လာမှာပါ။ ပြီးတော့ လိုင်းနံပါတ် (၁၁)က "Enter ZIP code" ဆိုတဲ့ arguments\_string ဟာ InputDialog Box ရဲ့အပေါ် Title Bar text အဖြစ် ပါဝင်လာမှာပါ။ ဒီအထိ သေားပေါက်ပြီးနော်။ လိုင်းနံပါတ် (၁) က "What is your ZIP code?" ဆိုတဲ့ arguments\_string ကတော့ InputDialog Box ရဲ့အတွင်း Input TextBox နေရမှာ ပါဝင်လာမယ့် စာသားပါပဲ။ arguments တွေ ဘူးနေရမှာနဲ့သူ အလုပ်လုပ်တာကြောင့် ပေါ်လာတဲ့ Dialog Box ပုံစံလေးကို ရှင်းပြီးနော်။ 1<sup>st</sup> arguments null ကြည့် ခြင်း showInputDialog Dialog Box ဟာ command line windows (DOS) ရဲ့

အလယ်တည့်တည့်မှာလာပြီးတော့ Output ကို ဖော်ပြုပေးမှာပါ။ ကဲဒါဆိုရင်တော့ code\_line တွေကို ရှင်းပြီးနော်။ လိုင်းနံပါတ် (၁၁) ကတော့ main () function ရဲ့ close brace } ပါပဲ။ လိုင်းနံပါတ် (၁၂) ကတော့ primary class dialog 4 ရဲ့ close brace } ပါပဲ။

ဒါတော့ အဲဒီ sample\_program code\_line ကို Notepad ထဲမှာသွားပြီးတော့ ရေးသားလိုက်ပါပြီး။ အဲဒီလို ရေးသားပြီးတော့ system CLASSPATH's C:\j2sdk 1.4.1\_01 \Bin ထမာ သွားရောက်ပြီးတော့ save လုပ်ပါ။ File\_Name ကိုတော့ dialog4လို့ primary Class\_Name အတိုင်းပဲ အမည်ပေးလိုက်ပါ။ အဲဒီလို code\_line တွေ ရေးသားပြီးတော့ save လုပ်ထားပဲကို Figure 3.15 အနေနဲ့ ရေးသားဖော်ပြုပေးထားပါတယ်။ လေ့လာ ကြည့်ပါ။

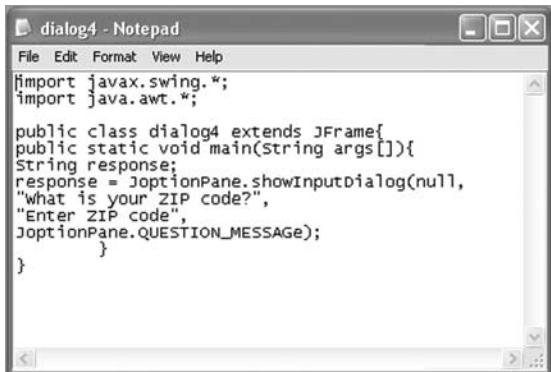


Figure 3.15 Write the sample program code\_line in the Notepad

ဒဲ Figure 3.15 မှာရေးသားထားတဲ့ပဲဟာ NotePad မှာ ဖော်ပြုပေးထားတဲ့ program code\_line တွေပါပဲ။ Code\_Line တွေကို ရေးသားပြီးလို့ save လုပ်ပြီးရင်

Start → Program → Accessories → command line Interface <command-line windows-DOS> ထဲကို သွားပါ။ DOS Windows ပွင့်လာရင်

CD C:\j2sdk 1.4.1\_01 \ Bin ↵

ဆိုပြီးမှ Directory CLASSPATH's ကိုပြောင်းပါ။ ဒီအခါ မှာ prompt သက်တယာ အောက်ပါအတိုင်း ပြောင်းလဲသွားမှာပါ။

C:\ j2sdk 1.4.1\_01\ Bin > ↵

ဒဲ အဲဒီလို prompt ပေါ်နေခိုန်မှာ အောက်ပါ command နဲ့ compile လုပ်ပါ။

Note : JAVA Language အကြောင်းကို ကျွန်ုင်တော် ခဏေက ရှင်းပြုးပါတယ်။ C & C ++ ကို အခြေခံပြီးမှ ဂုဘြား ချက် တူညီမှုတွေကို ရှင်းပြုးပါတယ်။ C & C ++ Language မှာတော့ Ctrl + F9 ကို နိုင်ပြီး Compile (သို့မဟုတ်) Run အစရိတ် Menu တစ်ခုကို ပေါ်ယူ လိုက်ရင် Output .exe file အထိ ထွက်လာတာပါ။ Compile ပြီးတော့ Run အစရိတာတွေကို အဆင့်ဆင့် ပေါ်ယူစရာ မလိုပါဘူး။ VC + + <visual C + > Language မှာကတော့ အရင်ဆုံး Build Menu က compile command\_menu ကို တောင်းရမှာပါ။ Compile လုပ်လို့ error မရှိတော့ဘူးဆိုမှ Build → Make.exe သို့မဟုတ် Make exe အစရိတ် command menu နဲ့ output file ကို .exe အနေနဲ့ တည်ဆောက် ရယူရတာပါ။ C & C ++ Language မှာလို့ output .exe file ဟာ တန်းထွက်မလာပါဘူး။ JAVA Language ကလည်း visual C + + ဆန်ဆန်ပါပဲ။ အခြေခံ code\_line တွေကို ရေးသားပြီးတော့မှ save လုပ်ပြီးရင် javac keyword ကို အသိပြုပြီးတော့ compile လုပ်ရမှာပါ။ error ကုန်းပြီးဆိုမှ Run လုပ်ပြီး Output ကို လေ့လာရတာနော်။

ဒဲ program ကို compile လုပ်ရအောင်လား။

javac dialog4.java ↵

အဲဒီလို compile လုပ်လိုက်လို့ error ရှိခဲ့ရင်လည်း error ကို ရှင်းပြီးတော့ Save As လုပ်ပြီးတော့ compile ပြန်လုပ်ပါ။ ဒဲ error မရှိတော့ဘူးဆိုရင် အောက်ပါ command နဲ့ Run လုပ်ပါ။

java dialog4 ↵

ဒဲ အဲဒီလို Run လိုက်ရင်တော့ ကျွန်ုင်တော်တို့ ရေးသားထားတဲ့ program code-line ကို အဖို့ပြုပြီး ထွက်လာတဲ့ Output program လေးဖြစ်တဲ့ ShowInputDialog Box ကလေးကို DOS Windows ရဲ့ အလယ်တည့်တည့်မှာ တွေ့မြင်ရမှာပါ။ DOS Windows ရဲ့ အလယ်တည့်တည့်မှာ လာပေါ်ရခြင်းရဲ့ အကြောင်းရင်းကတော့ ဘာဖတ်သူ code-line တွေကို Analysis



## Windows XP tips

### ပါဝါဓလုတ်ကို မြှောင်းလဲခြင်း

ကွန်ပူးတာကို တစ်နေကုန် သုံးနေရသူများအတွက် ဓာတ်နားတဲ့ အချိန်များမှာ ဖွင့်လိုက် ပိတ်လိုက်လုပ်ရတာ တကယ်တော့ အချိန်ကုန်ပြီး အလုပ်ရှုပါတယ်။ ပြီးရင် ဓာတ်ပေါ်တွေ ပြန်ဖွင့်ရတာ၊ ဖိုင်တွေ ပြန်ဖွင့်ရတာလည်း အလုပ်ထိတစ်ခုလို ဖြစ်နေတတ်ပါတယ်။ ငှါးအတွက် 'Standby' မှာ ထားနိုင်ပါတယ်။ Start > Settings ကနေ Control Panel သို့ ဝင်ပြီး 'Performance and Maintenance' ကို ဖွင့်ပါ။ အချို့ ဝင်းစိုးစိန်အတွက် 'Performance and Maintenance' မလိုပါ။ ထို့နောက် 'Power Options' ကို ဖွင့်ပါ။ ငါးတွင် 'Advanced' (tab) သို့ ပြောင်းကာ အောက်ဘက်က 'When I press the power button on my computer' နေရာတွင် 'Standby' ကို ရှာဖြိုးရွေးချယ် ကလစ် လုပ်လိုက်ပါ။ 'Standby' နဲ့ ပိတ်ထားပြီး ကွန်ပူးတာကို ပြန်ဖွင့်ရင် လက်ရှိ အလုပ်လုပ်နေတဲ့ ဓာတ်ပေါ်နဲ့ ပိုင်များကို အသုံးပြန်ဖွင့်ရန် မလိုတော့ပါ။ 'Start' ကို ဖွင့်ရာမှာ ပုံမှန် 'Start menu' နဲ့ 'Classic Start menu' ဆိုပြီး စတိတင်လဲ ၂ မျိုး ရွေးချယ်နိုင်ပါတယ်။ ပုံမှန် 'Classic Start menu' မှာ Start > Control Panel သို့ တိုက်ရှိတိုင်နိုင်ပါတယ်။

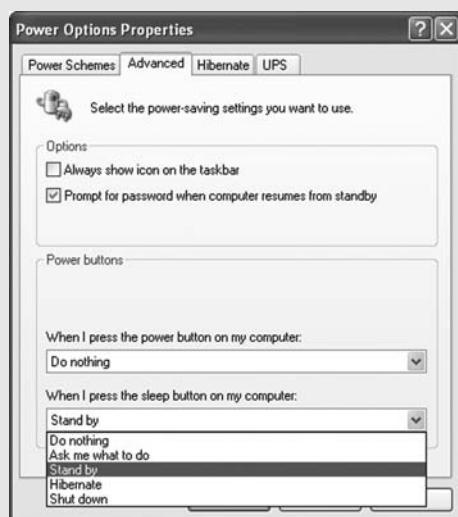


Figure 3.15.1 Compile & Run of the dialog4 . java [showInputDialog Box]

ကဲ Figure 3.15.1 မှာ ရေးသားဖော်ပြထားတဲ့ Figure ကတော့ ShowInput Dialog Box အဖြစ် Output ထွက်လာပုံကို ဖော်ပြပေးထားတာပါပဲ။ ကဲ လေ့လာလိုက်ပါ၌ီး။

ကဲဆက်လက်ပြီးတော့ standard Dialog Box အပ်စုဝင် တွေဖြစ်တဲ့ Show Message Dialog Box အကြောင်းကို 3.2 အနေနဲ့ ဆက်လက် ရေးသား ပို့ချေပေးထားပါတယ်။ ဆက်လက်ပြီး တော့ လေ့လာသင်ယူလိုက်ပါ၌ီး။ သေသေချာချာ ပို့ချေပေးထားပါတယ်။

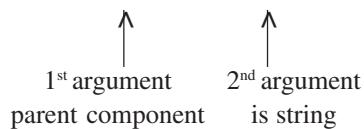
KYAW ZAYAR LAY [ IT ENGINEERING ]



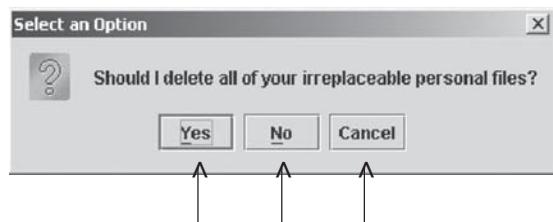


### 3.2.9 MESSAGE DIALOG BOXES

Show Message Dialog (Component, Object);

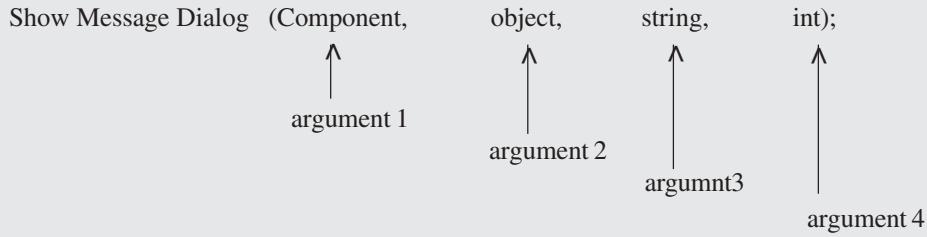


အခြားသော Standard Dialog Boxes အပ်စုဝင်တွေလို ပါပဲ။ 1<sup>st</sup> argument ဟာ parent component ဖြစ်ပြီးတော့ 2<sup>nd</sup> argument ကတော့ Message အဖြစ် အော်ပြေပေးမယ့် String Component ကို ရေးသားလို့ရပါတယ်။ အခြားသော Standard Dialog Boxes တွေနဲ့ မတူညီတဲ့ အချက်ကတော့ Message Dialog Box ဟာ Message ကို ဖော်ပြေပေးအပြီးမှာ return ပြန်ပြီးတော့ value တစ်ခုကို Response လုပ်မလေးနိုင်ပါဘူး။ ရှင်းအောင်ပြောရရင်တော့ Message Box ဟာ OK Button တစ်ခုတည်းသာ ပါဝင်ပါတယ်။ အခြားသော Standard Dialog Box အပ်စုဝင်တွေကတော့ ဥပမာ- Confirm Dialog Box Input Dialog Box အစရိတဲ့ Dialog Box တွေဟာ YES, NO, Cancel အစရိတဲ့ Option Return Value တွေ ရှိပါတယ်။ သုံးစွဲသူ User မဖော်ပြီးတော့မှ ရွေးချယ်လိုက်တဲ့ value ကို return ပြန်ပေးပြီးတော့မှ response ဖြစ်လာမှာပါ။ ဥပမာ- Confirm Dialog Box ဆိုကြပါစိုး။ အောက်မှာဖော်ပြထားတဲ့ Confirm Dialog Box ကို လေ့လာကြည်ပါ။



ဒီ Confirm Dialog Box တေလးမှာဆိုရင် Yes Button, No Button, and Cancel Button တွေ အသီးသီးပါဝင်ပါတယ်။ Yes Button ကိုသာ နှိပ်လိုက်ရင် value ဟာ return ပြန်သွားဖြဲ့ တော့ File တွေကို ဖျက်ပစ်နိုင်ဖို့အတွက် response ပေးမှာပါ။ If တကယ်လို့ NO Button ကိုသာ နှိပ်လိုက်ရင် မဖျက်ဘူးဆိုတဲ့ value ကို response ပေးနိုင်ဖို့အတွက် return ပြန်သွားမှာပါ။ Cancel Button ကိုသာ နှိပ်လိုက်ရင်တော့ select an Option ရဲ့ ဘယ်တန်ဖို့ကို အတည်ပြု မရွေ့ချယ်ပါဘူး ဆိုပြီးမှ value ကို response ပေးမှာပါ။

အခရှင်းပြတဲ့ အချက်တွေဟာ အကြမ်းဖျင်းအားဖြင့် ရင်းပြတာပါ။ တကယ်လို့ ကျွန်တော် အခ ရှင်းပြသင်ကြားနေတဲ့ Message Dialog Box ကိုသာ လေ့လာကြည့်မယ်ဆိုရင် return ပြန်ပြီးတော့ response လုပ်ပေးနိုင်မယ့် value တွေဟာ Message Box မှာ လုံးဝမပါဝင်ပါဘူး။ သာမန်သတိပေးတဲ့ Message ကို ထုတ်ဖော်ပြီး သတိပေးတဲ့အနေနဲ့ပဲ ဖော်ပြပေးနိုင်တာပါ။ နောက် အချက်တစ်ချက်ကလည်း showMessageDialog () method ကို အမြားနည်းလမ်းတစ်ခုအနေနဲ့ ဒေါ်ယူအသုံးချုတဲ့ အဓိမ္မာလည်း ပြီးခဲ့တဲ့သင်ခန်းဘာ 3.2.8 မှာ သင်ကြားပို့ချေပေးခဲ့တဲ့ ShowInputDialog () method နဲ့ Identical အရ အတူတူပဲ ဖြစ်ပါတယ်။ arguments တွေကလည်း အတူတူပဲ ဖြစ်ပါတယ်။ ဥပမာအနေနဲ့ ရေးပြရှင် -



အထက်ပါ Message ကို လေ့လာကြည့်မယ်ဆိုရင် ပြီးခဲ့တဲ့ သင်ခန်းစာမှာ သင်ယူခဲ့တဲ့ ShowInputDialog( ) Method နဲ့ arguments အရ တည်ဆောက်ပဲ အတူတူပါပဲ။ မတူညီတဲ့ အချက်ကတော့ show Message Dialog ( ) Method မှာ

return value ဟာ မပါဝင်ပါဘူး။ အဲဒီ show Message Dialog () method ကို အသုံးပြုပြီးတော့ message Box တစ်ခုကို အောက်ပါအတိုင်း ဖန်တီးလိုပါတယ်။

|                                         |                  |
|-----------------------------------------|------------------|
| JOptionPane .show Message Dialog (null, | ----- argument 1 |
| "An asteriod has destroyed the Earth",  | ----- argument 2 |
| "Asteriod Destruction Alert",           | ----- argument 3 |
| JOptionPane. WARNING_MESSAGE);          | ----- argument 4 |

ShowInputDialog () method တိုန်းကလိုပါပဲ။ arguments အသီးသီးရဲ့နေရာမှာ လိုအပ်တဲ့အချက်တွေကို ထည့်သွင်းပြီးတော့ Message Box ကို ဖန်တီးရယူလို့ ရပါတယ်။

ကဲ ဒါဆိုရင်တော့ Listing 3.12 အနေနဲ့ ရေးသားထားတဲ့ program ကို လေ့လာကြည့်ကြရအောင်လား။

#### Listing 3.12 Demonstrate the show Message Dialog ( ) method

1. import javax.swing.\*;
2. import java.awt.\*;
- 3.
4. public class dialog 5 extends JFrame {
5. public static void main (String args [ ]) {
- 6.
7. JOptionPane. Show Message Dialog (null,
8. "The program has been uninstalled.");
9. }
10. }

#### Analysis

၁။ ဒါ program လေးကို လေ့လာမယ်ဆိုရင် SWING component & AWT component တွေကိုခေါ်ယူပြီးတော့

အသုံးချခိုအတွက် package file နှစ်ခုကို import လုပ်ပါတယ်။ လိုင်းနံပါတ် (၁) နဲ့ (၂) မှာ Declare လုပ်ထားတာ ကို တွေ့ရမှာပါ။ ပြီးတော့ လိုင်းနံပါတ် (၃) မှာမှ dialog 5 ဆိုတဲ့ primary class ကိုရေးဖွဲ့ပါတယ်။ ရေးဖွဲ့ရမှာ JFrame Class ကို အနီးပြုပြီးတော့ extends လုပ်ထားတာနော်။ လိုင်းနံပါတ် (၅) မှာတော့ main () function ကို ရေးဖွဲ့ထားပါတယ်။

JII လိုင်းနံပါတ် (၇)မှာတော့ JOptionPane. Show Message Dialog () method ကို ခေါ်ယူအသုံးချခိုပါတယ်။ အသုံးချရမှာ သာမန် Message Dialog Box ပုံစံနဲ့ ရေးသားထားပါတယ်။

**Show Message Dialog (Component, object);**  
အစဉ်တဲ့ arguments type နဲ့ပဲ ရေးဖွဲ့ထားတာပါပဲ။ ကဲဒီတော့ လေ့လာကြည့်ကြရအောင်လား။

JOptionPane. ShowMessage <----- component  
Dialog (null, <----- argument

"The program has been <----- object  
uninstalled" ); <----- argument

Component argument ແກ້ຽມວາຕොແ ຖະໜີເຕັດິນ: null statement ເຕັດິນ: ອວຍໆ: ປຸ່ຍະ: ປິຕາຍ|| Object argument ແກ້ຽມວາຕොແ "The program has been uninstalled" ຂີ້ຕະ String object ກີ ໄຣ: ພະ: ດາວະ: ປິຕາຍ||

လိုင်းနံပါတ် (၉) ၊ ကတေသာ main () function မှာ close brace} ပါပဲ။ လိုင်းနံပါတ် (၁၀) ၊ ကတေသာ primary class မှာ close brace} ပါပဲ။

Note : Unlike the other dialog boxes, message dialog boxes do not return any kind of response value.

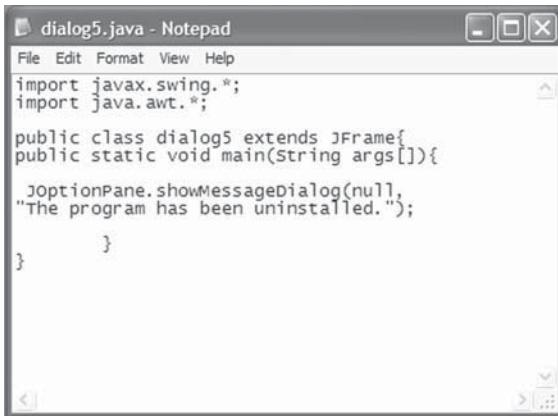


Figure 3.16 Write the program code \_line in the Notepad

ကဲ Figure 3.16 က Notepad ထဲမှာ code\_line တွေ  
ရေးသားထားတာကို လေ့လာပြီးရင်တော့ အဒီ File ကို File\_ Name အနေနဲ့ C:\j2sdk 1.4.1\Bin > ဆိတ် System CLASSPATH'S ထဲမှာ သွားရောက်ပြီးတော့ saveလုပ် သိမ်း  
ဆည်ပါ။ File\_Name ကိုတော့ dialog5.java လို့ အမည်ပေး  
လိုက်ပါ။ ကဲပြီးရင်တော့ command line windows (DOS)  
ထဲကို သွားပြီးတော့ အောက်ပါ commandနဲ့ Directory path  
ကို ဖြောင်းလုပ်ပါ။

CD C:\j2sdk 1.4.1\_01\bin ↴

ဒါနိဂင်တော့ command prompt ဟာ အောက်ပါလုံးစာတိုင်း  
ပြောင်းလဲသွားမှာပါ။

C:\j2sdk 1.4.1\_01\bin>

ကဲ command line ဟာ အောက်ပါ ပုံစံအတိုင်း ပြောင်းလုပ်သူများ ဖြစ်ပါစိန့်။ အောက်ပါ command နဲ့ program ကို compile လုပ်ပါ။

```
javac dialog5.java ↴
```

အေဒီအခါ program ကို compile လုပ်လို့ error မရှိတော့ဘူး  
ဆိုရင် အောက်ပါ command နဲ့ ဆက်လက်ပြီးတော့ program  
ကို Run လွယ်ပါ။

java dialog 5 ↴

အဲဒီလို program ကို Run လုပ်လိုက်ရင်တော့ java compiler ကနေဖြူးတော့ Figure 3.16.1 မှာမြင်ရတဲ့ Message Dialog Box ပုံစံလေးတို့ ဖြင့်ရပါလိမ့်မယ်။ Compile & Run လုပ်နေ့နှင့်ကို ဖော်ပြထားတဲ့ Figure 3.16.1 ကို လေ့လာကြည့်လိုက်ပါ၌။ စာဖတ်သူအနေနဲ့ ရီးရှင်းလှတဲ့ code\_line ရဲ့ စွမ်းဆောင်ချက်ကို တွေ့မြင်ရမှာပါ။

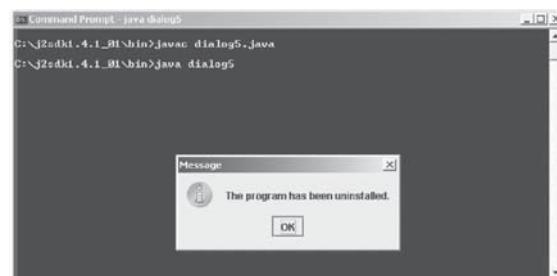


Figure 3.16.1 Compile & Run of the dialog  
5. java [Show Message Dialog ( ) method]

အချက်ပြထားတဲ့ Compile & Run type ကို လေ့လာ  
ကြည့်မယ်ဆိုရင် Message Dialog ဘာ command line windows မှာလာပြီး ဖော်ပြန်မှာပါ။ သတိပြုမယ်ဆိုရင် OK Button တစ်ခုသာ ပါဝင်တာကို တွေ့ရမှာပါ။ ဒါဟာ အပေါ်မှာ ကျွန်ုတ်တော်ရှင်းပြနေတဲ့ return ဖုန်းစိုး response value မရှိပါဘူး။ လက်ခံဖို့ ဘယ်လို Message ပဲဖြစ်ဖြစ် OK Button တစ်ခုသာ ပါဝင်ပါတယ်။

Note : Visual Basic Programming Language ကို အသုံးပြုတဲ့ programmer တွေအနှစ်ကတော့ အောက်ပါ method ကို အသုံးချရတာပါလဲ။ MsgBox ( ) method ပါ။ အလွယ်ဆုံးပြောရင် Command Button ကို Form\_ Windows ပေါ်မှာ နေရာ ချိပြုးတော့ အဲဒီ command Button ရဲ့ code\_windows ထဲ ရောက်အောင် D-

Click ခေါက်ပြီးတော့ဝင်ပါ။ ပြီးတော့ Message Box ကို အသုံးပြုဖို့အတွက် –

**MsgBox (Message String , ButtonsAnd Icon, Title String)**

အစရိတဲ့ Method ကိုသာ အသုံးချလိုက်ရင် အလွယ် တကူနဲ့ Message Box တစ်ခုကို ဖန်တီးနိုင်မှာပါ။ JAVA Language မှာကတော့ Visual Basic Language မှာလောက် မလွယ်ကူပါဘူး။ သတ်မှတ်ထားတဲ့

container class ကို အမိုပြုပြီးမှ ရေးဖွဲ့ရတာပါ။ ဒီ အချက်ဟာ Language နှစ်ခုကွာဟာချက်ကို အလွယ်ကူ ဆုံး နှင့်ဗုံးဖြတ်ပါပဲ။ Visual Basic Language မှာလောက်တော့ JAVA ဟာ မလွယ်ကူပါဘူး။

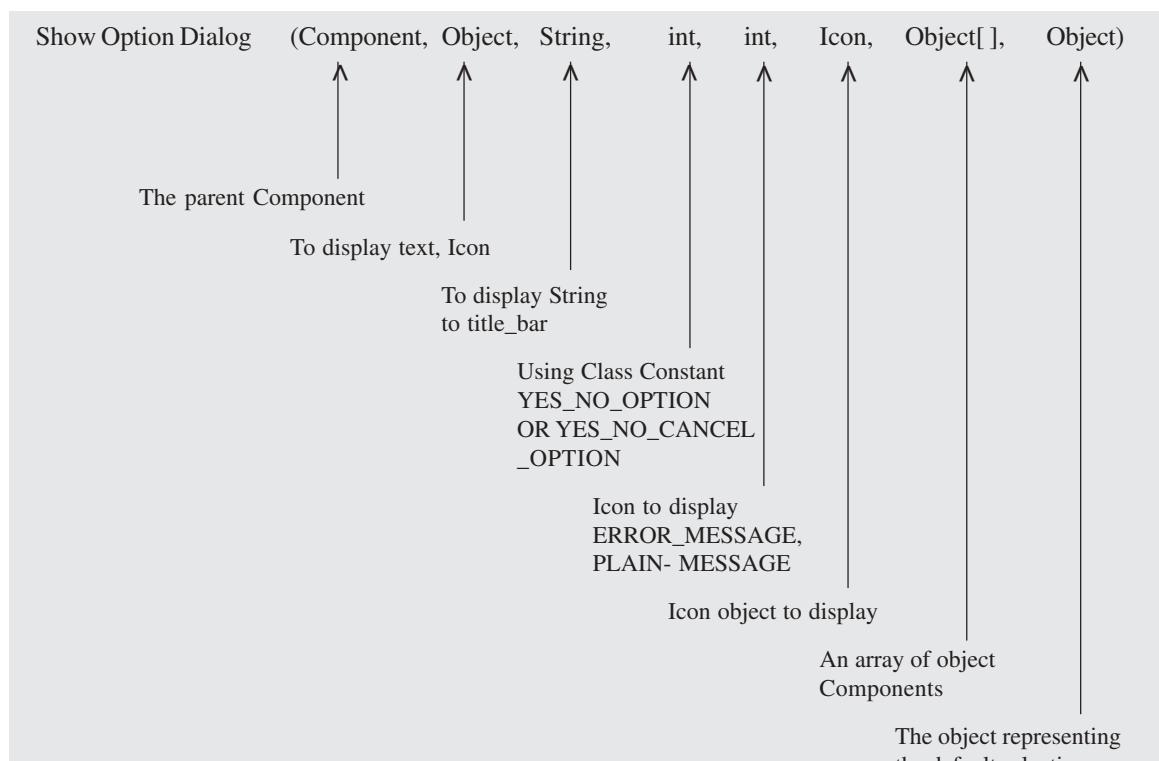
ကဲဆက် လက်ပြီးတော့ Standard Dailog Boxes အပ်စုဝင်တွေဖြစ်တဲ့ Option Dialog Boxes တွေအကြောင်းကို လေ့လာကြည့်ကြရအောင်လား။ Sub- Index 3.2. 10 အနော့။ ရေးပြေားထားပါတယ်။ ဆက်လက် လေ့လာလုံးကိုပါ။

### 3.2.10 | OPTION DIALOG BOXES

အချက်တွေအတွက်အမြတ်အမြတ် Dialog Boxes အပ်စုဝင်တွေ ထဲမှာ option Dialog Box ဟာလည်း အပ်စုဝင်တစ်ခုအနေနဲ့ ပါဝင်ပါတယ်။ Option Dialog Boxes ရဲ့ features ကတော့ အရှေ့မှာ သင်ကြား ပို့ချပေးခဲ့တဲ့ အခြားသော Dialog Boxes တွေရဲ့ features တွေ အားလုံးကို ပေါင်းစပ်ထားတာဆိုရင်တော့ မမှားပါဘူး။ သူကို ဖန်တီး ခေါ်ယူရေးသားရာမှာတော့ ပြီးခဲ့တဲ့ သင်ခန်းစာတွေမှာ မသင်ကြားပေးရသေးတဲ့ Show Option

Dialog () method ကို အသုံးချရမှာပါပဲ။ ကဲ လေ့လာကြည့်ကြရအောင်လား။

Show Option Dialog (Component, Object, String, int, int, Icon, Object [ ], Object) အထက်ပါ method မှာ arguments (or) components စုစုပေါင်း (၈) ခုတိတိ ပါဝင်ပါတယ်။ ဒီတော့ အဲဒီ arguments တွေကို သေသေချာချာ လေ့လာကြည့်ကြရအောင်လား။



အပေါ်မှာ ရေးပြုခဲ့တဲ့ ပုံစံကတော့ show Option Dialog () method ကို အလွယ်ကူဆုံးဖြစ်အောင် ရေးပြုခဲ့တဲ့ပုံစံပါပဲ။ ဒါပေမယ့် ပိုပြီး လွယ်ကူအောင် arguments တစ်ခုချင်းစီကို

သေသေချာချာ ရှင်းပြေားပါမယ်။ Exam တွေ ဖြေရာမှာ အထောက် အကူရစေနိုင်ဖို့အတွက် မူရင်း Guid Book ထဲကအတိုင်း ရေးသား ဖော်ပြေားပါမယ်။ Simple English ဖြစ်တဲ့အတွက်ကြောင့်



စာဖတ်သူတွေအနေနဲ့ အလွယ်တကူ နှားလည်သဘောပါက်နိုင်ပါ လိမ့်မယ်။ ကဲ လေ့လာကြည့်လိုက်ပါပြီး။

- The parent components of the dialog
- The text, icon, or component to display
- A String to display in the title bar
- The type of box, using the class constants YES\_NO\_OPTION or YES\_NO\_CANCEL\_OPTION, or the literal 0 (Zero) if other buttons will be used instead
- The icon to display, using the class constants ERROR-MESSAGE, INFORMATION-MESSAGE, PLAIN-MESSAGE, QUESTION-MESSAGE, or WARNING-MESSAGE, or the literal 0 (Zero) if none of them should be used
- An Icon object to display instead of one of the icons in the preceding arguments
- An array of objects holding the components or other objects that represent the choices in the dialog box, if YES\_NO\_OPTION and YES\_NO\_CANCEL\_OPTION are not being used
- The object representing the default selection if YES\_NO\_OPTION and YES\_NO\_CANCEL option are not being used

မှုလ Guid Book ထဲကအတိုင်း ရေးသားပေးရတာကတော် certification Practice အနေနဲ့ အလေ့အကျင့်ရအောင်ပါပဲ။ ဒါမှ သာလျှင် အတွေအကြံရှိမှာ ဖြစ်မှာပါပဲ။

ကဲ Listing 3.13 အနေနဲ့ ရေးသားထားတဲ့ code-line တွေကို အရင်ဆုံး လေ့လာကြည့်ကြရအောင်လား။

Listing 3.13 Demonstrate the opting Dialog Boxes

```
1. import javax.swing.*;
2. import java.awt.*;
3.
4. public class dialog6 extends JFrame{
5. public static void main (String args [ ]) {
6.
7. JButton [ ] g = new JButton [3];
8. g [0] = new JButton ("Male");
9. g [1] = new JButton ("Female");
10. g [2] = new JButton ("Non of Your Business");
11. int responce = JOptionPane. ShowOption Dialog
(null, // open bracket for
    // showOption Dialog method
12. "What is your gender?" ;
13. "Gender",
14. O ,
```

```
15. JOptionPane.INFORMATION_MESSAGE,
16. null,
17. g,
18. g [2] ; // close bracket for show option Dialog
method
19. }
20. }
```

Listing 3.13 အနေနဲ့ ရေးသားထားတဲ့ code-line စုစုပေါင်း <Line-spacing ကျို့တာအပါအဝင်> လိုင်း ဂုဏ် တိတိရှိပါတယ်။ Show Option Dialog () method မှာ arguments parameter အားလုံး (၈) ခုတိတိ ပါဝင်ပါတယ်။ အဲဒီ parameter arguments တွေကို ရှင်းရင်းလင်းလင်း သိရှိရအောင်လို့ လိုင်းနဲ့ပါတ်တွေခြဲ့ပြီး ရေးသားထားပါပဲ။ Note Pad ထဲမှာ code\_line တွေ ရေးသားရင်လည်း ဒီပုံစံအတိုင်းပဲ ရေးသားရင် ရပါတယ်။ အတူတူပါပဲ။

ကဲ အဲဒီ code\_line တွေကို Notepad Program ထဲမှာ သွားရောက်ပြီးတော့ JAVA\_code တွေကို ရေးသားရှိက်သွင်းပါ။ အဲဒီလို JAVA\_code တွေ ရေးသားရပုံတွေကို Figure 3.17 အနေနဲ့ ရေးသားဖော်ပြပေးထားပါတယ်။ အဲဒီ Figure 3.17 ကို လေ့လာပြီးတော့ အဲဒီအတိုင်း code တွေကို ရေးသားထားပါ။

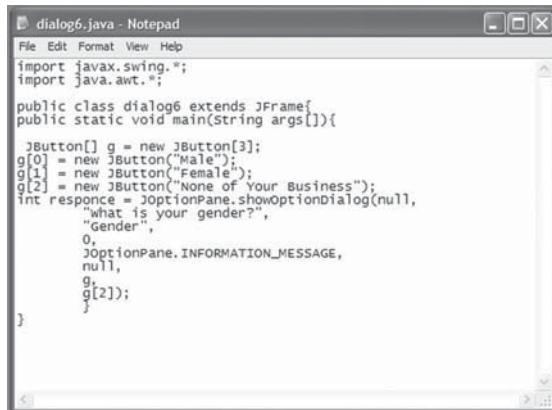


Figure 3.17 write the program code-line In the Notepad

၁။ ကဲ code\_line တွေကို Analysis လုပ်ပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။ လိုင်းနဲ့ပါတ် (၁) နဲ့ (၂) မှာ AWT & SWING components တွေကို ခေါ်ယူအသုံးချခို့အတွက် package File နှစ်ခုကို importလုပ် ခေါ်ယူထားပါတယ်။ ပြီးတော့ လိုင်းနဲ့ပါတ် (၃) မှာ dialog 6 ဆိတဲ့ primary class ကို ရေးသား ဖန်တီးပါတယ်။ အဲဒီ primary class ထဲမှာ main () function ကို ရေးသားထားပါတယ်။ အဲဒီ main () function ထဲမှာမှ JButton



Class ကို object အသင်းပေါင်းယူအသုံးချပါတယ်။ JButton Class ဟာ JFrame Class ထဲမှာ ပါဝင်ပါတယ်။ အဲဒါကို အသုံးချဖိုင်းဖို့အတွက် dialog 6 ဆိုတဲ့ primary class ကို ရေးသားဖန်တီးစဉ်ကတည်းက extends လုပ်ထားတာပါပဲ။  
J<sup>II</sup> JButton Class ကို object\_Name 9 အဖြစ် ရေးသားဖန်တီးစဉ်တုန်းက Array\_Type အနေနဲ့ ရေးသားခဲ့ပါတယ်။ Array <OR> Demisional Array\_Type နဲ့ ရေးသားခဲ့တာပါပဲ။ အဲဒါကို စာဖတ်သူအနေနဲ့ သတိပြုမိမယ်။

Note : Array ပတ်သက်တဲ့ သင်ခန်းစာတွေကို Chapter [1] & [2] မှာ ရေးသား ပို့ခြုံပေးခဲ့ပြီးသားပါ။ သတ်ထားပြီး မလေ့လာခဲ့ရသေးရင် လာမယ့် ရွှေ့သင်ခန်းစာတွေမှာ ထပ်မံ ပို့ချရင်းလင်းပေးပါ၌မယ်။

Demisional -Array အနေနဲ့ ရေးသားပြီးတော့  
g [ 0 ] = new JButton ("Male");  
g [ 1 ] = new JButton ("Female");  
g [ 2 ] = new JButton ("None of Your Business");  
ဆိုပြီးမှ Array\_Type ကို (၃) ခဲ့ပြီး ရေးသားထားပါတယ်။ Demisional Array ရဲ့ စုတိ Initialize ကိန်းဟာ [ 0 ] က စပါတယ်။ ဒီတော့ Array Initialize လုပ်ထားတဲ့ g [ ] ကို အသုံးပြုပြီးတော့ Option Dialog Box မှာ Button အဖြစ် ထည့်သွင်းရေးသားမှာပါ။ ဒီအချက်တွေ သေသေချာချာ ဖတ်ပြီးတော့ လေ့လာကြည့်ပါ။

ကဲ လိုင်းနံပါတ် (၁၁)မှာတော့ response ဆိုတဲ့ Variables\_Name ကို Declare လုပ်ပြီးတော့မှ JOption Pane. show Option Dialog () method ကို ခေါ်ယူအသုံးချပါတယ်။ ဒီအခါ မှာ ပါဝင် ထည့်သွင်းရေးသားရမယ့် arguments တွေကို ဖြည့်သွင်းပေးရနိုင်အတွက် လိုင်းနံပါတ် (၁၁) ကနေပြီးတော့ လိုင်းနံပါတ် (၁၈) အထိ အဆင့်ဆင့် code\_line တွေကို ခွဲပြီး ရေးသားပေးထားပါတယ်။ ရေးသားရာမှာလည်း arguments-type တွေ ကို သူ့၏ type အလိုက် ထည့်သွင်းထားတာမို့ အထွေအထူး ထပ်မံရှင်းပြေစရာမလိုတော့ပါဘူး။ ပို့ပြီး မျက်စိတ်ရှင်းသွားရမယ့်လို့ ဒီသင်ခန်းစာရဲ့ ရှေ့ခံ့မှာ ရေးသားရှင်းပြေခဲ့တဲ့ Show Option Dialog () method ရဲ့ Analysis ကိုသာ ပြန်ပြီးဖတ်ပြီးမှ လေ့လာသင့်ပါတယ်။ ဒါဆိုရင် Arguments\_Type တွေကို နားလည့်သဘောပါက်သွားမှာပါ။

လိုင်းနံပါတ် (၁၉) ကတော့ main () function ရဲ့ close brace } ပါပဲ။ လိုင်းနံပါတ် (၂၀) မှာတော့ primary\_class dialog 6 ဆိုတဲ့ class ရဲ့ close brace } ပဲ ဖြစ်ပါတယ်။

ကဲဒီတော့ program code\_line တွေကို ရေးသားပြီးတော့ save လုပ်ကြရအောင်လား။ ဒီတော့ Notepad ထဲမှာ code-

line တွေ ကို ရေးသားပြီးရင် File ကနေ Save လုပ်ပါ။ Save လုပ်ရာမှာ File\_path လမ်းကြောင်းကို C:\j2sdk 1.4.1\_01\Bin ဆိုတဲ့ directory အောက်မှာ save လုပ်ပါ။ Save လုပ်ရမယ့် File\_Name ကတော့ dialog 6.java ပဲ ဖြစ်ပါတယ်။

Note : Save လုပ်ပုံ / Compile & Run လုပ်ပုံတွေကို အရှေ့သင်ခန်းစာမှာလည်း ထပ်ကြောင်းတလဲ ရေးသားပေးခဲ့ပြီးသားပါ။ အဲက ရည်ရွယ်ချက်ကတော့ JAVA သင်ခန်းစာကို စိတ်ဝင်စားတဲ့ ဘယ်သူမဆို အချက်ရှိတဲ့ အချက်ရှိတဲ့ Chapter-Index နေရာကနေ ကြားဖြတ်ပြီးလေ့လာသင်ယူယူရင် အလွယ်တကူ လေ့လာလို့ ရနိုင်အောင် ရေးသားပေးရတာပဲ ဖြစ်ပါတယ်။

က C:\j2sdk 1.4.1\_01\Bin ဆိုတဲ့ CLASS\_PATH'S ထဲမှာ save လုပ်ပြီးရင် compile & Run လုပ်နိုင်ပုံအတွက် Start → Program → Accessories → command prompt ထဲကို သွားပါ။ Command Line Windows <DOS> ပွင့်လာရင် Save လုပ်ထားတဲ့ Directory ရှိရာကို သွားဖို့အတွက် အောက်ပါ command -line ကို ရိုက်သွင်းပါ။

CD C:\j2sdk 1.4.1\_01\Bin ↵

အဲဒီအခါကျရင် prompt ဟာ C :\j2sdk 1.4.1\_01\Bin > ဆိုပြီး ပေါ်လာမှာပါ။ ဒီအခါမှာ Save လုပ်ခဲ့တဲ့ dialog6.java ဆိုတဲ့ code\_File ကို အောက်ပါ command လုပ်ပါ။

javac dialog6.java ↵

Compile လုပ်လိုက်လို့ Error တက်လာရင်လည်း Line\_Number ကို မှတ်ထားပြီးတော့ Error ကို ဖြင့်ပါ။ ပြီးရင် Save As လုပ်ပြီးတော့ DOS ထဲမှာ ထပ်မံပြုပြီးတော့ compile ပြန်လုပ်ပါ။ အကယ်၍ Error မရှိတော့ဘူးဆိုရင် အောက်ပါ command နဲ့ ထပ်မံပြုပြီးတော့ program ကို Run လုပ်ပါ။

Note : C & C + + program မှာလိုပျိုး Run လိုက်တာနဲ့ Compile & Build ဖြစ်သွားတာပျိုး မဟုတ်ဘူးဆိုတာကို အဲဒီ နယ်ပယ်ကလာတဲ့ programmer တွေအနဲ့ အထူးသတိပြုပါ။ code\_line ရေးသားပုံကတော့ သိပ်မကွာ မြားပါဘူး။ NotePad ထဲမှာပဲ C & C + + program code တွေ ရေးသားပြီးတော့ file extension ကို .CPP / .C အစရှိဖြင့် ထည့်သွင်းပြီး Save လုပ် C & C + + code file အဖြစ် windows က သတ်မှတ်ပေးတာကို အချို့သော programmer တွေ အသုံးပြုဖူးပါလိမ့်မယ်။ အဖြစ် Theory ကျတော့လည်း သိပ်မကွာမြားပါဘူး။



Run & Compile လုပ်ပုံမှာကတော့ တော်တော်လေး  
ကွဲခြားပါတယ်။ အဲဒီအချက်ကို သတိပြုကြည့်ပါ။  
ကဲ ဒီတော့ ဆက်လက်ပြီးတော့ compile လုပ်ပြီးတော့  
Error မရှိရင် program ကို Run လုပ်ကြည့်ကြရအောင်လား။  
အောက်ပါ command ကို အသုံးပြုပါ။

java dialog 6 ←

အဲဒီ command ကို အသုံးပြုပြီးတော့ Run လိုက်ရင် အပေါ်  
မှာ ရေးသား သင်ကြားပို့ချပ်ပေးခဲ့တဲ့ dialog 6.java ရဲ့ Output  
Windows ကို တွေ့ဖြင့်ရမှာပါ။ ဒီတော့ programs ကို com-  
pile & Run လုပ်ပဲ၏ output ထွက်လာပဲကို Figure 3.17.1  
အနေနဲ့ ရေးသားပြီးတော့ ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်  
ပါ။



Figure 3.17.1 Compile & Run of the dialog 6.java

ကဲ Figure 3. 17.1 မှာ ရေးပြထားတာကတော့ dialog 6.java ဆိုတဲ့ program ကို compile & Run လုပ်လိုက်တဲ့  
အတွက် ရှုံးလာတဲ့ Output ပါပဲ။ ဒီတော့ Standard Dialog  
Box တွေ အကြောင်းကို လေ့လာလိုအပြီးမှာ ဆက်လက်ပြီးတော့  
3.3 Working with Swing Components ဆိုတဲ့ သင်ခန်းစာကို  
ဆက်လက် လေ့လာကြည့်ကြရအောင်လား။ အလွန်အသုံးဝင်တဲ့  
Components Tools တွေကို အသုံးပြုတဲ့ Method တွေကို  
ရေးသားပို့ချပ်ပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

Note : စာဖတ်သူဟာ သင်ခန်းစာတဲ့ Code\_File တွေကို  
အသုံးပြုပြီး Program တည်ဆောက်ရာမှာ Error  
တက်တာကြောင့် အက်အခဲဖြစ်နေတယ်ဆိုရင် ကျွန်ုတ်  
အနေနဲ့ code\_line ရေးပြီးသား File လေးတွေကို CD  
ရွေ့နဲ့အတူ ကွန်ပျူတာဂျာနယ်ကို ပေးပို့ပေးထားပါတယ်။  
အလွယ်တကူ ကူးယူပြီး မိမိရဲ့ကွန်ပျူတာထဲမှာ compile  
& Run လုပ်လိုရနိုင်မှာပါ။ ဒါပေမယ့် အခု လက်ရှိ  
chapter [3] နဲ့ လာမယ့်သင်ခန်းစာ program အခါးပဲ  
ပါဝင်ပါတယ်။ ပြီးခဲ့တဲ့ သင်ခန်းစာအဟောင်းက pro-

gram တွေကိုတော့ အားလုံးစုစုည်းပြီး ပေးပို့ပေးထား  
ပါတယ်။ အရေးကြီးတာကတော့ စာဖတ်သူဟာ Microsoft  
Visual Studio. Net ကို Install လုပ်ထားပါ။ ဒါကို C:\j2sdk  
1.4.1\_01\Bin ထဲကို ကူးထည့်လိုက်ရင် အောက်မှာ  
ဖော်ပြထားတဲ့ Figure 3.17.2 က ပုံအတိုင်း Icon  
သင်္ကာပြနောပါ။ D-Click ခေါက်ပြီးတော့ ဖွင့်လိုက်  
ရင်လည်း Microsoft Development Environment [design] ဆိုပြီးမှ ဖွင့်လာမှာပါ။ Compile & Run  
လုပ်ရမှာ ထုံးအတိုင်း command prompt ထဲသွားပြီး  
Compile & Run လုပ်ရှိပါပဲ။



Figure 3. 17.2 The JAVA code file is showing with the Visual J# language (Microsoft Visual Studio .Net Framework)

အကယ်၍ စာဖတ်သူဟာ ထုံးအတိုင်း Notepad နဲ့ပဲ  
ဖတ်ကြည့်ချင်တယ်ဆိုရင်တော့ File \_Icon ကို R-Click ခေါက်ပြီး  
တော့ open with → Notepad ကိုရေးပါ။ ဒါဆိုရင်တော့  
Notepad Program နဲ့ပဲ ဖွင့်လာမှာပါ။ ဘယ်လိုပဲ ဖွင့်လာဖွင့်လာ  
Compile & Run လုပ်တာကတော့ အတူတူပါပဲ။ ဒီတော့  
စာဖတ်သူအနေနဲ့ အခုလက်ရှိ Standard Dialog Box နဲ့  
ပတ်သက်တဲ့ program တွေ အပါအဝင် ရှေ့လာမယ့် Chapter  
[3] ရဲ့ Sub- Index က program တွေကိုပါ ကူးယူ လေ့လာပြီး  
တော့ Run လုပ်လိုပါတယ်။ ဒီတော့ 3.3 Working with  
Swing Components ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် လေ့လာ  
ကြည့်ပါ။ Code\_File CD နဲ့ ဆိုရင်တော့ သင်ခန်းစာရဲ့  
Analysis ကို လေ့လာပြီးတော့ Compile & Run လုပ်မယ်ဆိုရင်  
ပါပြီး အဆင်ပြုပါလိမ့်မယ်။ လေ့လာကြည့်ပါ။



## WORKING WITH THE SWING COMPONENTS

### 3.3 USING THE JRadio BUTTON IN JAVA

JRadio Button နဲ့ ပတ်သက်ပြီးတော့ အရှင်သင်ခန်းစာတွေ မှာ ကျွန်တော် ရှင်းပြပေးခဲ့ဖူးပါတယ်။ Visual Basic Language မှာတော့ Option Button လို့ ခေါ်ဝေါ်ပါတယ်။ ဘူး၏ သဘော တရားကတော့ ရွေးချယ်လိုက်တဲ့ JRadio Button ကို True အဖြစ် တန်ဖိုးသတ်မှတ်ပေးပြီးတော့ မရွေးချယ်ထားတဲ့ JRadio Button ကို တန်ဖိုး False အဖြစ် သတ်မှတ်ပေးထားပါတယ်။ တစ်ကြိမ်မှာ JRadio Button တစ်ခုကိုသာ select လုပ်နိုင်ပါတယ်။ ဒါဟာ အကြမ်းဖျင်းအားဖြင့် ရှင်းပြတာနော်။ JRadio Button အကြောင်း ကို လက်ထွေ နားလည့် သဘောပါက်သွားအောင် Listing 3.14 အနေနဲ့ ရေးပြထားတဲ့ code\_line တွေကို အရင်ဆုံး လေ့လာ ကြည့်လိုက်ပါဉီး။ NotePad Program မှာ ရေးသားထားတာကို အရင်ဖော်ပြပေးထားပါတယ်။ Figure 3.18 အနေနဲ့ဖြေပါပဲ။

```
radio.java - Notepad
File Edit Format View Help
Import javax.swing.*;
Import java.awt.*;
public class radio extends JFrame{
JRadioButton[] rb = new JRadioButton[7];
public radio(){
super("Choose Your's Name");
setSize(150,250);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
rb[0] = new JRadioButton("Kyaw_Zayar_Lay",true);
rb[1] = new JRadioButton("Ma_Zar_Zar_Oo");
rb[2] = new JRadioButton("Kyaw_Myo_Htet");
rb[3] = new JRadioButton("Ko_Ko");
rb[4] = new JRadioButton("Ma_Woe_Thiba");
rb[5] = new JRadioButton("Ma_Wit_Vi");
rb[6] = new JRadioButton("Ma_Zar_Chin");
JPanel pane = new JPanel();
ButtonGroup b = new ButtonGroup();
for(int i=0;i<rb.length;i++){
b.add(rb[i]);
pane.add(rb[i]);
}
setContentPane(pane);
show();
}
public static void main(String[] arguments){
radio r = new radio();
}
}
```

Figure 3.18 Write the program code\_line  
In the Notepad

အခါ code\_line တွေ ရေးသားထားတာကတော့ NotePad program ထဲမှာ code\_line တွေ ရေးသားပြထားတာပါပဲ။ File\_Name ကိုတော့ radio.java လို့ အမည်ပေးထားပါတယ်။ အဲဒုံး Code\_File ကို C:\j2sdk 1.4.1\_01\Bin > ဆိတ္တာ directory အောက်မှာ သွားရောက်ပြီးတော့ save လုပ်ပါ။ ဒါဆိုရင်တော့ JAVA Radio Button တစ်ခုကို ဖန်တီးရယူဖို့ အတွက် code\_line တွေကို ရေးသားဖန်တီးပြီးပြီးဆိုကြပါစိုး။ Program ကို Compile & Run မလုပ်ခင်မှာ အရင်ဆုံး Analysis လုပ်ပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။

ကဲဒီတော့ Line No: တွေ ပါဝင်ပြီး ထည့်သွင်း ရေးသားထားတဲ့ Listing 3.14 ကို အရင်ဆုံး လေ့လာဖတ်မျှကြည့်လိုက် ပါဉီး။ Listing 3.14 Demonstrate the JAVA Radio Button လို့ အမည်ပေးထားပါတယ်။ လေ့လာကြည့်ပါ။ Code\_Line အရေအတွက် စုစုပေါင်း (၃၃) ကြောင်း ရှိပါတယ်။ NotePad ထဲမှာရေးရင်တော့ Line\_No : တွေ ထည့်ရေးစရာ မလိုပါဘူး။ လေ့လာကြည့်ပါ။

#### Listing 3.14 Demonstrate the JAVA Radio Button

1. import javax.swing.\*;
2. import java.awt.\* ;
- 3.
4. public class radio extends JFrame {
5. JRadioButton[] rb = new JRadioButton[7] ;
- 6.
7. public radio () {
8. super ("Choose Your's Name");
9. setSize (400,150);
10. set Default Close Operation (JFrame. EXIT\_ON\_CLOSE);



```

11. rb [0] = new JRadio Button ("Kyaw Zayar Lay",
   true);
12. rb [1] = new JRadio Button ("Kyaw_Myo_Htet");
13. rb [2] = new JRadio Button ("Ko_Bo_Min");
14. rb [3] = new JRadio Button ("Ma_Zar_Zar_Oo");
15. rb [4] = new JRadio Button ("Ma_Moe_Thida");
16. rb [5] = new JRadio Button ("Ma_Wit_Yi");
17. rb [6] = new JRadio Button ("Ma_Zar_Chi");
18. JPanel pane = new JPanel ();
19. Button Group b = new Button Group ();
20.
21. for (int i = 0; i<rb. Length; i + +) {
22. b.add (rb [i]);
23. pane.add (rb [i]);
24. }
25.
26. set Content Pane (pane);
27. show ();
28. }
29.
30. public static void main (String [ ] arguments) {
31. radio r = new radio ();
32. }
33.
    
```

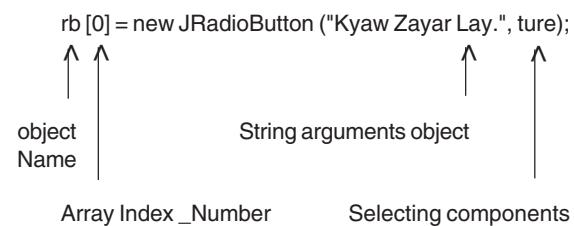
ကဲ code\_line တွေကို line No: နဲ့တက္က တွေရဖြစ်ထိတော့  
Analysis လုပ်ကြည့်ကြရအောင်လား။

#### Analysis

၁။ လိုင်းနံပါတ် (၁)နဲ့ (၂) မှာတော့ ထုံးခံအတိုင်း awt & Swing components တွေကို ခေါ်ယူ အသုံးချင့်ဖို့အတွက် package File နှစ်ခုကို import လုပ်ပြီး ခေါ်ယူထားပါတယ်။ အဲဒီနောက်မှာမှ radio ဆိုတဲ့ primary class ကို ဖန်တီးပါတယ်။ လိုင်းနံပါတ် (၅) မှာ ရေးသားထားတဲ့

JRadio Button [ ] rb = new JRadio Button [7]; ဆိုပြီး ရေးသားထားတာကို တွေ့ရမှာပါ။ အဲဒီ code\_line ဟာ Array Type [ ] တစ်နည်းပြောရှင်တော့ Demisional Array အနေနဲ့ JRadio Button ကို ဖန်တီးပါတယ်။ တစ်နည်း ပြောရင်တော့ JRadio Button မှန်သမျှကို ခေါ်ယူတိုင်း object\_Name rb အနေနဲ့ပဲ ခေါ်ယူရပါတယ်။ ခေါ်ယူတဲ့အခါမှာ Array Type အဖြစ်ဖန်တီးတဲ့ square bracket [ ] ထဲမှာပါဝင်တဲ့ index\_Number ပဲပြောင်းလဲ သွားမှာပါ။ အရေအတွက်က [7] ခဲ့တော့ Index\_Number အနေနဲ့ကတော့ Index [0] to Index [6] အထိပေါ်နော်။ ဒီတော့ စာဖတ်သူ မျက်စိတဲ့မှာ ရှင်းသွား အောင် constructor ( ) function ထဲကိုသွားပြီး လေ့လာကြည့်ကြရအောင်လား။

၂။ ကဲ လိုင်းနံပါတ် (၁၁) မှာ ရေးသားထားတဲ့ code\_line ကို လေ့လာကြည့်ကြရအောင်။



ဒါဆိုရင်တော့ Array ရဲ့ Index \_Number တွေ သွားပဲကို ရှင်းသွားပြီနော်။ အဲဒီပုံစံအတိုင်းပဲ Array Type တွေနဲ့ အောက်ပါ Code Line တွေကို ရေးသားထားတာပါပဲ။

rb [1] = new JRadio Button ("Kyaw\_Myo\_Htet");  
rb [2] = new JRadio Button ("Ko\_Bo\_Min");  
rb [3] = new JRadio Button ("Ma\_Zar\_Zar\_Oo");  
rb [4] = new JRadio Button ("Ma\_Moe\_Thida");  
rb [5] = new JRadio Button ("Ma\_Wit\_Yi");  
rb [6] = new JRadio Button ("Ma\_Zar\_Chi");  
ဒီတော့ rb [ ] ထဲ့ပဲ ခေါ်ယူလိုက်ရင် အပေါ်မှာ ရေးသားခဲ့တဲ့ Array Object ဖြစ်တဲ့ JRadio Button တွေ အားလုံးကို ဆိုလိုတာပါပဲ။ ဒါဆိုရင် အပေါ်က Declare လုပ်ခဲ့တဲ့ လိုင်းနံပါတ် (၅) မှာ ရေးသားခဲ့တဲ့ JRadio Button [ ] rb = new JRadio Button [7] ; ဆိုတဲ့ Code Line နဲ့ ပတ်သက်ပြီးတော့ နားလည်သဘောပါကြပြီ နော်။ ဒီနေရာမှာ အရေးကြီးတဲ့ ထည့်သွင်းရှင်းလင်းမှုတဲ့ အချက် ကတော့ လိုင်းနံပါတ် (၁၁) မှာ ရေးသားထားတဲ့ Code Line နဲ့ ကျန်တဲ့လိုင်းနံပါတ် (၂၂) ကနေ (၁၇) အထိ ရေးသားထားတဲ့ Code Line တွေရဲ့ ကွာဟာချက် မတူညီမှုပါပဲ။ လိုင်းနံပါတ် (၁၁) က ရေးသားထားတဲ့ Code Line မှာ –

rb [0] = new JRadio Button  
("Kyaw Zayar Lay", true);

String arguments အပြင် selecting Components တစ်ခုကိုပါ ထည့်သွင်း ရေးသားထားတာကို တွေ့ရမှာပါ။ ကျန်တဲ့ array object တွေ ပေါ်ဝင်ပါဘူး။ ဒါဟာ program ကို Run ပြီဆိုတာနဲ့ array Index [0] ကို always Default အနေနဲ့ အမြဲတမ်း selectလုပ်ပြီးသားဖြစ်အောင် ရေးသားထားတာပါပဲ။ ထည့်သွင်းချင်တဲ့ array-Index တစ်ခုခုမှာ အဲဒီ arguments



ကို ထည့်သွင်းရုံပါပဲ။ ကဲ ဟုတ်ဖြေလေ။ အဲဒီ code\_line မှာပါဝင် တဲ့ true ဆိတဲ့ selecting components arguments ကို မထည့်သွင်းတော့ဘူးလို့ ဆိုကြပါစို့။

အဲဒီကျရင် ဘယ်လိုဖြစ်လာမလဲ။ အဲဒီ arguments မှာ ပါဝင်တဲ့ true ကို ဖြေတားလိုက်ရင် Output ဟာ Figure 3.18.1 မှာ ဖော်ပြထားတဲ့အတိုင်း လာပြမှာပါပဲ။ Code\_Line နဲ့ Figure ကို ယုံ့တွဲပြီး ဖော်ပြထားပါတယ်။ လေ့လာကြည့်ပါ။

```
public radio(){  
    super("Choose Your's Name");  
    setSize(400,150);  
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);  
    rb[0] = new JRadioButton("Kyaw_Zayar_lay.");  
    rb[1] = new JRadioButton("Kyaw_Myo_Htet");  
    rb[2] = new JRadioButton("Ko_Bo_Min");  
    rb[3] = new JRadioButton("Ma_Zar_Zar_Oo.");  
    rb[4] = new JRadioButton("Ma_Moe_Thida");  
    rb[5] = new JRadioButton("Ma_Wit_Yi");  
    rb[6] = new JRadioButton("Ma_Zar_Chi");  
    JPanel pane = new JPanel();  
    ButtonGroup b = new ButtonGroup();  
  

```

Figure 3.18.1 Change the Code\_Line in the radio .java program.

ဒါဟာ arguments အနေနဲ့ပါဝင်တဲ့ true ဆိတဲ့ဘာကို ဖြေတားတဲ့အတွက် Output ထွက်လာတဲ့အခါမှာ selecting အနေနဲ့ ရွှေ့ချယ်ထားခြင်း မရှိတော့ပါဘူး။ ဒါဆိုရင်တော့ စာဖတ် သူအနေနဲ့ နားလည်သောပါက်သွားမှာပါ။

၃။ လိုင်းနံပါတ် (၉) မှာ ရေးသားထားတဲ့ setsize() method ကို လေ့လာကြည့်ကြရအောင်။ setsize() method မှာ ပါဝင်တဲ့ Width, Height ဆိတဲ့ arguments နှစ်ခုနေရာမှာ မိမိလိုချင်တဲ့ Width, Height အရွယ်အစားကို ထည့်သွင်းရေးသားနိုင်ပါတယ်။ ဒါကြောင့် SetSize(400,150);

ဆိုပြီးမှ ရေးသားထားတာကို တွေ့ရမှာပါ။ ကျန်တဲ့ Code\_Line ထွက်တော့ စာဖတ်သူအနေနဲ့ အရှေ့ program ထွေမှာ တွေ့မြင်ပြီးသားဖြစ်တဲ့အတွက် အလွယ်တက္က နားလည် သောပါက်မှာပါ။ ဘာကြောင့်လဲဆိုတော့ ဒီ program မှာ ပါဝင်တဲ့ Code\_Line ထွေဟာ အရှေ့ program ထွေမှာ အသုံးချ ရေးသားခဲ့တဲ့ Code\_Line ထွေ အတိုင်းပါပဲ။ အမိကကျတာကို လေ့လာကြည့်ကြရအောင်။

လိုင်းနံပါတ် (၂) မှာ ရေးသားထားတဲ့ Looping ကို လေ့လာကြည့်ကြရအောင်။

For [int i ဟာ 0 က စတယ်။ i ဟာ rb.length [JRadioButton \_Array ရဲ့ Length] ထက်တော့ ငယ် မယ်။ i ကို postfix order နဲ့ တစ်ခုပြီးတစ်ခု တိုးမယ်။ ဆိုပြီးတော့ Looping ပတ်ထွက်ထားပါတယ်။

Object\_Name b ဟာ Button Group Class ကို ခေါ်ပါပဲ။ JFrame Class ထဲမှာ ပါဝင်ပါတယ်။ Object\_Name အဖြစ် လိုင်းနံပါတ် (၁၉) မှာ ဖော်တီးထားတာကို တွေ့ရမှာပါ။ ဒီတော့ အဲဒီ Class ထဲမှာပါဝင်တဲ့ add() method ကို ခေါ်ယူ အသုံးချင်ဖို့ for looping statement ရဲ့အောက်မှာ step တစ်ခုတိုးတိုင်း brace ထဲဝင်ပြီး

```
{ b.add(rb[i]);  
pane.add(rb[i]); }
```

array \_index ကို တိုးပြီးတော့ ဖော်ပြုပေးနိုင်ဖို့အတွက် ရေးသားပေးထားပါတယ်။ Primary Class ထဲမှာပါဝင်တဲ့ ကျန်တဲ့ Code\_Line ထွေကတော့ သိပ်အထွေအထူး မရှိတော့ပါဘူး။ အလွယ်တက္က နားလည် သောပါက်သွားမှာပါ။ လိုင်းနံပါတ် (၂၀) မှာတော့ ပင်မ main() function ကို ရေးသားထားပါတယ်။ အဲဒီ main() function ထဲမှာမှ အပေါ်မှာ ဖော်တီးရေးသားခဲ့တဲ့ radio ဆိတဲ့ class ကို ခေါ်ယူလိုက်တာပါပဲ။ ဒီတော့ Constructor function ဟာ အလုံအလောက် ဝင်လာတော့တာပါပဲ။ Constructor function ကတစ်ဆင့် JRadioButton [] Array ထွေကို သွားရောက်ခေါ်ယူ အသုံးချတာပါပဲ။

ဒီအချက် ထွေဟာ Listing 3.1 မှာ ရေးသားထားတဲ့ Code\_Line ထွေကို Analysis လုပ်ကြည့်တာပါပဲ။ ကဲ ဒီတော့ Code\_Line ထွေကို နားလည်သောပါက်သွားပြီဆိုတော့ program ကို compile လုပ်ကြည့်ကြရအောင်လား။ Command Prompt (DOS) ထဲကိုသွားပါ။ CD C:\j2sdk 1.4.1\_01\Bin ဆိုပြီးမှ directory ကိုပြောင်းပါ။ ပြီးရင် CLS Command နဲ့ Clear Screen လုပ်လိုက်ပါ၌း။ ပြီးရင်တော့ အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac radio.java ↵

အကယ်၍ compile လုပ်လိုက်လို့ error တက်ခဲ့ရင် NotePad ထဲမှာ ရေးသားထားတဲ့ Code\_Line ထွေမှာ သွားပြီးတော့ ပြင်ပါ။ Java compiler ကလည်း error ကို ဘယ် Line NO: ဘယ်နေရာမှာ error ရှိတယ်ဆိုတာကိုလည်း အတိအကျ ဖော်ပေးပါတယ်။ ဒီတော့ error ကို ပြင်ပြီးရင် အဲဒီ Code\_File ကို



Save As လုပ်ပါ။ ပြီးရင် အပေါ်က ရေးပေးခဲ့တဲ့ command နဲ့ပဲ compile ပြန်လုပ်ပါ။ error မရှိတော့ဘူး ဆိုပါရို့နော်။ အောက်မှာ ရေးပြထားတဲ့ command နဲ့ Run လုပ်လိုက်ပါ။

java radio ←

အဲဒီလိုမျိုး compile & Run လုပ်ပုံကို Figure 3.18.2 အနေနဲ့ ရေးသားဖော်ပြထားပါတယ်။ လေ့လာကြည့်ပါ။

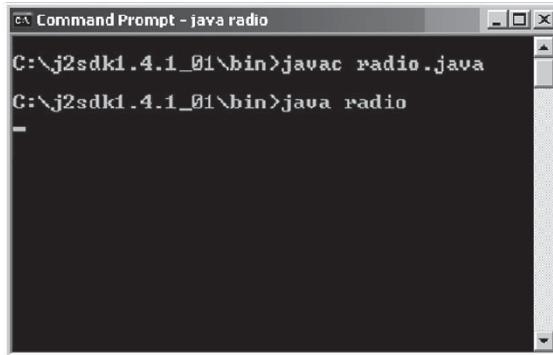


Figure 3.18.2 Compile & Run of the radio.java program

ကဲ အဲဒီလိုမျိုး Compile & Run လုပ်လိုက်ပြီဆိုတာနဲ့ Figure 3.18.3မှာ ရေးသားဖော်ပြေားထားတဲ့အတိုင်း radio.java ဆိုတဲ့ program ငယ်လေးဟာ computer Desktop Screen ရဲ့ ဘယ်ဘက်အပေါ်ခုံးတောင့်မှာ လာပြီးတော့ Run နေဖို့ပါပဲ။



Figure 3.18.3 Running the radio.java program

အခုံးပြုပဲတဲ့ output windows ဟာဆိုရင် အပေါ်မှာ ရေးသားရှင်းပြဲခဲ့တဲ့ code\_line တွေကို အလုပ်လုပ်ခိုင်းထားတာပါပဲ။ ဆက်လက်ပြီးတော့ ရှင်းပြတာကတော့ လိုင်းနဲ့ပါတ် (၉) မှာ ရေးပြထားတဲ့-

SetSize(400,150);

ဆိုတဲ့ Output Windows < JFrame ရဲ့ Frame Size > ကို Width = 400, Height = 150 ဆိုပြီးတော့မှ သတ်မှတ်ပေးထားတာပါပဲ။ အကယ်၍မှား Width = 150, Height = 250 ဖြစ်အောင် အောက်ပါအတိုင်း ပြင်ရေးလိုက်ရင်ရော ဘယ်လိုဖြစ်လာမလဲ။

SetSize(150,250);

အဲဒီအခါကျရင်တော့ အောက်မှာရေးပြထားတဲ့ Figure 3.18.4 အတိုင်း တွေ့မြင်ရမှာပါပဲ။ Code\_Line ပြင်ရေးထားပုံကိုပါ ထည့်သွင်းပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```
public radio(){
    super("Choose Your's Name");
    setSize(150,250);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    rb[0] = new JRadioButton("Kyaw_Zayar_lay",true);
    rb[1] = new JRadioButton("Kyaw_Myo_Htet");
    rb[2] = new JRadioButton("Ko_Bo_Min");
    rb[3] = new JRadioButton("Ma_Zar_Zar_Oo");
    rb[4] = new JRadioButton("Ma_Moe_Thida");
    rb[5] = new JRadioButton("Ma_Wit_Yi");
    rb[6] = new JRadioButton("Ma_Zar_Chi");
    JPanel pane = new JPanel();
    ButtonGroup b = new ButtonGroup();
```



Figure 3.18.4  
Change the  
Code\_Line Compile  
& Run of the  
radio.java

ကဲဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ ပိမိနှစ်သက်တဲ့ Frame-Size အချယ်အစားနဲ့ ပတ်သက်ပြီးတော့ ဖန်ဝါးရယူလို့ရတယ် ဆိုတာကို ရှင်းပြီ့နော်။ ဆက်လက်ပြီးတော့ JRadioButton နဲ့ ပတ်သက်တဲ့ သင်ခန်းစာအပြီးမှာ Java ရဲ့ ComboBox အကြောင်းတို့ 3.4 Using the ComboBox In JAVA ဆိုပြီးတော့ ရေးသား ပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

### 3.4 Using the ComboBox In JAVA

#### ComboBox (OR) JComboBox

Visual Basic programming language ကို အသုံးပြုတဲ့ programmer တွေ အဖို့ကတော့ comboBox ကို အသုံးချဖူးမှာ ပါ။ Visual Basic Tool Bass ထဲက Combo Box ကို D-Click ခေါက်ပြီးတော့ Form\_Windows ပေါက် ခေါ်တင်ရပါ တယ်။ ပြီးမှ Properties\_Windows ထဲကိုဝင်ပြီးတော့ Data Field တွေကို setလုပ်မယ်။ ပြီးရင် Code\_Windows ထဲကို ဝင်ရောက်ပြီးတော့လိုအပ်တဲ့ Code\_Line တွေကို ရေးသားမယ်။ ဒါဟာ VB မှာ တွေ့ဖြင့်အသုံးချရမယ့် အပိုင်းမှာပါ။

Java Language မှာတော့ visual Basic Language မှာ လောက် အသုံးချရတာ မလွယ်ကူပေးမယ် အသုံးချရမှာတော့ အထွန်ရှိရင်းပါတယ်။ ဘယ် Language မှာပဲဖြစ်ဖြစ်၊ comboBox ဟာ ထည့်သွင်းထားတဲ့ Data တွေထဲက Data တစ်ချက်ကို select လုပ်နိုင်ဖို့အတွက် ရွေးချယ်ရမယ့် Data Field ပါပဲ။ အလွယ်ကူဆုံး ပြောရရင်တော့ page Maker program မှာ page size / paper size ကို ရွေးချယ်သလိုမျိုး print ထုတ်ရမှာ စာရွက်အရွယ်အစားကို select လုပ်သလိုမျိုးပေါ့။ အဲဒီ ပုံစံမျိုး ComboBox ကို အသုံးချပါတယ်။

ComboBox ထဲက Data တစ်ချက် select လုပ်လိုက်ရင် ဆက်နွယ်နေတဲ့ အခြားသော components set တွေက Data တွေကိုပါ ပြောင်းလဲသွားအောင် ဖန်တီး ရေးသားလို့ရပါတယ်။ ဒါပေမယ့် အခုံ့ချမယ့် သင်ခန်းစာမှာ comboBox ကို အသုံးချုပ် Demonstrate လုပ်ပြုထားပါတယ်။ လေ့လာကြည့်ပါ။ အရင်ဆုံး Listing 3.15 အနေနဲ့ ရေးသားထားတဲ့ program ကို လေ့လာကြည့်ကြရအောင်လား။

#### Listing 3.15 Testing the Java ComboBox

```

1. import javax.swing.*;
2. import java.awt.*;
3. public class comboBox extends JPanel
4. {
5.     public comboBox ()
6.     {
7.         String S [] = {"Kyaw Zayar Lay", "IT Engineering",
8.                     "Katha" };
9.         JComboBox cbo = new JComboBox (S );
10.        add (cbo);
11.    }
12.    public static void main (String args [])
13.    {
14.        JFrame f = new JFrame ("Testing the JAVA"

```

```

        ComboBox" );
15.        comboBox cb = new comboBox ( );
16.        f.setSize (300, 200);
17.        f.getContentPane ().add (cb);
18.        f.setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
19.        f.show ( );
20.    }
21. }
```

#### Analysis

၁။ လိုင်းနံပါတ် (၁)နဲ့ (၂) မှာတော့ Swing Components နဲ့ awt components တွေကို ခေါ်ယူအသုံးချဖိုင်ဖို့ အောက်ပါ အတိုင်း package file နှစ်ကို Declare & import လုပ်ပါတယ်။

```

import javax.swing.*;
import java.awt.*;
အဲဒီနောက်မှာတော့ JPanel ကို extends လုပ်ပြီးတော့
comboBox ဆိုတဲ့ primary Class ကို တည်ဆောက်ပါ
တယ်။ အဲဒီနောက် constructor function ဖြစ်တဲ့
ComboBox () ဆိုတဲ့ function ကို ရေးသားပါတယ်။
အဲဒီ constructor function ထဲမှာ string Data type
ကို အသုံးပြုပြီးတော့ S [ ] ဆိုတဲ့ Demisional Array ကို
Declare လုပ်ပါတယ်။ ဒီအခါမှာ
```

```

S [0] = "Kyaw Zayar Lay"
S [1] = "IT Engineering"
S [2] = "Katha"
```

ဆိုပြီးတော့ Array\_Type အတိုင်း တွေ့ရမှာပါ။ အဲဒီနောက်
JPanel ထဲမှာပါဝင်တဲ့ JComboBox Class ကို လုပ်းပြီး
တော့ ခေါ်ယူပြီးမှ object အဖြစ် ဖန်တီးပါတယ်။ Object\_
\_Name ကတော့ cbo ပါ။ ဒါကြောင့် လိုင်းနံပါတ် (၃)
မှာ-

```

JComboBox Cbo = new JComboBox (S );
one -demisional Array-S
လို့ ရေးသားထားတာပေါ့နော်။
၂။ ကဲ ဆက်လက်ပြီးတော့ main () function ကို လေ့လာကြည့်
ကြရအောင်လား။ JFrame Class ကို ခေါ်ယူပြီးတော့
Object_ Name f အဖြစ် ဖန်တီးပါတယ်။ ဒီအခါမှာ
JFrame မဲ့ Passing Arguments နေရမှာ -
Testing the JAVA Combo Box
```

ဆိုပြီးတော့ ရေးသားပါတယ်။ ဒါဟာ Output Windows
မဲ့ Title Bar မှာ ပါဝင်ရမယ့် စာသားပါပဲ။ ဒါကို သဘော
ပေါက်မှာပါ။ လိုင်းနံပါတ် (၁၅) မှာတော့ အပေါ်မှာ ဖန်တီးခဲ့



တဲ့ comboBox ဆိတဲ့ primary class ကို self-assign လုပ်ပြီးတော့ object\_Name Cb လိုပေးပြီးတော့ ခေါ်ယူပါတယ်။ ဒီအခါမှာ constructor function ရဲ့ သဘော အတိုင်း အလိုလို ဝင်လာပြီပေါ့။

Note : Constructor function ( ) ရဲ့ သဘောတရားကို သူရဲ့ မူလသဘောအတိုင်း ရေးပြရမယ်ဆိုရင်-

The Constructor is a special method that is invoked only at the beginning of an object's lifetime. If contains any code that must be executed for the object to work ဆိုပြီးတော့ တွေ့ဖြင့်ရပါတယ်။ ဒီအချက်ကို ရှင်းပြတာဟာ ပြီးခဲ့တဲ့ သင်ခန်းစာတွေထဲကပါပဲ။ ဒါပေမယ့် swing & awt components အပိုင်းမှာတော့ JPanel/ JFrame အစရှုတာတွေကို extends လုပ်ရပါတယ်။ extends လုပ်ပြီးတော့ primary\_class တစ်ခုခုကို ရေးဖွံ့ရာမှာ constructor function သဘောတရားကို အသီးချု မယ်ဆိုရင် program code\_line ရေးသားရာမှာ အရေအတွက်ကို လျှော့ချေပေးနိုင်သလို Compile & Run လုပ်ရာမှာလည်း ပိုမိုပြီး လွယ်ကူစေနိုင်ပါတယ်။

၃။ ကဲ ဆက်လက်ပြီးတော့ လေ့လာကြည့်ကြရအောင်။ လိုင်း နံပါတ် (၁၅) မှာ Class comboBox ကိုလှုံးပြီး object အဖြစ် ဖန်တီးရာမှာ constructor function ဟာ အလို အလျောက် Invoked ဖြစ်လာပါပြီ။ အဲဒီနောက်မှာတော့ Form\_Windows ရဲ့ အရွယ်အစားကို သတ်မှတ်ပေးနိုင်ဖို့ အတွက်

f.setSize(300,200);

ဆိုပြီးတော့မှ width & Height ကို မိမိလိုချင်တဲ့ အချွေ အစား ရရှိအောင် ဖန်တီးရေးသားပါတယ်။ ကဲ အဲဒီနောက်မှာ တော့ -

getContentPane ( ) method

add ( ) method

SetDefault Close Operation ( ) method

အစရှုတာတွေကို ခေါ်ယူအသုံးချပါတယ်။ ဒီ method ( ) တွေအကြောင်းကို ရှေ့သင်ခန်းစာတွေမှာ အကြိမ်ကြံး ရှင်းပြ ခဲ့ပြီးပါပြီ။ ဒီ method ( ) တွေ API အနေနဲ့ ပါဝင်လာတဲ့ အလွန် အသုံးဝင်တဲ့ method ( ) တွေပဲ ဖြစ်တာကြောင့် ခေါ်ယူအသုံးချရမှာလည်း အလွန် လွယ်ကူမှ ရှိပါတယ်။ ကဲ အဲဒီနောက်မှာတော့ Show ( ) method ကို ခေါ်ယူ ပြီးတော့ အသုံးချထားပါတယ်။ လိုင်းနံပါတ် (၂၀) က

Close brace } ဟာ main () function ရဲ့ Close brace ပါ။ လိုင်းနံပါတ် (၂) ကတော့ primary class comboBox ရဲ့ Close brace ပါပဲ။ ဒါဆိုရင်တော့ ရှင်းသွားပြန်နော်။ ကဲ အပေါ်မှာ ရှင်းပြုခဲ့တဲ့ Listing 3.15 က code\_line တွေကို နားလည် သဘောပေါ်သွားရင်တော့ start → Accessories → NotePad ထဲတိသွားပြီးတော့ code\_line တွေကို ရှိက်သွင်းပါ။ NotePad မှာ ရှိက်သွင်းပြီးရင် File\_Name ကို combobox.java လို့ အမည်ပေးပြီးတော့ save လုပ်ပါ။ Save လုပ်ရမယ့် File\_Path လည်းကြောင်းကတော့ C:\j2sdk 1.4.1\_01\Bin ထဲမှာပါ။ အဲဒီလို code\_line တွေရောပြီး save လုပ်ထားပဲကို Figure 3.20 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```

combobox.java - Notepad
File Edit Format View Help
import javax.swing.*;
import java.awt.*;
public class combobox extends JPanel
{
    public combobox()
    {
        String s[]={"Kyaw Zayar Lay.", "IT Engineering.", "Katha."};
        JComboBox cbo = new JComboBox(s);
        add(cbo);
    }
}
public static void main(String args[])
{
    JFrame f = new JFrame("Testing the JAVA ComboBox");
    combobox cb=new combobox();
    f.setSize(300,200);
    f.getContentPane().add(cb);
    f.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    f.show();
}

```

Figure 3.20 Write the program code\_line  
in NotePad

ကဲ အခုရေးပြထားတာကတော့ Code\_Line တွေကို NotePad မှာ ရှိက်သွင်းထားတာပါပဲ။ ပြီးရင်တော့ command-line Interface <DOS> ထဲကို ဝင်ပြီးတော့ အောက်ပါ command-line နဲ့ Directory path ရှိပြောင်းပါ။

CD C:\j2sdk 1.4.1\_01\Bin ↵

ဒီအခါမှာ promptပော့ C:\j2sdk 1.4.1\_01\Bin ↵ ဆိတဲ့ prompt အတိုင်းပေါ်နော် CLS ↵ command နဲ့ Clear Screen ဖြစ်အောင် ရှင်းလိုက်ပါပြီး။

အဲဒီအခါကျေရင် Figure 3.20.1 အတိုင်း တွေ့ဖြင့်ရရင် ပဲမှာ ဖော်ပြထားတဲ့အတိုင်း-

java comboBox.java ↵

နဲ့ compile လုပ်ပါ။ ပြီးရင်

java combobox ↵

command နဲ့ Run လုပ်ပါ။ Compile & Run လုပ်ပုံကို အောက်က Figure မှာ လေ့လာကြည့်ပါ။

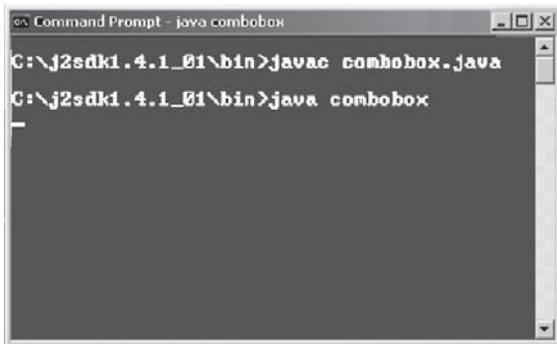


Figure 3.20.1 Compile & Run of the combobox.java program

ကဲ ဆက်လက်ပြီးတော့ အဲဒီအတိုင်း Compile & Run လုပ်လို့ ရရှိလာမယ့် Output Windows ကို Figure 3.20.2 မှာ ရေးသား ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။ အဲဒီ Figure ဟာ java combobox ကို ဖန်တီးရယူထားတဲ့ programs ငယ်လေးပါပဲ။

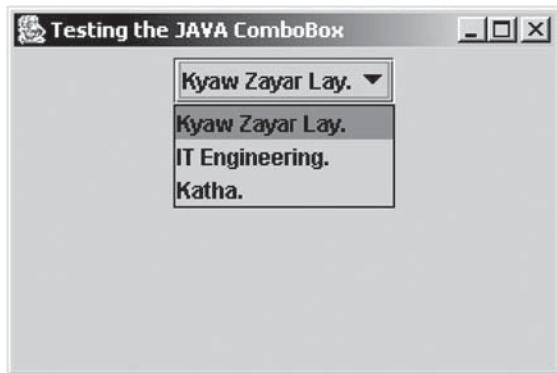


Figure 3.20.2 Running the combobox.java program

အပေါ်မှာ ရေးသား ရှင်းပြုခဲ့တာကတော့ comboBox ကို Demonstrate လုပ်ပြုခဲ့တာပါပဲ။ အခြားသော Java components တွေမှာလိုမျိုးပါပဲ။ နောက်ထပ် examples တစ်ခုအနေ နဲ့ရော comboBox လဲ၊ several methods တွေကိုရော သေသေ ချာချာ လေ့လာကြည့်ကြရအောင်လား။ ဒါကြောင့် Drop-Down lists and Combo Boxes ဆိုပြီးတော့ example program နဲ့အတူ ရေးသား ရှင်းပြပေးထားပါတယ်။ ဆက်လက် လေ့လာကြည့်ပါ။

#### ◆ Drop -Down Lists and Combo Boxes

အရှေ့သင်ခန်းစာမှာတုန်းကတော့ အခြေခံ သဘောတရားနဲ့ပဲ Demo program ရေးပြခဲ့တာနော်။ အခြေခံ Theory အချက် အလက်တွေ ဘာတစ်ခုမျှ မပါဝင်ပါဘူး။ ကျွန်ုတ်တော့ ရဲ့ ထုတ္တုံးစွဲများနဲ့ပဲမြတ်ပါဘူး။ ဒါမှသာ ဘယ် method() ကြောင့် ဘာတွေဖြစ်လာရတာလဲဆိုတာကို နားလည်သောပေါက်မှာပေါ်နော်။ Swing Class ထဲမှာပါဝင်တဲ့ JComboBox ကို အသုံးပြု မယ်ဆိုရင် drop- down lists အနေနဲ့ရော၊ ComboBox အနေနဲ့ ရော၊ user Interface အနေနဲ့ စိတ်ကြိုက် ဖန်တီးရယူလို့ ရနိုင်ပါတယ်။

- Drop-down lists ဆိုတာကတော့ အခု အသုံးချုခဲ့တဲ့ JComboBox ကို ခေါ်စေပါတယ်။ ဒါ features အနေ နဲ့ အသုံးပြုမယ်ဆိုရင် ပေးထားတဲ့ Data item field ထဲက single item တစ်ခုကို select လုပ်ပြီးတော့ ရွေးကို ရွေးချယ်ရမှာပါ။ ဒါမှသာလျှင် program ရဲ့ ဆက်လုပ်ရ မယ့်အတိုင်းမှာ ပါဝင်ပြီးတော့ လုပ်ဆောင်နိုင်မှာပါ။ [Option Button (OR) Radio Button နဲ့ အနည်းငယ် ဆင်တူပါတယ်] ဒါကြောင့် သူ့ကို choice lists လို့လည်း ခေါ်စေပါတယ်။

*Note :* ကျွန်ုတ်တော့ အလွယ်တကူပဲ Drop-lists လို့ပဲ ခေါ်ပါတယ်။

- Combo Boxes အနေနဲ့ကျတော့ မူလ Drop-lists features ထက် ပိုမြဲး ကျယ်ပြန်လာပါတယ်။ ဘာတွေကျယ်ပြန်လာသလဲဆိုတာကို လက်တွေ့လေ့လာပြီးတော့ အသုံးချုပ်လည့်ကြရအောင်လား။  
JComboBox() ကို ခေါ်ယူ အသုံးချုမယ်ဆိုရင် သူ့ရဲ့ Class ထဲမှာ ပါဝင်တဲ့ constructor() function ဟာ အလိုအလျောက်ပါဝင်လာပါပြီ။

အထူးသတိပြုရယ် အချက်ကတော့ JComboBox() ရဲ့ constructor function မှာ arguments လုံးဝ မပါဝင်ပါဘူး။ အလွယ်ပြောရရင်တော့ JComboBox() has with no arguments ပေါ့နော်။

နောက်တစ်ခုကျက်ကတော့ JComboBox() method ကို အသုံးပြုပြီးတော့ Data -Field ထဲကို Data item တွေ ထည့်သွင်းမယ်ဆိုရင် add Item (object) method ကို အသုံးပြုရပါမယ်။ ဒါမှသာလျှင် List ထဲကို Data Item တွေကို ဖြည့်သွင်းပြီးသားဖြစ်မှာပေါ့နော်။ ဒါက Data Item တွေကို ကိန်းသော Input လုပ်ထားတာကို ပြောတာပါ။ ဒီအခါမှာ SetEditable()



method ကို အသုံးပြုရမှာနော်။ အသုံးပြုရမယ့် arguments ကိုတော့ true လုပ်ထားရမှာပေါ့။ ဒါဟာ Drop-down lists ကိုဖန်တီး အသုံးချရမယ့် နည်းလမ်းပါပဲ။ ComboBox အနေနဲ့က တော့ အဲဒီ field ထဲကို user က enter text လုပ်ဖို့တာနဲ့ တစ်ဖက် Drop-List ကနေပြီးတော့ Data Item ကို pick လုပ်ပါတယ်။

ကဲ ဘယ်လိုပြုဖြစ်ဖြစ်ပေါ့နော်။ JComboBox Class ထဲမှာ ပါဝင်တဲ့ serval method တွေကို လေ့လာကြည့်ကြရအောင်လား။ အခု ရေးသားပေးမယ့် serval method တွေကတော့ Drop-down list (OR) combo Box ကို control လုပ်မယ့် method တွေပါပဲ။ မူလ method တွေအတိုင်းပဲ ရေးသားပေးထားပါတယ်။

- getItemAt (int) - Return the text of the list item at the index position specified by the integer argument. As with arrays, the first item of a choice list is at index position 0, the second at position 1, and so on.
- getItemCount () - Return the number of items in the list.
- getSelectedIndex () - Return the index position of the currently selected item in the list.
- getSelectedItem () - Return the text of the currently selected item.
- setSelectedIndex (int) - Select the item at the indicated index position.
- setSelectedIndex (object) - Select the specified Object in the list.
- setMaximumRowCount (int) - Set the number of rows in the combo box that are display at one time.

ကဲ အခု ရေးပြခဲ့တာတွေကတော့ JComboBox Class ထဲမှာ ပါဝင်တဲ့ JComboBox (OR) Drop-down list နဲ့ ယဉ်တဲ့ ပြီးတော့ အသုံးချရမယ့် method () တွေပါပဲ။ တစ်နည်းအားဖြင့် JComboBox အထဲမှာ ပါဝင်ပြီးသား method တွေပါပဲ။ ဒီတော့ JComboBox နဲ့ ပတ်သက်တဲ့ သင်ခန်းစာတစ်ပုံပို့ ဆက်လက် လေ့လာကြရအောင်။ Listing 3.16 အနေနဲ့ ရေးသားပေးထားပါတယ်။ သေသေချာချာ လေ့လာကြည့်ပါ။

#### Listing 3.16 Advanced Method of JComboBox

1. import javax.swing.\*;
- 2.
3. public class combo extends JFrame {
4. JComboBox monthBox = new JComboBox () ;
5. JComboBox yearBox = new JComboBox () ;
- 6.

```

7.    public combo () {
8.        super ("selecting the Date");
9.        setSize (350,250);
10.       setDefaultCloseOperation(JFrame.EXIT_ON_
11.           CLOSE);
12.       JPanel pane = new JPanel ();
13.       pane.add (exp) ;
14.       for (int i = 1; i < 13; i + +)
15.           month .addItem ( " " + i);
16.       for (int j = 2006; j < 2020; j + +)
17.           year .addItem ( " " + j);
18.       pane.add (month) ;
19.       pane.add (year);
20.       setcontentPane (pane) ;
21.       Show () ;
22.   }
23.
24.   public static void main (String [ ] args) {
25.       combo Cb = new Combo () ;
26.   }
27. }
```

ကဲ ဒါဆိုရင် Listing 3.16 မှာ ရေးပြထားတဲ့ code\_line တွေကို Analysis လုပ်ကြည့်ကြရအောင်လား။ အရင် လေ့လာက တာပေါ့။

#### Analysis

၁။ လိုင်းနံပါတ် (၁) မှာ import java.swing.\*; package ကို Declare လုပ်ပြီး အော်လားပါတယ်။ လိုင်းနံပါတ် (၂) မှာမူ combo ဆိုတဲ့ primary class ကို extends လုပ်ယူ ဖန်တီးပါတယ်။ အဲဒီ primary class ထဲမှာမူ constructor () function ကို မရေးခင်မှာ – monthBox

yearBox အစုရိတ္တာ Variable\_Name နှစ်ခုကို [တစ်နည်း ပြောရင်တော့ object\_Name ပါပဲ။] Declare လုပ်ပါတယ်။ ဒါဟာ Java Swing Package ထဲမှာပါဝင်တဲ့ component object တွေဖြစ်တဲ့ JComboBox ကို အော် အသုံးချတာပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၃) မှာမူ combo () ဆိုတဲ့ constructor function ကို ရေးဖွံ့ပါတယ်။ လိုင်းနံပါတ် (၄) မှာမူ

Super ("Selecting the Date");  
ကို ရေးသားထားတာပါ။ ဒါဟာ Output အဖြစ်ရလာမယ့် Frame\_Windows ဒဲ Title Bar မှာ ပါဝင်ရမယ့် စာသား



အတွက်ပါပဲ။ Super () Interitance Method နဲ့ အသုံးပြုထားတာပါ။ ပြီးတော့ Frame\_Windows ပဲ အဆင့်အစားအတွက် setSize () method ကို အသုံးချိပြီးတော့ Width=350, Height = 250 ဆိုပြီးတော့ သတ်မှတ်ပါတယ်။ ပြီးတော့ program\_Run နေစဉ်မှာ EXIT\_ON\_CLOSE လုပ်နိုင်ဖို့ အတွက် setDefault Close Operation () method ကို လိုင်းနံပါတ် (၁၀) မှာ ခေါ်ယူ အသုံးချထားပါတယ်။ ကဲပြီးတော့ JPanel Class ကို Object \_Name pane အဖြစ် ပြောင်းလဲ ဖန်တီးပါတယ်။ ဒီတော့ JPanel Class ရဲ့ Object \_Name ဟာ pane ဖြစ်သွားပြီးနောက်။ ပြီးတော့ JLabel Class ကို ခေါ်ယူအသုံးချထားလိုင်းနံပါတ် (၁၃) မှာ ရေးသားထားတဲ့ pane.add(exp);

ဆိုတဲ့ code\_line ဟာ ဖန်တီးထားတဲ့ JLabel တွေကို JPanel ထဲကို add () လုပ်နိုင်ဖို့ ရေးသားထားတဲ့ code\_line ပါပဲ။

JII Month & Year တွေကို ရိုက်ထုတ်ပေးနိုင်ဖို့ အတွက် for looping ကို အသုံးချထားတာကို တွေ့ရမှာပါ။ month.addItem(" " + i) ကို ရိုက်ထုတ်ပေးနိုင်ဖို့ အတွက် for (int i ဟာ 1 က စတယ် ; i ဟာ 13 ထက်တော့ ငယ်လိုရတယ် ; i ကို postfix order နဲ့ နောက်တစ်လိုင်းမှာမှ 1 တိုးမယ်ဆိုတဲ့ Looping နဲ့ ရိုက်ထုတ်ပါတယ်။ အဲဒီလိုပါပဲ။ for (int j ဟာ 2006 ကစတယ် ; j ဟာ 2020 ထက်တော့ ငယ်လိုမယ် ; j ကို postfix order increment နဲ့ နောက်တစ်လိုင်းမှာမှ 1 တိုးမယ်ဆိုပြီးတော့ ရေးသားပေးထားပါတယ်။ ဒါဟာ month & year ကို looping ကို တွေ့ရိုက်ထုတ်ပါတယ်။ အဲဒီနောက် JPanel Class ထဲမှာ ပါဝင်တဲ့ add () method ထဲကို looping ပါဝင်ထားတဲ့ month & year ကို ထည့်သွင်းပါတယ်။

ဒါတွေဟာ combo () constructor function ထဲမှာ ပါဝင်တဲ့ အဓိက program ရဲ့ ကျော်ရှုံးအပိုင်းတွေပါပဲ။ ဒီပိုင်းမှာ ပိုင်းကို တည်ဆောက်ပြီးသွားတဲ့ အခါမှာ constructor () function ကို ဝင်လာအောင်လို့ class ကို object အဖြစ် ဖန်တီးလိုက်တာနဲ့ အလုံအလျောက် ဝင်လာတော့မှာပါ။ ဒါကြောင့် လိုင်းနံပါတ် (၂၄) မှာတော့ main () function ကို ရေးသားထားပါတယ်။ အဲဒီ main () function ထဲမှာမှ combo ဆိုတဲ့ class ကို cb ဆိုတဲ့ Object \_Name အဖြစ်ပေးပြီး ခေါ်ယူအသုံးချလိုက်ပါတယ်။ ဒါဆိုရင်အပေါ်မှာ ရေးသားခဲ့တဲ့ ပင်မကျော်ရှုံးအပိုင်းဖြစ်တဲ့ constructor () function ဟာ အလုံအလျောက် ပါဝင်အလုပ်လုပ်သွားတော့မှာပါ။ ကဲခိုး code\_line တွေကို နားလည်

သောာပေါက်ပြီးနောက်။ ဒါဆိုရင်တော့ အဲဒီ code\_line တွေကို NotePad ထဲမှာ သွားရောက်ပြီးတော့ ရေးသားကြရအောင်လား။ NotePad ထဲမှာ code\_line တွေ ရေးသားထားပဲကို Figure 3.20.3 အနေနဲ့ ရေးသားပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```

combo.java - Notepad
File Edit Format View Help
import javax.swing.*;
public class combo extends JFrame{
JComboBox month = new JComboBox();
JComboBox year = new JComboBox();
public combo() {
super("Selecting the Date ");
setSize(350, 250);
setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
JPanel pane = new JPanel();
JLabel exp = new JLabel("Selecting the Date ");
pane.add(exp);
for(int i=1;i<13;i++)
month.addItem(""+i);
for(int j=2006;j<2020;j++)
year.addItem(""+j);
pane.add(month);
pane.add(year);
setContentPane(pane);
show();
}
public static void main(String[] args){
combo cb = new combo();
}
}

```

Figure 3.20.3 Write the program code\_line In NotePad

အချို့ဖြစ်သွားပါသေားတဲ့ Figure 3.20.3 ဟာ Listing 3.16က program code\_line တွေကို ရေးသားပြီးတော့ save လုပ်ထားတာပါပဲ။ Saveလုပ်ရမယ့် CLASS PATH'S ကတော့ C:\j2sdk 1.4.1\_01\Bin ထဲမှာနောက်။ File\_Name ကတော့ combo.java ပဲ ဖြစ်ပါတယ်။ ဒါ အဲဒီ File ကို Save လုပ်ပြီးရင် command Prompt <DOS> ထဲကို သွားပြီးတော့ CD C:\j2sdk 1.4.1\_01\Bin က ဆိုပြီးမှ Directory path ကိုပြောင်းပါ။ ပြီးရင် အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac combo.java ↵

အကယ်၍ compile လုပ်လိုက်လို့ error မရှိဘူးဆိုရင်တော့ အောက်ပါ command နဲ့ Run လုပ်လိုက်ရှုံးပါပဲ။

java combo ↵

အဲဒီလို command prompt <DOS> ထဲမှာ compile& Run လုပ်ပဲကို Figure 3.20.4 မှာ ရေးသားဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



```
Command Prompt - java combo
C:\j2sdk1.4.1_01\bin>javac combo.java
C:\j2sdk1.4.1_01\bin>java combo
-
```

Figure 3.20.4 Compile & Run of the combo.java program

ဒီအခါမှာတော့ program ရဲ output ကိုအောက်က Figure 3.20.5 အတိုင်း တွေဖြင့်ရမှာပါပဲ။ လေ့လာကြည့်ပါဦး။

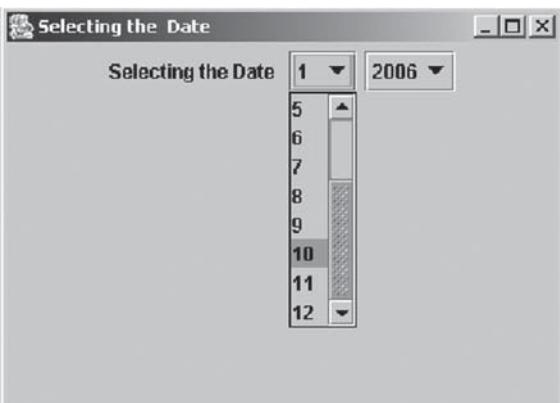


Figure 3.20.5 Running the Combo.java Program

ကဲ ဒါဆိုရင်တော့ JComboBox (OR) DropDownList အကြောင်းကို နားလည်သောပေါက်ပြီးတော့ program တွေရေးတတ်လာမှာပါပဲ။

ဆက်လက်ပြီးတော့ 3.5 Using the ImageIcon In JAVA ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် လေ့လာလိုက်ပါဦး။ Java Language ထဲမှာ Image Icon Type တွေကို အသုံးချ ရေါပြထားပါတယ်။ သေသေချုံချုံလေ့လာကြည့်ပါ။ Image File တွေကို File\_type ပြောင်းလဲပြီးတော့ အသုံးချုံနည်းလည်း ပါဝင်ပါတယ်။

## ဘာတွေ ပျောက်ပျောက်

### ဘယ်နေရာ ရောက်ရောက်

အသက်ကလေး ရလာတယ်ဆိုရင်ပဲ သတိကမှုတတ်လာပါတယ်။ ကားသေ့ဗို့ အခုပု စားပွဲပေါ်တင်လိုက်မိတယ်ထင်တယ်။ ရှာမတွေ့တော့ပါ။ Loc8tor Ltd ([www.loc8tor.com](http://www.loc8tor.com))မှ ထုတ်လုပ်လိုက်တဲ့ Loc8tor tracking device ကိုပဲအားကိုးပူးပါမယ်။ အဆိုပါကိုရှာလေးက အရွယ်အစား ၁၉.၅ × ၂၀.၅ × ၈.၅ မီလီမီတာသာရှိလို့ မိဘိုင်းလုပ်နှင့်တို့လည်း ပြောနိုင်ပါတယ်။ သယ်ယူရလွယ်ကူသလို့ အသုံးပြုရတာလည်း လွယ်ပါတယ်။ ပေ ၆၀၀ ဝန်းကျင်မှာ ရှိနေတဲ့ ကျွန်ုင်တော်တို့ရဲ့ ပျောက်ဆုံးနေတယ်လို့ ထင်ရသော ပစ္စည်းတွေကို ရှာပေးမှာပါ။ ပစ္စည်းဆုံးလို့ သေ့ဗို့တွေ့၊ အငေးထိန်း ပစ္စည်း အတိုအတွေ့ ဆိုပါတော့။ ပထမတော့ အဲဒီထဲက ပရှိကရမ်ထဲကို သင် ရှာပေးစေခဲင်တဲ့ ပစ္စည်းတွေကို ထည့်ရပါမယ်။ ဒီလို့ ထည့်နိုင်တဲ့ အရေအတွက်ပေါ် မူးတည်ပြီး စက်ကလေးရှုံးတာနိုးက ၉၉ ဒေါ်လာကနေ ၁၆၉.၉၉ ဒေါ်လာအထိ ရှိနိုင်ပါတယ်။ ဒီလို့ စက်ကလေးကို ကျွန်ုင်တော်တို့လို့ သတိမှုတတ်သူတွေသာမက ကလေးတွေနဲ့ အဘို့း အဘွားတွေလည်း ဆောင်ထားသင့်ပါတယ်။ ဘယ်လို့ အသုံးဝင်မယ် ဆိုတာထက် လက်တွေ့၊ အသုံးကြည့်ရင် ကျေးဇူးတင် မိလိမ့်မယ်လို့ ဆိုလိုက်ပါရစေ။



KYAW ZAYAR LAY [ IT ENGINEERING ]



## JAVA DEVELOPER GUIDE

### 3.5 USING THE IMAGE ICON IN JAVA

Listing 3.17 Demonstrate the JAVA ImageIcon

```
1. import javax.swing.*;  
2.  
3. public class icon extends JFrame {  
4. JButton [ ] buttons = new JButton [25];  
5. public icon () {  
6. Super ("Demonstrate the Image Icon in JAVA");  
7. setSize (480,380);  
8. setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);  
9. JPanel pane = new JPanel ();  
10. ImageIcon ic = new ImageIcon ("fo ood.GIF"); / * this. GIF image is  
11. include in the C:\j2sdk 1.4.1_01\Bin\fo ood.GIF,  
if this image is not include  
12. you will use to another image.GIF type */  
13.  
14. for (int i = 0; i < 25; i + +) {  
15. buttons [i] = new JButton (ic);  
16. pane.add (buttons [i]);  
17. }  
18. set ContentPane (pane);  
19. show ();  
20.  
21. }  
22.  
23. public static void main (String [ ] args) {  
24. icon c = new icon ();  
25. }  
26. }
```

ကဲ ImageIcon နဲ့ ပတ်သက်ပြီးတော့ ပို့ချေရာမှာ အကျယ်ချုပ်မရင်းပြခင်မှာ အပေါ်မှာရေးပြခဲ့တဲ့ Listing 3.17 ကို အရင်ဆုံး

လေ့လာကြည့်လိုက်ပါၤး။ ဒီသင်ခန်းစာမှာ JAVA ImageIcon တွေကို ဘယ်လိုမျိုး ဖုန်တိုးရယူနိုင်သလဲဆိုတာကို တွေ့မြင်ရမှာပါ။ ImageIcon တွေနဲ့ ပတ်သက်ပြီးတော့ <http://www.java21pro.com> ၍ Day 9 page မှာ သွားရှာမယ်ဆိုရင် အလွယ်တကူ ရှုံးနိုင်ပါတယ်။ တကယ်လို့ စာဖတ်သူဟာ Icon အမျိုးမျိုးကို ရုပ်တယ်ဆိုရင်တော့ အောက်ပါ website မှာ သွားယူ ရယူနိုင်ပါတယ်။

<http://www.Zeldman.com/icon.html>

အကယ်၍ စာဖတ်သူဟာ ImageIcon တွေကို အလွယ်တကူ ရှာဖွေလို့ မရဘူးဆိုရင်တော့ အခြားသော \*.bmp, \*.JPG, \*.JPEG, \*.TIF, \*.ICO အစရိတ္တဲ့ မိမိနှစ်သက်တဲ့ပုံ ဘယ်လို File\_type မျိုးပဲဖြစ်နေပါစေ။ အလွယ်တကူနဲ့ \*.GIF File\_type ကို ပြောင်းလဲပြီးတော့ ခေါ်ယူ ထည့်သွင်း အသုံးချုနိုင်ပါတယ်။ ဒီတော့ အခုပ်ကို listing 3.17 ကို မရှင်းပြခင်မှာ Image တွေကို File\_type ပြောင်းလဲတဲ့နည်းကို အရင်လေ့လာကြည့်ကြပါၤ့။

#### ◆ Changing the File\_type In \*. GIF type

အရင်ဆုံး စာဖတ်သူနှစ်သက်တဲ့ picture ပုံကို ရှာဖွေပါ။ အခိုက်အချက်ကတော့ အခုပ်ကိုရှုံးသင်ခန်းစာမှာ ထည့်သွင်း အသုံး ချမယ့်ပုံ ဖြစ်နိုင်တော့ ပို့ပြီး အပေါင်ပြတောပေါ်နေ၏။ ကျွန်ုတ်ကတော့ Listing 3.17 မှာ အသုံးပြထားတဲ့ Image နဲ့ မဆိုင်တဲ့ အခြား software.icon ဆိုတဲ့ picture ပုံလေးကိုပဲ File\_type ပြောင်းပေးထားပါတယ်။ ဘယ်လို Image လေးပဲဖြစ်ဖြစ်ပေါ်နေ၏။ Copy ကူးယူပြီးတော့ My Computer → Drive C: → j2sdk 1.4.1\_01 → Bin ထဲကိုသွားပါ။ Path အတိုင်းပြောမယ်ဆိုရင် တော့ C:/j2sdk 1.4.1\_01\Bin ထဲကိုသွားပါ။ ပြီးရင် အဲဒီပုံကို

Paste ချလိုက်ပါ။ ကျွန်တော်ကတော့ အောက်မှာ ဖော်ပြပေးထား ထဲ့ Software.Icon Image ကိုပဲ အသုံးပြုပေးထားပါတယ်။ ကျွန်တော့ရဲ့ code\_File CD ထဲမှာတော့ ပါဝင်ပါတယ်။



Figure 3.21.1 The Icon image

ကဲ ဒီပိုလေးပဲဖြစ်ဖြစ်၊ စာဖတ်သူသဘောကျတဲ့ ပိုလေးပဲဖြစ်ဖော်နော်။ ကျွန်တော်ပြောတဲ့ C:\j2sdk 1.4.1\_01\Bin ထဲမှာ သုံးပြီးတော့ ထားပြီးသွားပြီးရရှိရင် File\_type ဖြောင်းနဲ့အတွက် အဲဒီ Image ပဲကို paint program နဲ့ ဖွင့်ရမှာပါ။ ဒီတော့ အဲဒီ Image ကို R-click ခေါက်ပြီးတော့ ပေါ်လာတဲ့ Pop-up Menu ထဲက Open With → Paint ကို ရွေးပါ။ အောက်မှာရေးပြထား ထဲ့ Figure 3.21.2 မှာ လက်တွေလုပ်ပြထားပါတယ်။ လေ့လာ ကြည့်ပါ။

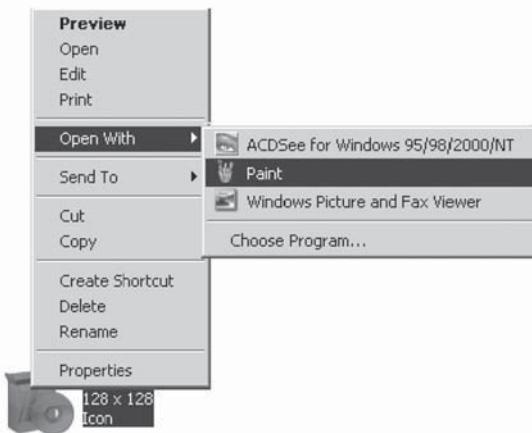


Figure 3.21.2 Open With the Paint Program Using the Icon Image

အဲဒီအခါကျေရင်တော့ အဲဒီ Image ပဲဟာ paint program နဲ့ ဖွင့်လာပါပြီ။ အဲဒီကို စာဖတ်သူဟာ အောက်ပါအတိုင်း တွေဖြင့် ရမှာပါ။ ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

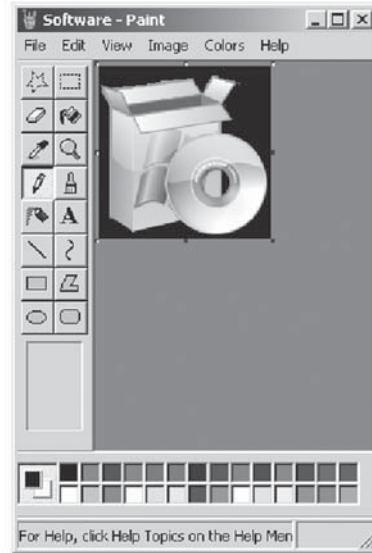


Figure 3.21.3  
Opening the  
Paint  
Program, the  
Icon Image is  
Showing

အဲဒီအခါကျေရင် စာဖတ်သူအနေနဲ့ File menu ကနေပြီးတော့ Save As ကိုတောင်းပါ။ အောက်မှာဖော်ပြထားတဲ့အတိုင်း File → Save As ကို တောင်းရမှာနော်။ Figure 3.21.4 Save As the Icon Image in Paint Program က အပေါ်မှာဖော်ပြထားတဲ့အတိုင်း Save As menu ကို တောင်းလိုက်ပြီ ဆိုကြပါစို့။ ကျွန်တော်တို့က အောက်ပါ Image ပဲကို C:\j2sdk 1.4.1.01\Bin ထဲမှာ ထားတာဆိုတော့ ဖွင့်လာမယ့်၊ ကျေလာမယ့် Save As Dialog Box ကလည်း bin ထဲမှာပဲ Save As လုပ်ရမယ့် Dialog Box ပဲပိုပါပဲ။ အောက်မှာဖော်ပြထားတာကို လေ့လာကြည့်ပါ။

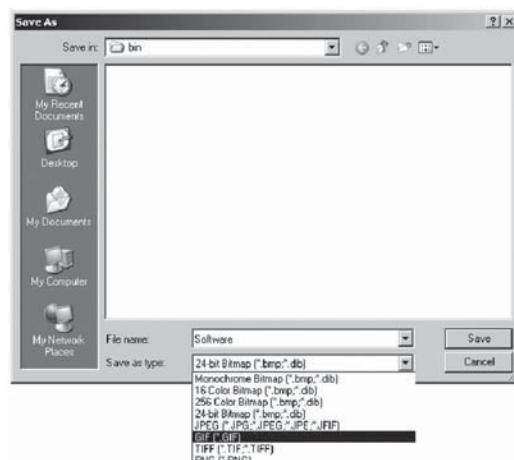


Figure 3.21.5 To Select the GIF (\*.GIF) File Type in Save As Dialog Box.



အဲဒီပေါ်လာတဲ့ Save As Dialog Box ထဲက အောက်ခြားမှ ပါဝင်တဲ့ Save as type ဆိုတဲ့နေရာမှာ GIF [\*GIF] ဆိုတဲ့ file\_type ကို ရွေးချယ်ပါ။ အဲဒီဟာ လက်ရှိ Image ရဲ့ File\_type ကို ပြောင်းလဲပစ်ဖို့ပါပဲ။ ဖော်ပြထားတဲ့ Figure ကို လေ့လာပြီးတော့ သေသေချာချာ ရွေးချယ်ပါ။ သဘောပေါက်ဖြေနော်။ File\_type မမှားပါစေနဲ့။ File\_type ကို ရွေးပြီးရင် Save As Button ကိုနှိပ်ပါ။

အဲဒီအခါမှာ \*.Ico Image Type ကနေပြီးတော့ \*.GIF Image Type ကို ပြောင်းလဲပြီးမှ Save လုပ်လိုက်တဲ့အတွက် Some loss of Color Information ဖြစ်သွားတာတွေကို သတိပေးတဲ့ Message Box တစ်ခု ပေါ်လာပါလိမ့်မယ်။ Yes\_Button ကို နှိပ်လိုက်ပါ။ ပေးလာမယ့် Message Box ကို ဖော်ပြပေးထားပါတယ်။

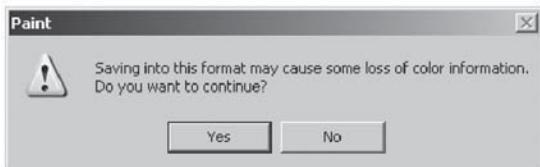


Figure 3.21.6 Prompt the Message, Changing the File Type of \*. Ico Image

ကဲ ဒီ Message Box မှာပါဝင်တဲ့ Yes\_Button ကို နှိပ်လိုက်ပြီးဆိုပါဖို့နော်။ color Information အချို့ဟာ ပြောင်းလဲသွားပါတယ်။ အောက်မှာဖော်ပြပေးထားတဲ့ paint program ကို လေ့လာကြည့်လိုက်ပါ။ သတိပြုမိပါလိမ့်မယ်။

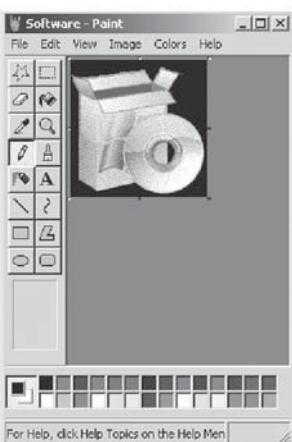


Figure 3.21.7 \*.  
GIF color format  
type is cause some  
lost color  
information

ပြီးရင်တော့ အုပ်ဖော်ပြထားတဲ့ Paint Program ရဲ့ ညာက် အပေါ်ထောင့်မှုရှိတဲ့ Close Button ကိုပိတ်ချုပ်ဖြစ်စေ။

File → Exit ကို ရွေးပြီးဖြစ်စေ program ကို ပိတ်ပစ်လိုက်ပါ။ ဒီအခါမှာ C:\j2sdk 1.4.1\_01\Bin ကို သွားကြည့်မယ်ဆိုရင် [My\_computer → Drive C: → j2sdk 1.4.1\_01 → Bin ထဲကို D-Click ခေါက်ပြီးသွားရင်ပဲ့] မှုလရှိတဲ့ Software. Icon Type Image ပုံစံအပြင် အောက်ဖော်ပြပါပဲဖို့နဲ့ Software.GIF ဆိုတဲ့ Image ကိုပါ .\*.GIF File\_type နဲ့ တည်ရှိနေတာကို စာဖတ်သူအနေနဲ့ တွေ့ရှိရမှာပါပဲ။ လေ့လာကြည့်ပါ။



Figure 3.21.8 The GIF Image

အခုဖော်ပြထားတဲ့ Image Icon လေးကို စာဖတ်သူရဲ့ C:\j2sdk 1.4.1\_01\Bin ဆိုတဲ့ folder ထဲမှာတွေ့မြင်ရမှာပဲ။ ဒီတော့ အဲဒီ GIF Image ရဲ့ properties ကိုသိချင်ရင် Image ကို R-click ခေါက်ပြီးတော့ ပေါ်လာတဲ့ Pop-up Menu ထဲက အောက်ဆုံးမှာ ပါဝင်တဲ့ Properties ကို ရွေးချယ်ပါ။ အဲဒီအာကျုရင် Software ဆိုတဲ့ Image ဖြစ်တဲ့အတွက် Software Properties ဆိုပြီးတော့ Properties Dialog Box ပေါ်လာမှာပါ။ Figure 3.21.8 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။

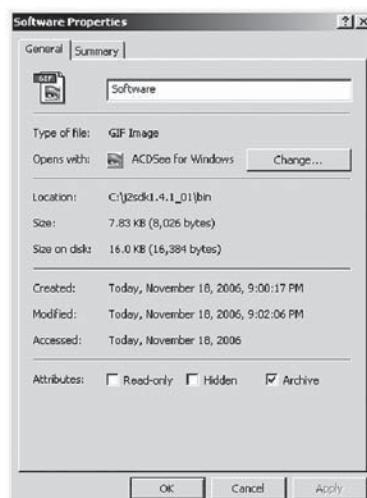


Figure 3.21.8  
The properties  
windows.\*GIF  
Type of Image

အုပ်ဖော်ပြထားတဲ့ Figure 3.21.8 ဟာ Image ရဲ့ File\_type ကို \*.GIF Type အဖြစ် ပြောင်းလဲပေးပြီးတော့မှ ဖော်ပြပေးထား၊ File\_size အစရိတာတွေကို လေ့လာလို့ရအောင် ဖော်ပြပေး

ထားတဲ့ properties windows ပါပဲ။ လေ့လာကြည့်ပါ။ ကဲဒါဆို ရင်တော့ မိမိနှစ်သက်တဲ့ အမြား File\_type များကို .\*GIF အဖူးး အစား File\_type အဖြစ် ရရှိအောင် ပြောင်းလဲတတ်သွားမှာ ဖြစ် ပါတယ်။ ဒီတော့ စာဖတ်သူရဲ့ computer ထဲမှာရှိတဲ့ မိမိနှစ်သက် တဲ့ picture တွေကို ပြောင်းလဲ ထည့်သွင်း အသုံးပြုနိုင်ပါတယ်။ ဒါဆိုရင်တော့ ဆက်လက်ပြီးတော့ လေ့လာကြရအောင်လား။

**Listing 3.17** မှာ ရေးသားထားတဲ့ Demonstrate the JAVA Image Icon ဆိုတဲ့ သင်ခန်းစာကို Analysis လုပ်ပြီးတော့ လေ့လာကြည့်ကြရအောင်။  
Analysis

- ၁။ လိုင်းနံပါတ် (၁) မှာ Swing package ကို import လုပ် ခေါ်ပူးထားပါတယ်။ ပြီးတော့ JFrame Class ကို extends လုပ်ပြီးတော့ icon ဆိုတဲ့ primary class ကို ရေးသားဖော်တီး ထားပါတယ်။ အဲဒီ primary class ထဲမှာမှ Demisional Array Method နဲ့ JButton [ ] ကို ဖန်တီးပါတယ်။ Array အရေအတွက်ကတော့ [25] ခုပါ။ အဲဒီပြီးတော့ icon ( ) ဆိုတဲ့ constructor function ကို ဖန်တီးပါတယ်။ အဲဒီ constructor function ထဲမှာ ဖန်တီးမယ့် program မှာ ပါဝင်ရမယ့် [Title bar မှာ ပါဝင်ရမယ့် text] ရယ်။ Windows Frame ရဲ့ Width & Height အရွယ်အစား၊ နောက်ပြီးတော့ Windows ကို Close လုပ်ဖို့ Set Default Close Operation () method အစ ရှိတာတွေကို ထည့်သွင်း ရေးသားထားပါတယ်။ အဲဒီနောက် လိုင်းနံပါတ် (၉) မှ JPanel ကို object\_Name pane အဖြစ် ဖန်တီး ခေါ်ပူးပါတယ်။ ကဲ ဒီအထိ ရှင်းပြီးနောက်။
- ၂။ လိုင်းနံပါတ် (၁၀) မှာတော့ ImageIcon ကို ဖန်တီးရယူမှု အတွက် အောက်ပါအတိုင်း ရေးသားထားပါတယ်။

```
Image Icon ic = new Image("foood.GIF");
                         ↑
                         Image
                         Icon
```

ထည့်သွင်းလိုသည့် \*.GIF type  
Image ရှိသည့် နေရာ၊ path  
လမ်းကြောင်းကို ထည့်သွင်းလျှင်လည်းရသည်။ ဥပမာ ["C:\j2sdk1.4.1.01\bin\foood.GIF"]

အထက်ပါအတိုင်း ရေးသားလိုက်ပါက C:\j2sdk 1.4.1\_01\bin folder ထဲရှိ တည်ရှိနေသော foood.GIF Image ကိုသာ ခေါ်ယူရှိနိုင်မည်ဖြစ်သည်။ အခြားသော folder \_path လမ်းကြောင်းထဲရှိ .\*GIF Image ကို ရယူချင်ပါက

ငှုံး Image တည်ရှိရာ path လမ်းကြောင်းကို " " Double quote နှစ်ခုကြားတွင် ရေးသားရမည်ဖြစ်သည်။ လိုင်းနံပါတ် (၉) က code\_line ကတော့ C:\j2sdk 1.4.1.01\bin ထဲ ဖြစ်တဲ့အတွက် ကြောင့် path လမ်းကြောင်း ထည့်ရေး စရာ မလိုပါဘူး။

အဲဒီနောက်က ရေးသားတဲ့ /\* ခံပြီး ရေးသားတာကတော့ comments ပါပဲ။ ထည့်မရေးလည်းရပါတယ်။ လိုင်းနံပါတ် (၁၂) အထိ ရေးသားထားပါတယ်။ \*/ ပိတ်ထားတဲ့အထိပါပဲ ပဲ။ ထည့်ရေးရင်လည်း Java Compiler က အသိအမှတ် မပြုပါဘူး။ လိုင်းနံပါတ် (၁၄) မှာမှ for looping ကို အသုံးပြုပြီးတော့ အပေါ်မှာ Declare လုပ်ခဲ့တဲ့ JButton [ ] တွေကို ပယ်ထုတ်ပါတယ်။ ဒီ looping ပါဝင်တဲ့ လမ်းကြောင်းကိုတော့ အရေးသင်ခန်းစာတွေမှာ တွေ့မြင်ဖြူပြီးသား ဖြစ်တဲ့အတွက် အကျယ်ချဲ့ပြီးတော့ မရင်းပြတော့ပါဘူး။

**Note :** သတိပြုရမယ့်အချက်ကတော့ လိုင်းနံပါတ် (၄) မှာ JButton [ ] ကို object\_Name buttons လို့ အမည် ပေးသားတာနော်။ လိုင်းနံပါတ် (၁၀) မှာကျတော့ Image Icon ကို object\_Name ic လို့ အမည်ပေးပြီးတော့ ခေါ်ယူထိက်တာပါပဲ။ လိုင်းနံပါတ် (၁၆) မှာ

pane.add (buttons [i]);

ရေးသားထားတာပါ။ အဲဒီမတိုင်ခင် လိုင်းနံပါတ် (၁၅) မှာလည်း Button [ ] တွေထဲကို Image Icon ကျ ပါဝင်လာဖို့အတွက်

buttons [i] = new JButton (ic);

လို့ ရေးသားထားတာပါပဲ။

က ဒီအထိ ရှင်းပြီးနောက်။ ဒါဆိုရင် ဆက်လက် လေ့လာကြည့်ကြရအောင်လား။

၃။ ကျနိုင်းပြီး code\_line တွေက အထွေအထူး ရှင်းပြုရမလိုတဲ့ အတွက် main () function ကိုပဲ ဆက်လက် လေ့လာကြရအောင်လား။ main () function ထဲမှာတော့ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ primary class icon ကို object\_Name ic လို့ အမည်ပေးပြီးတော့ ခေါ်ယူထိက်တာပါပဲ။ လိုင်းနံပါတ် (၂၄) မှာ လေ့လာကြည့်ပါ။

က ဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ listing 3.17 ကို Analysis လုပ်ပြီး လေ့လာကြည့်တာ နားလည် သဘောပေါက သွားမှာပါ။ ဒီတော့ ဒီ code\_line တွေကို Notepad မှာသွားပြီး ရေးသားကြရအောင်။ Notepad ထဲမှာ code\_line တွေ ရေးသားပြီးသွားရင် icon.java ဆိုပြီးတော့ File\_Name ပေးပြီး save လုပ်ရပါမယ်။ File ကို Save လုပ်ရမယ့် path ကတော့



C:\j2sdk 1.4.1.01\bin ထဲမှာနော်။ ကဲ code\_line တွေကို ရေးသားပြီးတော့ save လုပ်ထားပဲကို Figure 3.21.9 အနေဖြင့် ရေးသားဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။ ဒီ program မှာ AWT package ကို ဖြေတဲ့ထားတဲ့အတွက်

```
import java.awt.*;
ဆိတ် code_line ဟာ မပါဝင်ပါဘူး။ ဒါကို သဘောပေါက် တယ်နော် သတိထားကြည့်ပါ။
```

```
icon.java - Notepad
File Edit Format View Help
import javax.swing.*;
public class icon extends JFrame{
    JButton[] buttons = new JButton[25];
    public icon(){
        super("Demonstrate the ImageIcon in JAVA");
        setSize(480, 380);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        ImageIcon ic = new ImageIcon("foodod.GIF");// this .GIF image is include in the C:\j2sdk1.4.1_01\Bin\foodod.GIF, if this image is not include you will use to another image.GIF type
        for(int i=0;i<25;i++){
            buttons[i] = new JButton(ic);
            pane.add(buttons[i]);
        }
        setContentPane(pane);
        show();
    }
    public static void main(String [] args){
        icon c = new icon();
    }
}
```

Figure 3.21.9 Write the program code\_line In Notepad

icon.java ဆိတ် File ကို C:\j2sdk 1.4.1\_01\bin အောက် မှာ save လုပ်ပြီးရင် Command Prompt <DOS> ထဲကို ဝင်ပြီး ရင် အောက်ပါ command နဲ့ compile လုပ်ပါ။ compile မလုပ် ခင်မှာ—

CD C:\j2sdk 1.4.1\_01\bin ↵

ဆိပြီးမှ DIR ကို change လိုက်ပါပြီး။ ပြီးမှ အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac icon.java ↵

အကယ်၍ compile လုပ်စဉ်မှာ Error တက်လာရင် လိုင်နဲ့ပါတ် ကို မှတ်ထားပြီး icon.java ဆိတ် Notepad ထဲမှာသွားပြီးတော့ ပြင်ပါ။ ပြီးရင် Save As ပြန်လုပ်ပြီးမှ အထက်ပါ command နဲ့ compile ပြန်လုပ်ပါ။

Note : ဒါကြောင့် ကျွန်ုတ်တော့ Notepad ရော command Prompt ကိုရော တစ်ပြိုင်နက် ဖွင့်ထားပြီးမှ compile & Run လုပ်စေခဲ့ပါတယ်။ ဒါမှသာ error တက်လာရင် အချိန်ကုန်သက်သာမှာ ဖြစ်ပါတယ်။

ကဲ error မရှိတော့ဘူးဘဲ ဆိုကြပါစို့။ အောက်ပါ command နဲ့ program ကို Run လုပ်ပါ။

java icon ↵

Program ကို compile & Run လုပ်ပဲကို Figure 3.21.10 အနေဖြင့် ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

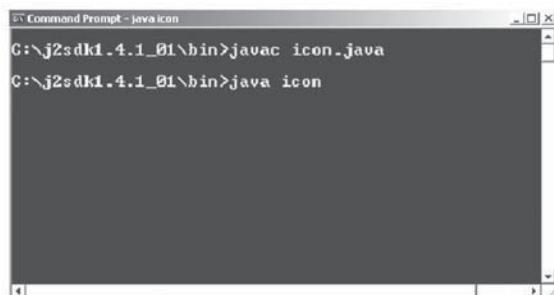


Figure 3.21.10 Compile & Run of the icon.java program

ကဲ Figure 3.21.10 မှာ ဖော်ပြထားတဲ့အတိုင်း compile & Run လုပ်လိုက်ရင်တော့ computer screen ရဲ့ ဘယ်ဘက် အပေါ်ထောင့်မှာ အောက်ပါအတိုင်း Demonstrate the ImageIcon in JAVA ဆိတ် Title Bar နဲ့ program ဟာ Run နေမှာ ပါပဲ။



Figure 3.21.11 Running the icon.java program

ကဲ ဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ ImageIcon တွေကို JAVA Language နဲ့ ဘယ်လို ဖန်တီးယူလို့ရတယ်ဆိတာကို သဘောပေါက်သွားလောက်ပါပြီ။ ဒါတော့ စာဖတ်သူ နှစ်သက်တဲ့ Image ပဲတွေကို File\_type ပြောင်းပြီးတော့ ထည့်သွင်း အသုံးပြုနိုင်ပါတယ်။ ဒါကြောင့် ဆက်လက်ပြီးတော့ 3.6 Using the Password Field In JAVA ဆိတ် သင်ခန်းစာကို ပို့ချေပေးထားပါတယ်။ ဆက်လက်လေ့လာကြပါစို့။

### 3.6 USING THE PASSWORD FIELD IN JAVA

Listing 3.18 Using the Password Field in JAVA

```

1. import javax.swing.*;
2.
3. public Class form extends javax.swing.JFrame {
4.
5.     J Text Field username = new JTextField (15) ;
6.     JPassword Field password = new JPasswordField (15);
7.     JTextArea comments = new JTextArea (4,15);
8.
9.     public form () {
10.        super ("Using the password Field in JAVA");
11.        setsize (300,200);
12.        setDefaultCloseOperation(JFrame.EXIT_ON_
13.            CLOSE);
14.        JPanel pane = new JPanel () ;
15.        JLabel usernamelabel = new JLabel
("UserName:");
16.        JLabel passwordlabel = new JLabel
("Password:");
17.        JLabel commentslabel = new JLabel
("Comments:");
18.        Comments.Setline Wrap (true) ;
19.        Comments .Set Wrap Style Word (true) ;
20.        pane.add (username label);
21.        pane.add (passwordlabel);
22.        pane.add (password);
23.        pane.add (commentslabel);
24.        pane.add (comments);
25.        set Content Pane (pane) ;
26.
27.        show () ;
28.    }
29.    public static void main (string [ ] args) {
30.        form f = new form () ;
31.    }
32. }
```

Listing 3.18 အနေနဲ့ ရေးပြထားတဲ့ program code\_line တွေကို အရင်ဆုံး Analysis လုပ်ကြည့်ကြရအောင်လား။ ပြီးမှ လက်တွေ တည်ဆောက်ကြည့်ကြတာပေါ့။

#### Analysis

၁။ Program ရဲ့အစမှာ swing components တွေကို ခေါ်ယူ အသုံးချဖိုင်စိုး Swing package ကို import လုပ်ပြီးတော့

ထည့်သွင်းပါတယ်။ ပြီးတော့ primary class form ဆိုတာ ကို ဖန်တီးပါတယ်။ ဖန်တီးရာမှာလည်း javax.swing.JFrame Class ကို extends လုပ်ထားတာနော်။ ဒီလို extends လုပ်ထားရာမှာ public class form extends javax.swing.JFrame { လို့ရေးခြင်းဟာ အတူတူပါပဲ။ ဒီတော့ အဲဒီ Class ကို ဆက်လက်ပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။ username လို့ object\_variable\_Name ပေးထားတဲ့ JTextField က တစ်ခု password လို့ object\_variable\_Name ပေးထားတဲ့ JPasswordField ကတစ်ခု comments လို့ object\_variable\_Name ပေးထားတဲ့ JTextArea က တစ်ခုကို ခေါ်ယူထည့်သွင်းပါတယ်။ အဲဒီ နေရာမှာ JTextField နဲ့ JTextArea ကတော့ အရှေ့က ပြီးခဲ့တဲ့ program တွေမှာ အသုံးပြုဖြီးသားပါ။ JPasswordField ကတော့ အခု သင်ခန်းစာများမှ အသုံးပြုဖူးတာပါ။ လိုင်းနဲ့ပါတ် (၉) မှာမှ form ( ) ဆိုတဲ့ constructor function ကို ရေးသားပါတယ်။ အဲဒီ constructor function ထဲမှာ Title Text အနေနဲ့ ပါဝင်ရမယ့် စာသားကို super ("Using the Password Field in JAVA"); လို့ရေးသားပါတယ်။ Super ( ) method ဟာ JFrame Class ထဲမှာ ပါဝင်ပါတယ်။ အရှေ့သင်ခန်းစာများလည်း ပါဝင်ပြီးသားပါ။ အဲဒီနောက် Frame ရဲ့ အရွယ်အစားကို SetSize(300,200); ဆိုပြီးမှ သတ်မှတ်ပါတယ်။ အဲဒီ Frame ကို EXIT\_ON\_CLOSE လုပ်နိုင်စိုးအတွက် SetDefault Close Operation ( ) method ကို ခေါ်ယူ အသုံးချပါတယ်။

J<sup>II</sup> လိုင်းနဲ့ပါတ် (၁၂) မှာမှ JPanel Class ကို object\_Name pane ဆိုတဲ့ အမည်ပေးပြီး ခေါ်ယူပါတယ်။ အဲဒီနောက် JLabel ကို အသုံးပြုပြီးတော့ Label Box သုံးခုကို ဖန်တီးပါတယ်။ သူတို့ရဲ့ Name တွေကတော့ Usernamelabel  
password label  
comments label လို့ ဖြစ်ကြပါတယ်။ ဒါတွေဟာ Object \_variable \_Name တွေအနေနဲ့ပါပဲ။ ဒီနာမည်တွေနဲ့ပဲ



ဆက်လက်ပြီးတော့ အသုံးချမှာဖော်။ အဲဒီ label Box တွေမှာပါဝင်စေချင်တဲ့ စာသားကို JLabel() method နဲ့ arguments နေရာမှာ တစ်ခါတည်း ထည့်သွင်း ရေးသားပြီးသားပါ။ ဒီအချက်ကို သတိထားပြီးတော့ စာဖတ်သူ လေ့လာကြည့်ပါ။ လိုင်းနံပါတ် (၁၇) နဲ့ (၁၈) မှာ တွေ့ရတာကတော့ comments လို့ အမည်ပေးထားတဲ့ JTextArea() ကို ကျွန်ုတ်တို့ အသုံးမချုပ်သေးတဲ့ – SetLineWrap() method နဲ့

setWrapStyleWord() method ရယ်ဆိုပြီး တွေ့ရမှာပါ။ ဒီတော့ ဒီ method နှစ်ခုဟာ JTextArea() နဲ့ပတ်သက်တဲ့ method တွေပဲ ဖြစ်ပါတယ်။ အဲဒီ method နဲ့ ပတ်သက်တာ တွေကို ရှင်းပြုနဲ့ လိုလာပြုပါ။ ကဲ ဒီတော့ Note: အနေနဲ့ ရေးသားထဲအချက်တွေကို သေသေချာချာ လေ့လာကြည့်ပါ။ API method အနေနဲ့ အလွန်အသုံးဝင်ပါတယ်။ ဒီတော့ စာဖတ်သူ အနေနဲ့လည်း Exam ဖြစ်ဆုံးရှုံး အထောက်အပံ့ရနိုင်အောင် မူရင်းအရေးအသားအတိုင်ပဲ ဖော်ပြပေးလိုက်ပါတယ်။

Note : • Set Line Wrap () method

Set Wrap Style Word () method

- The selline Wrap (boolean) method determines whether text will wrap to the next line when it reaches the far edge of the component. Call setlineWrap (true) to cause line wrapping to occur.
- The SetWrap Style Word (boolean) method determines What wraps to the next line either the current word (an argument of true) or the current character (false).

ကဲ အခု ရေးပြုခဲ့တဲ့ API method(j) ခုကို သေသေချာချာ လေ့လာပြီးသွားရင် code\_line တွေကို ဆက်လက်လေ့လာကြရအောင်လား။ လိုင်းနံပါတ် (၁၇) မှာ ရေးသားတဲ့

comments.setlinewrap(true);

method ကြောင့် JTextArea ထဲမှာ လက်ခံထားတဲ့ text တွေကို Wrapping လုပ်နိုင်ဖို့အတွက် လက်ရှိ Text အတကန်ပြီးတော့ စတင် မှတ်ယူပါတယ်။ လိုင်းနံပါတ် (၁၈) မှာ ရေးသားထားတဲ့

coments.setwrapStyleWord(true);

ဆိုတဲ့ code\_line ကတော့ လက်ရှိ Text နေရာကန်ပြီးတော့ ရှေ့လာမယ့် ရိုက်သွင်းတဲ့ current character ကို wrap လုပ်ဖို့ အတွက်ပါပဲ။

ဒီတော့ အထက်ပါ method() နှစ်ခုကို ခေါ်ယူပြီးတော့ code\_line မှာ ထည့်သွင်းလိုက်တဲ့အတွက် comments လို့

အမည်ပေးထားတဲ့ JTextArea မှာ ရိုက်သွင်းတဲ့ စာသား Text တွေဟာ သတ်မှတ်ထားတဲ့ limiting ဖြစ်လာရင် အောက်ကို ပြုတဲ့ ကျေလာမှပါ။ ဒဲဒေါဟာ Text Wrapping ရဲ့ သဘောတရားပါပဲ။ အကျယ်ချုံကြည့်ကြရအောင်လား။

JText Area (int , int); လို့ အသုံးပြုခဲ့တဲ့အခါမှာ



Specified Number  
Of Row

Specified Number of  
Columns

လို့ အရေးသင်ခန်းစာတွေမှာရေးပြီး ရှင်းပြုခဲ့ဖူးတယ်နော်။ အလွယ်မှတ်ရင်တော့

JTextArea ထဲမှာဆုံးမယ့် စာကြောင်း အရေအတွက်နဲ့ /JTextArea မှာဆုံးမယ့် စာလုံးအရေအတွက်လို့ပြောရင် မမှားပါဘူး။ ဒါဟာ JText Area တစ်ခုကို အကြမ်းပျော်အားဖြင့် ဖန်တီးတာပါပဲ။ ဒါပေမယ့် ဒီဖန်တီးလိုက်တဲ့ JTextArea မှာ သတ်မှတ်ထားတဲ့ အရေအတွက်ထက် စာလုံးရောကို ပိုပြီးတော့ ရိုက်သွင်းမိရင် JTextArea ဟာ အလွယ်အလေ့တော်တော် အမှန်တကယ်တော့ JTextArea ကို Resizeable လုပ်ပေးပါတယ်။ အမှန်တကယ်တော့ JTextArea ကို အသုံးပြုမယ်ဆိုရင် Scrolling Panes လို့ခေါ်တဲ့ JScrollPane() method ကိုပါ ခေါ်ယူအသုံးချုပ်ပါတယ်။

ဥပမာ JScrollPane (VERTICAL\_SCROLLBAR\_ ALWAYS);

ဆိုတဲ့ code\_line မျိုးနဲ့ JTextArea မှာ VERTICAL & HORIZONTAL Scroll Pane တွေကို ထည့်သွင်းပေးရပါမယ်။ ဒါဆိုရင်တော့ လိုင်းနံပါတ် (၁၇)၊ (၁၈) မှာ ရေးသားတဲ့

setlineWrap() method နဲ့

setWrapStyleWord() method တွေကို ထည့်သွင်းအသုံးပြုစရာမလိုဘူးပေါ့နော်။ ဒါပေမယ့် အခု ဥပမာ ရေးပြထားတဲ့ code\_line ဟာ အမှန်မဟုတ်ပါဘူးနော်။ ဥပမာပေးပြီး ရေးပြထားတာပါ။ အခုလက်ရှိ listing 3.18 မှာ အခု ရှင်းပြနေတဲ့ JScrollPane တွေကို ထည့်သွင်းအသုံးပြုမထားတဲ့အတွက်ကြောင့် အပေါ်ကရှင်းပြနေတဲ့ method(j) ခုကို ထည့်သွင်း အသုံးပြုရတာပါပဲ။

ဒီ method(j) ခုသာ မပါဝင်ခဲ့ရင် စာလုံးရော် သတ်မှတ်ထားတဲ့ limit ထက်ပို့ရင် JTextArea ကိုရော့၊ Form\_Window ကိုရော့ အလွယ်အလေ့တော်တော် Resizeable လုပ်ပြီး ချုံးမေးမှာပါ။ ဒီအခါမှာ သတ်မှတ်ထားတဲ့ Form\_Windows ပုံစံဟာ ပျက်ပြုပေါ့နော်။



Note : စာဖတ်သူဟာ setlinewrap ( ) method / setwrap Style Word ( ) method နှစ်ခုကို ဖြတ်ထားပြီးတော့ compile & Run လုပ်ကြည့်ပါ။ စာလုံးရော့ စာကြောင်း ရော့များရင်များသလောက် အလိုအလျောက် ချုံသွားပြီး တော့ လက်ခံနေပါလိမ့်မယ်။ တကယ်လို့ ဒီ method ( ) တွေကို မသုံးချင်ရင်တော့ JScrollPane ကို ခေါ်သုံးရမှာပေါ့နော်။ စာဖတ်သူအနေနဲ့ စမ်းကြည့်ပါ။

၃။ ကဲ ဆက်ပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။ အဲဒီ လိုင်း နံပါတ် (၁၇)၊ (၁၈) က Text Wrapping method နှစ်ခု ကြောင့် JTextArea ကို သတ်မှတ်ထားတဲ့ specified of Number Rows (4) Rows [စာကြောင်းအရေတွက်(၄) ကြောင်း] specified Number of columns (20) [စာလုံးရော့ အရေအတွက် (၂၀) လုံး တိတိ] ကို လက်ခံမှုပါ ဖြစ်ပါတယ်။ သတ်မှတ်ထားတဲ့ စာလုံးရော့ (၂၀) ထက် ပိုလာရင် နောက် တစ်ကြောင်းကို အလိုအလျောက် Enter ↴ key နှင့်စရာ မလိုပါဘူးနဲ့ ဆင်းသွားမှာပါ။ ဒါဟာ Text wrapping method တွေရဲ့ သဘောတရားပါပဲ။  
ကဲ code\_line နံပါတ် (၁၉) (၂၀) မှာတော့ pane လို့ Object\_Name ပေးထားတဲ့ JPanel Class ထဲက add () method ကို အသုံးချုပြီးတော့ panel ထဲမှာ add လုပ်ပါတယ်။ ဒါကြောင့်

```
pane.add(username.label); // add the JPanel
pane.add(username); // add the JTextField
```

လို့ ရေးသားတာပေါ့။ ဒါဟာ JTextField တစ်ခုနဲ့ JLabel တစ်ခုကို form ပေါ်မှာ တပ်ဆင်လိုက်ပြီနော်။

Note : Swing Containers ,including frames, windows, applets, and dialog boxes, do not allow components to be added in this manner. Those containers are broken down into panes, a kind of container within a container. Ordinarily , components are added to the container's content pane.

အထက်ပါ နည်းလမ်းအတိုင်းပါ JPasswordField နဲ့ JLabel တစ်ခုစီကိုလည်းကောင်း၊ JTextField နဲ့ JLabel တစ်ခုစီကိုပါ အထက်ပါ code\_line တွေ အတိုင်း ထည့်သွင်းပါတယ်။ လိုင်းနံပါတ် (၂၂) ကနေ (၂၅)အထိပါပဲ။ လေ့လာကြည့်ပါ။ အဂွယ်တကူ သဘော ပေါက်နိုင်ပါတယ်။

အဲဒီနောက်မှာတော့ show () method ကို ခေါ်ယူ အသုံး ချပါတယ်။ Frame ကို Display ပြနိုင်ဖို့အတွက်ပေါ့နော်။ ဒီနောက်မှာ show () method ကို အသုံးမပြုချင် ရင် setvisible (true) method ကို အသုံးပြုရင်လည်း ရုပါတယ်။ အချို့ program တွေမှာ ကျွန်ုတ် အသုံး ပြုခဲ့ဖူးပါတယ်။

Note : အချို့သော Swing container ထဲမှာပါဝင်တဲ့ API method တွေဟာ မရှင်းပြုရသေးဘဲ ရှိနေပါသေးတယ်။ အဲဒါတွေနဲ့ ပတ်သက်တာတွေကိုတော့ Appendix of Swing & AWT components လို့ အမည်ပေးပြီး တော့ ပို့ချေပေးထားပါတယ်။ သီးသန့်သင်ခန်းစာအနေနဲ့ ပါပဲ။ လက်ရှိ Swing & AWT နဲ့ ပတ်သက်တဲ့ သင် ခန်းစာတွေအပြီးမှာပါ။ အဲဒီအခါမှာ လက်ရှိ အသုံးပြုပြီးတော့ နားမလည်တဲ့ API method တွေကိုပါ compare လုပ်ပြီးတော့ ရှင်းပြုပေးထားပါတယ်။ အဲဒီနောက် လိုင်းနံပါတ် (၂၉) မှာတော့ main () function ကို ရေးသားထားပါတယ်။ အဲဒီ main () function ထဲမှာမှ အပေါ်မှာ ရေးသားခဲ့တဲ့ form ဆိုတဲ့ class ကို Object\_Name f လို့ ပေးပြီးတော့ ခေါ်ယူ လိုက်ပါတယ်။ ဒီအခါမှာ form () ဆိုတဲ့ class ရဲ့ constructor function ဟာ အလိုလိုဝင်လာပြီပေါ့။ အဲဒီအခါမှာ form () ဆိုတဲ့ constructor ထဲမှာ ရေးသားထားတဲ့ (ကျွန်ုတ်တို့ဖြစ်ချင်တဲ့ code\_line တွေ) က အလုပ်လုပ်ပြီပေါ့။ လိုင်းနံပါတ် (၂၀) မှာပါ။ ကဲကျွန်ုတ် brace } တွေကတော့ main () function နဲ့ primary class ရဲ့ close brace } တွေပါပဲ။

ဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ listing 3.18 က code-line တွေကို Analysis လုပ်တာ အလွယ်တကူ နားလည် သဘော ပေါက်သွားပြီပေါ့နော်။ ဒီတော့ အဲဒီ code\_line တွေကို program အနေနဲ့ လက်တွေတည်ဆောက်ကြရအောင်။ Notepad program ထဲကိုသွားပြီးတော့ အဲဒီ code\_line တွေကို ရေးသားပါ။

အဲဒီလို့ code\_line တွေကို ရေးသားပြီးရှင်တော့ File → Save ကိုတောင်းပါ။ Save လုပ်ရာမှာ C:\j2sdk 1.4.1\_01\bin ဆိုတဲ့ folder အောက်မှာ save လုပ်ရမှာနော်။ File\_Name ကို တော့ form.java လို့ အမည်ပေးရပါမယ်။ အဲဒီလို့ အမည်ပေးပြီး တော့ save လုပ်ထားပဲကို Figure 3.22 အနေနဲ့ ရေးပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

```

form.java - Notepad
File Edit Format View Help
Import javax.swing.*;
Import java.awt.*;

public class form extends javax.swing.JFrame{
    JTextField username = new JTextField(20);
    JPasswordField password = new JPasswordField(20);
    JTextArea comments = new JTextArea(5,20);

    public form(){
        super("using the Password Field in JAVA ");
        setSize(350,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        JLabel usernamelabel = new JLabel("UserName : ");
        JLabel passwordlabel = new JLabel("Password : ");
        JLabel commentslabel = new JLabel("Comments : ");
        comments.setLineWrap(true);
        comments.setEditable(true);
        pane.add(usernamelabel);
        pane.add(username);
        pane.add(passwordlabel);
        pane.add(password);
        pane.add(commentslabel);
        pane.add(comments);
        setContentPane(pane);
        show();
    }
    public static void main(String[] args){
        form f = new form();
    }
}

```

Figure 3.22 Write the program In the Notepad

ကဲအပေါ်မှာရေးပြုခဲ့တဲ့ Figure 3.22 ကတော့ Notepad မှာ Code\_Line တွေ ရေးသားပြီးတော့ save လုပ်ထားပုံပါပဲ။ ဒီတော့ File ကို Save လုပ်ပြီးရင် command prompt ထက် ဝင်ပြီးတော့ compile & Run လုပ်ကြရအောင်လား။ အရင်ဆုံး Directory path ကို ပြောင်းလိုက်ပါပြီး။ အောက်ပါ command နဲ့ပြောင်းပါ။

CD C:\j2sdk 1.4.1\_01\bin ↵

ဒီတော့ j2sdk 1.4.1\_01 ရဲ့ directory အောက်ကိုရောက်သွားရင် CLS ↴ command နဲ့ screen ကို clearလုပ်လိုက်ပါပြီး။ ပြီးရင် အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac form.java ↵

compile လုပ်ပြီးတော့ error မရှိခဲ့ရင်တော့ အောက်ပါ command နဲ့ ဆက်လက်ပြီးတော့ Run လုပ်ပါ။

java form ↵

အောက်ပါအတိုင်း compile & Run လုပ်ပုံကို Figure 3.22.1 အနေနဲ့ ရေးပြေားထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 3.22.1 Compile & Run of the form.java program

အကယ်၍မျှား Compile လုပ်စဉ်မှာ error တက်လာခဲ့ရင်လည်း Notepad ထဲက code\_line တွေကိုဝင်ပြီးတော့ error Line\_No အတိုင်းပြင်ပြီးတော့ Save As လုပ်ပါ။ ပြီးရင်တော့ Compile & Run လုပ်ပါ။ ဒီနည်းနဲ့ပဲ error ကင်းအောင် ကြိုးစား တည်ဆောက်ရမှာပါပဲ။ ကဲ error ကင်းသွားလို့ အမှား မရှိတော့ရင် Run လုပ်ပဲအခါမှာ အောက်ပါအတိုင်း output ထွက်လာမှာပါပဲ။ Figure 3.22.2 အနေနဲ့ ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

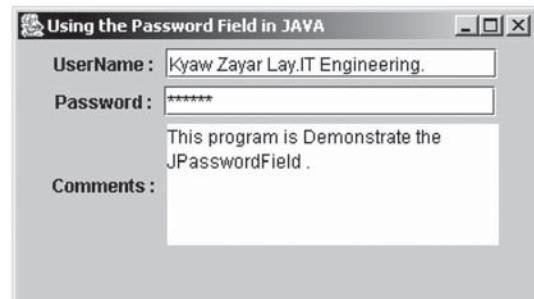


Figure 3.22.2 Running the form.java program

ကဲခါဆိုရင်တော့ JPasswordField ကို အသုံးချ ရေးသားထားတဲ့ program ထဲလေးကို Demonstrate လုပ်ပြလို့ ပြီးပါပြီ။ စာဖတ်သူအနေနဲ့ အရှေ့ကသင်ခန်းစာတွေနဲ့ ပေါင်းစပ်ပြီးတော့ အသုံးပြု ဖန်တီးကြည့်ပါ။ ဆက်လက်ပြီးတော့ swing & AWT components တွေဟာ များပြားလှတာကြောင့် sub-Index အနေနဲ့ ရင်းပြရတာဟာ များပြားလှတာကြောင့် Chapter [4] အဖြစ် ဆက်လက်ပြီးတော့ သင်တြား ပို့ချေပေးထားပါတယ်။ Swing & AWT components တွေတဲ့ ပါဝင်တဲ့ API method တွေကိုလည်း Appendix အဖြစ် ဆက်လက်ပို့ချေပေးမှာပါ။ ဒါကြောင့် ဆက်လက်ပြီးတော့ Chapter [4] ကို ဆက်လက်လေ့လာကြည့်ပါ။

KYAW ZAYAR LAY [ IT ENGINEERING ]



# JAVA DEVELOPER GUIDE

## Chapter (4)

### 4.1 CREATE A NEW TOOLBAR WITH JAVA LANGUAGE

Toolbar တစ်ခုကို ဖန်တီးမယ် ဆိုကြပါစို့။ JToolBar Class ဟာ Swing Package ထဲမှာပါဝင်ပြီးသားပါ။ JToolBar Class ဟာ အုပ်စုတဲ့ ထားတဲ့ several components တွေကို row (or) column အနေနဲ့ ဖန်တီးနိုင်ပါတယ်။ အများအစားဖြင့်တော့ Toolbar တွေကို ဖန်တီးရမယ့် JButton တွေကို အုပ်စုပြီးတော့ အသုံးချလေ့ ရှိပါတယ်။ Toolbar လို့ဆိုရမယ့် စာဖတ်သူဟာ သာမန် user တစ်ယောက်ပဲ ဖြစ်နော်ပေါ်တော့ အသုံးချဖြုံးသားပဲ ဖြစ်မှာပါ။ ဥပမာ- Microsoft Word, Netscape Navigator, or Lotus Wordpro အစရိတ်တွေမှာ toolbar တွေဟာ ပါဝင်ပြီးသားပါ။ ဒီနေ့တော် သာမန် user တစ်ယောက်အဖို့မှာတော် Toolbar တွေကို Family ဖြစ်ပြီးသားပါ။ Microsoft Word, Microsoft Excel မှာပဲ လေ့လာကြည့်လေ။ Pull-Down Menu ကနေ ကျလာမယ့် command ဆိုကို သွားမယ့်အစား alternative shortcut တွေပဲ အသုံးပြုမလား၊ ဒါမှုမဟုတ် Toolbar ပေါ်က shortcut button တွေကိုပဲ အသုံးပြုမလား၊ စသဖို့ အလွယ်တကူ အသုံးဝင်အောင် user interface အနေနဲ့ ဖန်တီးထားလို့ ရှိပါတယ်။

Toolbars တွေကို ဖန်တီးရမယ့် default အားဖြင့်တော့ Horizontal\_Toolbar\_Type အနေနဲ့ပဲ ဖန်တီးကြပါတယ်။ ဒါပေမယ့် လက်ရှိနာမည်ကြီး Application Software တွေမှာ လည်း vertical type toolbar တွေလည်း တွေ့မြင်ရပါလိမ့်မယ်။ ဒါဟာ ပြဿနာမရှိပါဘူး။ လိုချင်တဲ့ type ကို ရရှိနိုင်း။ Horizontal or Veritcal Class variables တွေဟာ JToolBar Class ရဲ့ Swing Constants interface ထဲမှာ ပါဝင်ပြီးသားပါ။ JToolBar Class ထဲမှာပါဝင်တဲ့ constructor methods တွေကို လေ့လာကြည့်ရအောင်။

- JToolBar() - Creates a new toolbar

- JToolBar(int) - Creates a new toolbar with the specified orientation

ဒဲ ဒီ construction method တွေကို လေ့လာကြည့်မယ် ဆိုရင် JToolBar Class ထဲမှာ ပါဝင်တဲ့ construction function တွေဟာ overloading ပုံစံနဲ့ ရေးသားထားတာကို တွေ့ရမယ့်ပါ။

Note : Overloading method အကြောင်း ပါဝင်လာလို့ ရှုပ်ထွေးမသွားပါနဲ့။ C & C++ နယ်ပယ်က programmer တွေကတော့ over loading ရဲ့ သဘောတရားကို သဘောပေါက်မှာပါ။ အလွယ်ပြောရရင် function ()\_Name တွေ တူညီပြီးတော့ parameter arguments type တွေ ကွဲပြားခြားနားတာကို အခြေခြားပြီးတော့ တွက်ချက်အသုံးချရမယ့် သက်ဆိုင်ရာ Function ()ကပဲ ဝင်ရောက်အလုပ်လုပ်တာလို့ပဲ အလွယ်မှတ်ယူထားပါ။ အမှန်တကယ်က တော့ overloading ရဲ့ သဘောတရားက ဒုံးထက် အများကြီး ကျယ်ပြန့်ပါတယ်။

ပြီးခဲ့တဲ့ chapter (1) & Chapter (2) မှာ လေ့လာကြည့်ပါ။ အဲဒီ Toolbar တွေပေါ်မှာ Components တွေကို ထည့်သွင်းချင်တယ်ဆိုရင်တော့ ရွှေ့သင်ခန်းစာတွေမှာ အသုံးချခဲ့တဲ့ add (Object) method ကို အသုံးချရမယ့်ပါပဲ။ Program အများစုံမှာ ဖန်တီးထားတဲ့ Toolbar တွေကို အလွယ်တကူ move လုပ်လို့ ရှိပါတယ်။ အဲဒီလို့ move လုပ်ရမယ့် အလွယ်တကူ ဈွှေ့ပြေားနိုင်တာကို dockable toolbars လို့ ခေါ်ပေါ်ပါတယ်။ Swing Package ကို အသုံးပြုပြီး ဖန်တီးထားတဲ့ Toolbar တွေဟာ dockable toolbars တွေပါပဲ။ ဒါကြောင့်

- Swing toolbars can also be docked into a new window, separate from the original.

က ဒီတော့ Toolbar နဲ့ပတ်သက်တဲ့ အချက်အလက်တွေကို  
လေ့လာပြီးသွားပြီဆိုရင်တော့ dockable toolbar တစ်ခုပါဝင်တဲ့  
Program ငယ်လေးတစ်ခုကို ဖုန်းတိုး တည်ဆောက်ကြည့်ကြရ  
အောင်။ Listing 4.1 အနေနဲ့ ရေးသားထားတဲ့ The Full Text  
ofToolBar.java ဆိုတဲ့ Program ကို အရင်ဆုံး လေ့လာလိုက်  
ကြရအောင်လေး။ က လွှေလာကြပါစိုး။

**Listing 4.1 The Full of ToolBar . java**

- ```
1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. public class Toolbar extends JFrame{
6.
7.     public class ToolBar( ) { // constructor function
8.         super ("Testing the JAVA ToolBar");
9.         setDefaultCloseOperation(JFrame.EXIT_ON_
CLOSE);
10.        ImageIcon images1 = new ImageIcon ("button
1.gif");
11.        JButton button 1 = new JButton (image1);
12.        ImageIcon image 2 = new ImageIcon ("button
2.gif");
13.        JButton button 2= new JButton (image 2);
14.        ImageIcon image 3 = new ImageIcon("button3.
gif");
15.        JButton button 3 = new JButton (image3);
16.        JToolBar bar = new JToolBar ( );
17.        bar.add (button 1);
18.        bar.add (button 2);
19.        bar.add (button 3);
```

```
20. JTextArea edit = new JTextArea (8,40);
21. JScrollPane scroll = new JScrollPane (edit);
22. JPanel pane = new JPanel ( );
23. BorderLayout bord = new BorderLayout( );
24. Pane.setLayout (bord);
25. Pane.add ("North", bar);
25. Pane.add ("enter", scroll);
27.
28. SetContentPane (pane);
29. } // Close brace of the constructor function ( )
30.
31. Public static void main (String [ ] arguments) {
32. ToolBar frame = new ToolBar ( );
33. Frame.pack ( );
34. Frame.setVisible (true);
35. } // close brace fn main function
36. }
```

ToolBar Java ဆိုတဲ့ Program ဟာ code\_line (36) ကြောင်း ပါဝင်တဲ့ program ငယ်လေးပါပဲ။ Notepad ထဲမှာ ရေးသားရာမှာတော့ line No: တွေ ထည့်သွင်းရေးသားစရာ မလိုပါဘူး။ ကဲ ဒီ program ကို Analysis လုပ်ဖြီးတော့ လေ့လာကြရအောင်လား။

### Analysis

၁။ `java.awt.*;java.awt.event.*;java.swing.*;`; ဆိတ် package file (၃) ခုကို import လုပ်ပြီး ခေါ်ယူထားပါတယ်။ လိုင်းနံပါတ် (၁) (၂) (၃) မှာ Declare လုပ်ထားတာကို တွေ့ရမှာပါ။ လိုင်းနံပါတ် (၅)မှာတော့ `JFrame` Class ကို extends လုပ်ပြီးတော့ `ToolBar` ဆိတ် primary class ကို ရေးသားပါတယ်။ လိုင်းနံပါတ် (၇) မှာတော့ `ToolBar` class ပဲ၏ constructor function ()ကို ရေးသားပါတယ်။ အခို့ constructor function ထဲမှာမှ `super()` method ကို အသုံးချဖြီးတော့ `JFrame`.`windows` နဲ့ `title_bar` text ကို ရေးသားလေးပါတယ်။ " " Double quote နှစ်ခု ကြေားမှာ ရေးသားထားတဲ့ စာသားကို တွေ့ရမှာပါ။ လိုင်းနံပါတ် (၉)မှာတော့ `SetDefaultCloseOperation()` method ကို အသုံးချဖြီးတော့ `SetDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);` ဆိုပြီးတော့ ရေးသားထားပါတယ်။ ဒါဟာ `JFrame` Windows ကို `exit` လုပ်နိုင်ဖို့အတွက် ထည့်သွင်း ရေးသားထားတာပါပဲ။ အရှေ့ chapter (၃) နဲ့ standard dialog box သင်ခန်းစာတွေမှာ ဒီ code line ကို အသုံးမပြုတဲ့ program တွေ ရေးပြုဖူးပါတယ်။ ပြန်လှန် လေ့လာကြည့်ပါ။ Compile လုပ်နေတာ Run နေတာကိုပါ ရုပ်ပစ်လိုက်မှုသာလျှင် program\_exit ဖြစ်သွားမှာပါ။

**Note :** အကယ်၍ စာဖတ်သူဟာ အထက်ပါ code\_line ကို  
အသုံးမပြုချင်လျှင် [SetDefault Close Operation  
( ) method ကို မသုံးချင်လျှင်] အောက်ပါ command  
code-line ကို ပြောင်းလဲ အသုံးပြုကြည့်ပါလား။  
System.exit(0);

ဒါဟာ javax.swing.JFrame; package ကို ခေါ်ပူး  
အသုံးချုပြီးတော့ Java command\_line windows  
တစ်ခု ဖန်တီးမယ်ဆိုရင်တော့ အကျိုးဝင်ပါလိမ့်မယ်။ အဲဒီ  
အပိုင်းနဲ့ပတ်သက်ပြီး 4.6 Developing a FrameWork  
ဆိုပြီးမှ ပို့ချေပေးထားပါတယ်။

J) အဲဒီနောက် လိုင်းနံပါတ် (၁၀)မှ ImageIcon () method  
ကို ခေါ်ပူး button.gif ဆိုတဲ့ Image file နဲ့ တွဲဖက်ပြီး  
အသုံးချေထားပါတယ်။

**Note :** ဒီ program မှာ button 1. gif/ button2. gif/  
button3. gif ဆိုတဲ့ Image file တွေဟာ တကယ်  
မရှိပါဘူး။ C:\j2sdk1.4.1\_01\Bin ဆိုတဲ့ folder ထဲမှာ  
ဘာပုံမျှ ထည့်သွင်းမထားတဲ့ အတွက် Program Run  
လုပ်တဲ့ အခါမှာ ToolBar ပေါ်မှာ ဘာပုံမျှ ထွက်လာမှာ  
မဟတ်ပါဘူး။ စာဖတ်သူဂုဏ်ချင်တဲ့ ImageIcon တွေကို  
.gif Type ပြောင်းပြီးတော့ C:\j2sdk1.4.1\_01\Bin  
အောက်မှာ သွားထားပြီးတော့ program code\_Line  
တွေ ရေးရာမှာ Image File\_Name ကို ထည့်ရေးပါ  
မယ်။ File\_type ပြောင်းနည်းကိုလည်း ပြီးခဲ့တဲ့ အရှေ့  
သင်ခန်းစာမှာ သင်ကြား ပို့ချေပေးပြီးသားပါ။ ဒါကြောင့်  
အလွယ်တော်က မိမိနှစ်သက်တာကို ထည့်သွင်း ဖန်တီးနိုင်  
ကြမှာပါပဲ။ ကျွန်ုတ်ကတော့ ဘယ် ImageIcon ပုံကိုမျှ ထည့်  
သွင်းထား ခြင်းမရှိပါဘူး။ လိုင်းနံပါတ် (၁၁)မှာကတော့  
JButton ( ) method ကို အသုံးချုပြီးတော့ JButton  
တစ်ခုကို ဖန်တီးပါတယ်။ အဲဒီ JButton မှာ ပါဝင်ရမယ့်  
IconImage ကတော့ အပေါ်မှာ ဖန်တီးရယူခဲ့တဲ့  
ImageIcon () method ထဲက Image ပါပဲ။ ဒီနည်း  
လမ်းအတိုင်း လိုင်းနံပါတ် (၁၂)ကနေ (၁၃)အထိ ဖန်တီး  
ပြီးတော့ ImageIcon က (၃)ခု ရယ်၊ JButton ကလည်း  
(၃)ခု ရယ်ကို ဖန်တီးပါတယ်။

၃) လိုင်းနံပါတ် (၁၆)မှာတော့ JToolBar Class ကို Object  
\_Name bar ဆိုတဲ့ အမည်ပေးပြီးတော့ ခေါ်ပူး အသုံးချပါ  
တယ်။ အဲဒီ JTool-Bar ထဲမှာမ အပေါ်မှာ ဖန်တီးခဲ့တဲ့  
button1/ button2/ button3 ဆိုတဲ့ JButton (၃)ခုကို  
ခေါ်ပူး ထည့်သွင်းပါတယ်။ ဒီအခါမှာ JButton တွေကို

ခေါ်ယူရင် ImageIcon တွေကိုပါ ခေါ်ယူပြီးသား ဖြစ်သွား  
ပြီပေါ်နော်။ ဒါကြောင့်

```
bar.add (button 1);
bar.add (button2);
bar.add (button3); လို့ရေးသားထားတာပါပဲ။ လိုင်း
နံပါတ် (၂၀)မှာတော့ JText-Area Class ကို object name
edit လို့ အမည်ပေးပြီးတော့ ဖန်တီးပါတယ်။ ဒါကြောင့်
```

```
JTextArea edit = new JTextArea (8,40);
လို့ ရေးသားထားပါတယ်။ JTextArea ( ) ရဲ့ arguments
မှာ ပါဝင်တာကတော့ စာလုံးရေး (၄၀)ဆုံးတဲ့ စာကြောင်း (၈)
ကြောင်းလို့ limit လုပ်ထားတာပါပဲ။ ဟုတ်ပြေနော်။ လိုင်း
နံပါတ် (၂၂)မှာကတော့ JScrollPane Class ကို အသုံးချုပြီး
တော့ JTextArea [Object_Name_Edit] ပေါ်မှာ သတ်
မှတ်ထားတဲ့ arguments ထက်ကျက်လာရင် Scrolling လုပ်
နိုင်ဖို့ [ScrollBar တွေ ပါဝင်လာဖို့] ရေးသား ထည့်သွင်းပေး
ပါတယ်။ ဒါကြောင့်လည်း JPanel Class ကိုပါ ခေါ်ပူး
အသုံးချေထားတာကိုလို့ လိုင်းနံပါတ် (၂၂) မှာ တွေ့ခြင်ရမှာပါပဲ။
အဲဒီနောက်မှာ JToolBar ကို တပ်ဆင်ဖို့အတွက် Border-
Layout ကိုခေါ်ယူပြီးတော့ အသုံးချပါတယ်။ ဒီအချက်နဲ့ပတ်
သက်တာကိုတော့ အပေါ်မှာ ရင်ပြုခြုံပြီးသားပါ။ အဲဒီနောက်
```

```
Pane.setLayout (bord):ဆိုပြီးတော့ JPanel Class
ကို မိမိပြီးတော့ BorderLayout ကို တပ်ဆင်ပါတယ်။ ဒီတော့
JToolBar ကိုပါ ပါဝင်လာစေပါပြီ။ ဒါကြောင့်
```

```
pane.add ("North.", bar);
pane.add ("Center", scroll); လို့ ရေးသားထားတာ
ပေါ်။ ပြီးတော့ လိုင်းနံပါတ် (၂၃) မှာတော့ အပေါ်က calling
လုပ်ထားတဲ့ object တွေ အကုန်ပါဝင်နိုင်စေဖို့အတွက်
SetContentPane (pane);
```

```
ဆိုပြီးမှ ခေါ်ပူး ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၂၄)
ကတော့ Constructor Function ToolBar () ရဲ့ Close
brace ပါပဲ။ အဲဒီနောက် blank_line_space ချုန်ထားပါ
တယ်။ လိုင်းနံပါတ် (၂၅) မှာတော့ main () function ကို
ရေးသားထားပါတယ်။ Main ( ) Function ထဲမှာတော့
အပေါ်မှာ ဖန်တီးခဲ့တဲ့ ToolBar Class ကို Frame လို့
Object_Name ပေးပြီး ခေါ်ယူ အသုံးချပါတယ်။ ဒီအခါမှာ
constructor. function () ဟာ အဓိလို့ ဝင်ရောက်လာပါ
ပြီ။ အဲဒီထဲက calling method ( ) တွေအားလုံး ပါဝင်
လာအောင်ရယ် pack () method ကြောင့် ဘာတွေဖြစ်လာ
သလဲဆိုရင် Sets its size to the preferred size of its
components လို့ပြောရရင်လည်းဖြစ်ပါတယ်။ ပြီးတော့
object ကို visible ဖြစ်အောင်
```

Frame.setVisible (tsuo);  
လို့ ရေးသားထားတာပဲ ဖြစ်ပါတယ်။ အဲဒီနောက် main () function ရဲ့ class brace } ကို လိုင်းနံပါတ် (၃၅)မှာ ရေးသားထားပါတယ်။ ToolBar Class ရဲ့ close brace } ကိုတော့ လိုင်းနံပါတ် (၃၆)မှာ လာပိတ်ထားပါတယ်။ သေ သေချာချာ ဖတ်ရှုပြီးတော့ လေ့လာကြည့်ပါ။ ဒါဆိုရင်တော့ ToolBar တစ်ခု ဖန်တီးရယူတဲ့ နည်းလမ်းကို သဘောပါက် သွားမှတ်။ ကဲ code\_line တွေကို နားလည်သဘောပါက် သွားရင်တော့ ဒီ code\_line တွေကို NotePad ထဲမှာ သွားရောက်ပြီးတော့ ရေးသားကြရအောင်လား။ Code\_Line တွေ ရေးသားထားပါတယ်။ Figure 4.1 အနေနဲ့ ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



```

ToolBar.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class ToolBar extends JFrame {
    public ToolBar() {
        super("Testing the JAVA Toolbar");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        ImageIcon image1 = new ImageIcon("image1");
        JButton button1 = new JButton(image1);
        ImageIcon image2 = new ImageIcon("image2");
        JButton button2 = new JButton(image2);
        ImageIcon image3 = new ImageIcon("image3");
        JButton button3 = new JButton(image3);
        JToolBar bar = new JToolBar();
        bar.add(button1);
        bar.add(button2);
        bar.add(button3);
        JTextArea edit = new JTextArea(8,40);
        JScrollPane scroll = new JScrollPane(edit);
        JPanel pane = new JPanel();
        BorderLayout bord = new BorderLayout();
        pane.setLayout(bord);
        pane.add("North",bar);
        pane.add("Center",scroll);
        setContentPane(pane);
    }

    public static void main(String [] args) {
        ToolBar tb = new ToolBar();
        tb.pack();
        tb.setVisible(true);
    }
}

```

Figure 4.1 Write the program code\_line in the NotePad

ကဲ အပေါ်က Figure 4.1 အတိုင်း code line တွေကို ရေးသားပြီးသွားရင် C:\j2sdk 1.4.1\_01\bin ထဲကို သွားပြီး save လုပ်ပေးရပါမယ်။ ကဲ ပြီးရင်တော့ Command Line Windows (DOS) ကို ဖွံ့ဖြိုးတော့ အောက်ပါ command နဲ့ Directory path ကို ပြောင်းပါ။

cd C:\j2sdk 1.4.1\_01\bin ↵

ပြီးရင်တော့ အောက်ပါ command နဲ့ compile လုပ်ပါ။ Error ရှိရင်တော့ code\_line ကို ပြန်ပြင်ရမှာပါပဲ။

### javac.ToolBar.java ↵

error ကို တွေ့ရတဲ့ code line ပြန်ပြင်ပြီးရင် notepad ထဲက Toolbar .java ဆိုတဲ့ file ကို ပြန်ပြီးတော့ save as လုပ်ပေးရပါမယ်။ အကယ်၍ compile error မရှိခဲ့ရင်တော့ အောက်ပါ command line နဲ့ run လုပ်လိုက်ပါ။

### java Toolbar ↵

အဲဒီအခါကျရင်တော့ ကျွန်ုင်တော်တို့ ရေးသားထားတဲ့ Toolbar.java ဆိုတဲ့ program ဟာ run လုပ်ပြီးတော့ computer screen ပေါ်မှာ လာ၍ ဖော်ပြုပေးလာမှာပါပဲ။ Compile & Run လုပ်ပုဂ္ဂိုလ် Figure 4.2 အနေနဲ့ ရေးသား ဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

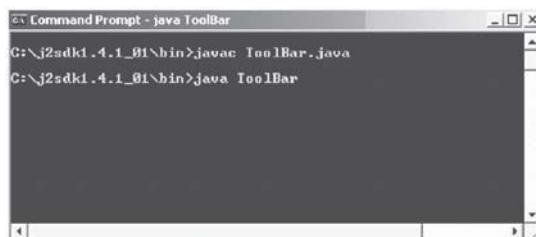


Figure 4.2 Compile & Run of the ToolBar.java

ကဲ Compile & Run လုပ်ပုဂ္ဂိုလ် လေ့လာပြီးတော့ Figure 4.3 မှာ ဖော်ပြထားတဲ့ program run နေပုက္ပါလေ့လာကြည့်ပါတော့။ JButton တွေပေါ်မှာတော့ IconImage တွေ ပါဝင်မှာ မဟုတ်ဘူး နော်။ စာဖတ်သူထည့်သွင်းချင်တဲ့ IconImage ကို .GIF File\_type ပြောင်းပြီးတော့ C:\j2sdk 1.4.1\_01\bin ထဲမှာ ထားပြီးတော့ ခေါ်ယူ ထည့်သွင်းလို့ရပါတယ်။ လေ့လာကြည့်ပါ။



Figure 4.3 Run the program, ToolBar.java

ကဲ Java Language နဲ့ ToolBar တွေကို ဖန်တီးတာ သွားပြီးတော့ ဆက်နှုံဖန်တဲ့ BorderLayout တွေအကြောင်းကို လေ့လာကြရအောင်လား။ 4.2 Using the Basic Interface Layout ဆိုပြီးမှ ရေးသား ပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်လိုက်ပါတော့။



## 4.2 USING THE BASIC INTERFACE LAYOUT

အချက်ရှိ သင်ကြားပုံချမော်နေတဲ့ Chapter (4)ရဲ့ ဆောင်မှုမှာကိုက Arranging Components on a User Interface လို့ အမည်ပေးထားတယ်နော်။ အချက်ရှိ Layout Manager ကို အသုံးပြုပြီးတော့ User Interface အပိုင်းကို စိတ်ကြိုက်ဖန်တီးရယူနိုင်ပါတယ်။ ကဲ ဒီတော့ Basic Layout Manager နဲ့ ပတ်သက်ပြီးတော့ သေသေချာချာ လေ့လာကြည့်ကြရအောင်။

### ◆ Laying Out an interface

ကျွန်ုတ်တို့ Swing components ရဲ့ အစိတ်အပိုင်း တစ်ခုရှုကို အသုံးပြုမယ်ဆိုရင် တစ်နည်းပြောရှုရင် container classes ရဲ့ အစိတ်အပိုင်းတစ်ခုရှုကို ခေါ်ယူ အသုံးချမယ်ဆိုရင် အဲဒီ components တွေကို arrangedလုပ်နိုင်ဖို့အတွက် layout manager ကနေ determine လုပ်ပါတယ်။ ဥပမာ- ပြီးခဲ့တဲ့ chapter index 4.1 က ToolBar.java program ကို လေ့လာကြည့်ပါ။ ဟုတ်ပြောနော်။

Layout manager class ကို အသုံးချမယ်ဆိုရင်တော့ ကျွန်ုတ်တို့ဟာ Flow Layout Class ကို default အားဖြင့် အသုံးချုပ်မှာပါပဲ။ ဒါဟာ Flowlayout တစ်ခုတည်းနဲ့ပဲ ပတ်သက်တာကို ပြောတာနော်။ ကျွန်ုတ်တဲ့ Layout manager တွေနဲ့ မသက်ဆိုင်သေးပါဘူး။ ဘာကြောင့်လဲဆိုတော့ Java Language မှာ ပါဝင်တဲ့ container class ထဲမှာ-

FlowLayout, GridLayout, BorderLayout,

CardLayout and GridBagLayout

ဆိုပြီးမှ ဂွဲပြားနေတာတွေကို index သင်ခန်းစာအလိုက် တွေ့ဖြင့် ရုပ်ပိုင်မယ်။ တကယ်လို့ method () တစ်ခုကို ခေါ်ယူ အသုံးချမယ်ဆိုရင် သက်ဆိုင်ရာ class ရဲ့ constructor method ကို ခေါ်ယူလိုက်ရှုပါပဲ။ ဥပမာ-

FlowLayout Flo = new flowlayout();  
လို့ ရေးသားပြီးတော့ ခေါ်ယူအသုံးချုပ်မှာပါပဲ။ အကယ်၍ layout manager ကို create လုပ်ပြီးတော့ အဲဒီ container class ထဲမှာ ပါဝင်တဲ့ -

Setlayout () method ကိုပါ ထည့်သွင်း အသုံးချုပ်တော့မှာပါပဲ။ ဒါပေမယ့် စာဖတ်သူဟာ layout manager တွေကို program ရေးစဉ်မှာ မထည့်သွင်းခဲ့ရင်လည်း default အားဖြင့် အောက်ပါအတိုင်း သတ်မှတ်ပေးလို့ ရပါတယ်။

- Flowlayout for panels
- BorderLayout for frames and windows

ကဲ ဒီတော့ layout manager နဲ့ setlayout () method ကို အသုံးချမယ်ဆိုရင် အရင်ဆုံး အောက်ပါ code\_line လေးကို

လေ့လာလိုက်ပါပါး။

Sample - public class starter extends javax. swing. JFrame{

```
    Public starter () {  
        Container pane = get content pane ();  
        Flowlayout Lm= new flowlayout ();  
        pane.setLayout (Lm);  
        // add component here  
        set contentPane (pane);  
    }  
}
```

အပေါ်က ရေးပြုတဲ့ sample code\_line မှာ // add component here နေရာမှာ components တွေကို ထပ်မံပြည့်သွင်းလို့ ရှိနိုင်ပါတယ်။ ဘာကြောင့်လဲဆိုတော့ layout manager ကို ဖြည့်သွင်း အသုံးချုပ်းသွားပြီးလေ။ ဒီတော့ components တွေကို add လုပ်တာကို container class ကနေ အလွယ်တကူ လက်ခံ ရှိနိုင်ပါလို့မယ်။ ဒါကို သဘောပေါ်အောင် အကြမ်းဖျင်းအားဖြင့် ရှင်းပြတာပါပဲ။

### ◆ Flow Layout

Flow layout calss ဟာ layout manager ထဲမှာ အရိုးရှင်းဆုံး class တစ်ခုပါပဲ။ ဒီ flow layout manager နဲ့ ပတ်သက်ပြီးတော့ ရှင်းပြရမယ်ဆိုရင် components တွေကို left to right အလိုက် စီစဉ်ဖန်တီးပေးပါတယ်။ ဒီနေရာမှာ JFrame Area ရှိတာကတော့ အကယ်၍ သတ်မှတ်ထားသော JFame Area က ဆုံးသွားပြီးဆိုရင်တော့ နောက် Row တစ်ခုကို စီစဉ်ပြီးတော့ ညိုယူပေးထားမှာပါ။

Flow Layout ကို သုံးမယ်ဆိုရင် flowlayout class ကို ခေါ်သုံးရမှာပါ။ အဲဒီအခါမှာ flowlayout () constructor method ကို ခေါ်သုံးရမှာပါ။ တကယ်လို့ flowlayout () constructor မှာ arguments မထည့်ဘူးဆိုရင်တော့ ထည့်သွင်းထားတဲ့ components တွေဟာ JFrame ရဲ့ အလယ်မှာ လာပေါ်ပေးနေမှာပါ။ တကယ်လို့ စာဖတ်သူဟာ ထည့်သွင်းထားတဲ့ components တွေကို aligned along the left or right edge အနေနဲ့ စီစဉ်မယ်ဆိုရင်

FlowLayout.LEFT or FlowLayout.RIGHT

ဆိုပြီး FlowLayout () constructor method ရဲ့ argument နေရာမှာ ထည့်သွင်းရေးသားရမှာပါ။ ဒါပေမယ့် FlowLayout () constructor မှာ arguments အနေနဲ့ တစ်ခုပဲ လက်ခံတာ ဖော်။ ကဲ ဥပမာအနေနဲ့ အောက်ပါအတိုင်း ရေးပြည့်ကြရအောင်။



```

FlowLayout righty = new Flowlayout
    (FlowLayout.Right);
ဒါဆိုရင်တော့ ထည့်သွင်းထားတဲ့ component ကို flow-
layout ဟာ ဉာဏ်အခြေးမှာ သွားပြီးတော့ စီစဉ်ပေးမှာပါပဲ။
တကယ်လို့
FlowLayout right = new FlowLayout
    (FlowLayout.Center);
လို့ ရေးလိုက်ရင် components တွေကို JFrames ရဲ့ အလယ်မှာ
လာပြီးတော့ စီစဉ်ပေးမှာပါပဲ။

```

ကဲ ဒီတော့ FlowLayout နဲ့ ပတ်သက်တဲ့ program တစ်ခု ကို Listing 4.2 အနေနဲ့ ရေးသားပိုချေပေးထားပါတယ်။ JButton (၆) ခုကို ဖန်တီးပြီးတော့ Flow Layout manager အနေနဲ့ ထိန်းချုပ်မှာပါပဲ။ Flowlayout () constructor method ထဲမှာ FlowLayout.LEFT ဆိုတဲ့ arguments ကို ထည့်သွင်းရေးသား ထားတဲ့ အတွက် ထည့်သွင်းထားတဲ့ JButton components တွေ ကို lined up အတိုင်း left side of the JFrame windows က နေ စီစဉ်ပြီးတော့ ဖန်တီးပေးပါလိမ့်မယ်။ သတ်မှတ်ထားတဲ့ JFrame ရဲ့ အချက်အစား width အတိုင်း ပြည့်သွားရင် နောက် lined up အတိုင်း စီစဉ်ပေးပါလိမ့်မယ်။ သေသေချာချာ လေ့လာ ကြည့်ပါ။

Listing 4.2 Demonstrate the flow layout method

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. class Alphabet extends JFrame {
6. JButton a = new JButton ("Alibi");
7. JButton b = new JButton ("Burglar");
8. JButton c = new JButton ("Corpse");
9. JButton d = new JButton ("Deadbeat");
10. JButton e = new JButton ("Evidence");
11. JButton f = new JButton ("Fugitive")
12.
13. Alphabet () { // constructor function
14. Super ("Alphabet");
15. Setsize (360, 120);
16. SetDefault Close Operation (JFrame. Exit . on .
Close);
17.
18. JPanel. pane = new JPanel ();
19. Flowlayout Lm = new Flowlayout (FlowLayout.
LEFT);
20. pane. setlayout (Lm);
21. pane. add (a);
22. pane . add (b);
23. pane. add (c);

```

```

24. pane . add (d);
25. pane. add (e);
26. pane. add (f);
27. setcontent pane (pane);
28. setVisible (true);
29. }
30. public static void main (string [ ] arguments){
31. JFrame. Frame = new Alphabet ();
32. Frame. show ();
33. }
34. }

```

### Analysis

၁။ အရင် ဆုံး : java.awt.\*; java.awt.event.\*; java.
swing.\*; ဆိုတဲ့ package file (၃) ခုကို program ရဲ့ အပေါ်ဘက်မှာ import & declare လုပ်ထားပါတယ်။ ပြီးတော့ JFrame Class ကို extends လုပ်ပြီးတော့ Alphabet ဆိုတဲ့ primary class ကို ဖန်တီး ရေးသားပါ တယ်။ အဲဒီ primary class ထဲမှာ JButton () constructor method ကို အဖို့ပြုပြီးတော့ Button (6) ခုကို ဖန်တီးပါတယ်။ JButton class ရဲ့ Object\_Name တွေ ကို a to f လို့ အသီးသီးပေးထားပါတယ်။ JButton () constructor ရဲ့ parameter arguments မှာ ရေးသား ထားတဲ့ စာသားတွေကတော့ Button အသီးသီးမှာ ပါဝင် ရမယ့် စာသားတွေပါပဲ။ ဒါတွေဟာ primary\_class Alphabet ထဲမှာ ရေးသားထားတာနော်။ အဲဒီကို သတိထား လိုက်ပါတီး။ လိုင်းနံပါတ် (၁၃) မှာတော့ Alphabet class ရဲ့ constructor function ကို ရေးသားထားပါတယ်။ အဲဒီ constructor function ထဲမှာ super () method ကို အသုံးပြုပြီး ရေးသားထားတာကတော့ Frame Windows ရဲ့ Title Bar မှာ ပါဝင်ရမယ့် စာသားကို ရေးသားထားပါ တယ်။ Frame Windows ရဲ့ အချက်အစား Width& Height ကိုတော့ SetSize (360, 120); // width and Height of Frame ဆိုပြီးမှ ရေးသားထားပါတယ်။ Frame windows ကို Close & Exit လုပ်နိုင်အောင်အတွက်ကတော့ – SetDefault Close Operation (JFrame. EXIT\_ON\_CLOSE); လို့ ရေးသားထားပါတယ်။ ပြီးတော့ လိုင်းနံပါတ် (၁၈) မှာ JPanel class ကို Object\_Name pane လို့ နာမည်ပေး ထားပြီးတော့ ခေါ်ယူထားလိုက်ပါပြီ။ ပြီးတော့ FlowLayout Class ကိုလည်း Object\_Name Lm လို့ အမည် ပေးပြီး ခေါ်ယူလိုက်ပါတယ်။ FlowLayout() ရဲ့ parameter arguments နေရာမှာတော့ အပေါ်မှာ ရှင်းပြုခဲ့တဲ့

အတိုင်း Frame Windows ရဲ့ ဘယ်ဘက်ကနေ စတင်ပေါ်လာအောင် FlowLayout.LEFT လို့ ထည့်သွင်းရေးသေးသားပါတယ်။ အဲဒီအချက်ကို သော့သေချာချာ လေ့လာကြည့်ပါ။ ပြန်ရေးပြုပေးသားပါတယ်။

FlowLayout Lm = new FlowLayout  
(FlowLayout.LEFT);

- JPanel class ထဲက setLayout () method ကို ခေါ်ယူ အသုံးချဖိုင်စိုးအတွက် object name နဲ့ ယုံ့တွေပြီးတော့ ခေါ်ယူ အသုံးချပါတယ်။ ဒါကြောင့် Pane.setLayout (Lm);  
လို့ တွေ့မြင်ရမှာပါ။ အဲဒီနောက် အပေါ်မှာ ဖန်တီးခဲ့တဲ့ JButton တွေကို JPanel Class ထဲက add () method ကို အသုံးပြုပြီးတော့ ခေါ်ယူပါတယ်။

Note : SetLayout () method နဲ့ add () method တွေဟာ JPanel class ထဲမှာ ပါဝင်တာတွေပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၂၇) (၂၈)မှာ ရေးသားတဲ့ SetLayout Pane () method ရယ်၊ SetVisible () method ရယ်ကို တော့ ရှင်းပြစ်ရာ မလိုတော့ပါဘူး။

- ၃။ လိုင်းနံပါတ် (၂၀)မှာ main () function ကို ရေးသားပေးထားပါတယ်။ အဲဒီ main () function ထဲမှာမှ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ Alphabet class ကို ခေါ်ယူလိုက်တာပါပဲ။ အဲဒီနောက် JFrame windows ကို show () လုပ်ခိုင်းတာပါပဲ။ ဒီအခါမှ Button (6) ရဲ့ ပါဝင်တဲ့ Alphabet ဆိုတဲ့ [FlowLayout method ကို အသုံးချထားတဲ့] program ငယ်လေးဟာ လာပြီးတော့ run နေမှာပါပဲ။ FlowLayout ကို အသုံးချတဲ့အတွက် swing components တွေကို အသုံးချရမှာ အရင်လို x, y coordinate မျဉ်းစိုင်းနေရာတွေကို ထည့်သွင်းပေးစရာမလိုတော့ပါဘူး။ ဒီအချက်ဟာ အချက်အလက်ပါပဲ။

ကဲ listing 4.2 က program ကို သဘောပေါက်သွားရှင်တော့ Notepad ထဲမှာ သွားပြီးတော့ code\_line တွေကို ရေးသားကြရအောင်လား။ File\_Name ကိုတော့ Alphabet.java လို့ အမည်ပေးပါ။ အဲဒီလို code\_line တွေကို ရေးသားပြီးတော့ save လုပ်ထားပုဂ္ဂိုလ် Figure 4.4 အနေနဲ့ ရေးသား ဖော်ပြထားပါတယ်။ လေ့လာကြည့်ပါ။

```

Alphabet.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class GridLayout extends JFrame{
    JButton a = new JButton("Kyaw Zayar Lay.");
    JButton b = new JButton("Phyu Phyu Kyaw.");
    JButton c = new JButton("Kyaw Myo Htet");
    JButton d = new JButton("Ma Moe Thida");
    JButton e = new JButton("Zar Zar oo.");
    JButton f = new JButton("Ko Bo Min");
    JButton g = new JButton("Ma Wit Yi");
    JButton h = new JButton("Ma Zarchi");
    JButton i = new JButton("Ko ThiHa");

    GridLayout(){
        super("GridLayout");
        setSize(360, 360);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

        JPanel pane = new JPanel();
        GridLayout obj = new GridLayout(3,3,10,10);
        pane.setLayout(obj);
        pane.add(a);
        pane.add(b);
        pane.add(c);
        pane.add(d);
        pane.add(e);
        pane.add(f);
        pane.add(g);
        pane.add(h);
        pane.add(i);
        setContentPane(pane);
        setVisible(true);
    }

    public static void main(String[] arguments){
        JFrame frame = new GridLayout();
        frame.show();
    }
}

```

Figure 4.4 Write the Program code line in the Notepad, the Alphabet. java program

ကဲ Notepad ထဲမှာ အထက်ဖော်ပြထားတဲ့အတိုင်း သွားရောက်ပြီးတော့ code\_line တွေ ရေးသားပြီးတော့ save လုပ်ပါ။ File name ကတော့ Alphabet.java နောက်။ ပြီးရင် compile & run လုပ်စိုးအတွက် command line windows (DOS) ထဲကို သွားပါ။ ပြီးရင်တော့ အောက်ပါ command line နဲ့ directory path ကို ပြောင်းလိုက်ပါ။

cd c:\j2sdk 1.4.1\_01\bin

ဒါဆိုရင်တော့ DOS windows ထဲက prompt သက်တာဟာ c:\j2sdk 1.4.1\_01\bin2 > လို့ ပေါ်နေမှာပါပဲ။ အဲဒီအခါမှာတော့ အောက်ပါ commd line နဲ့ Alphabet.java program ကို compile လုပ်လိုက်ပါ။

javac Alphabet ↵

အကယ်၍ error မရှိခဲ့ဘူးဆိုရင်တော့ အောက်ပါ command နဲ့ run လုပ်ပါ။

java Alphabet ↵

ဒါဆိုရင်တော့ Alphabet . java program ဟာ run နေမှာပါ။ Compile & Run လုပ်ပုဂ္ဂိုလ် Figure 4.5 အနေနဲ့ ရေးသားပြီးတော့ ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 4.5 Complie & Run of the Alphabet.java in the command prompt

Figure 4.5 ကတေသာ DOS Windows/Command line Interface ထဲမှ Compile & Run လုပ်နေတဲ့ပါဝဲ။ Program Output ထွက်လာပုံကို တော့ Figure 4.6 အနေနဲ့ ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 4.6 . Run the Alphabet.java Program

ဒါဆိုရင်တော့ စာဖတ်သူဟာ flowlayout method ကို ထည့်သွင်း အသုံးချင့် gap of pixels အဖြစ် ပါဝင်လာပုံတွေကို စာဖတ်သူအနေနဲ့ အလွယ်တကူ နားလည် သဘောပေါက်သွားမှာ ပါ။

Note : အခါ Alphabet application ထမား flow layout manager ကို အသုံးချတဲ့အခါမှာ default အနေနဲ့ gap ကို five pixels ဆိုပြီး ထည့်ပေးထားပါတယ်။ Gap of pixels ဆိုတာကတော့ components တစ်ခုနဲ့ တစ်ခု ကြားမှာ (တစ်နည်းပြောရင် row တွေကြားမှာပေါ့) အနည်းငယ် မြားပေးဖို့အတွက် ဖန်တီးရတာပဲ ဖြစ်ပါ

တယ်။ ဒါပေမယ့် gap of pixels ကို ထည့်သွင်းတဲ့ အခါမှာ horizontal and vertical gap တွေကို component အသီးသီးကြားမှာ စိတ်ကြော် ထည့်သွင်းလို့ လည်းရပါတယ်။ အဲဒီအခါမှာ flowlayout( ) constructor method ကို အသုံးပြုရမှာပါ။ ကဲ အဲဒီ method ကို အောက်မှာရှေ့ပြပေးထားပါတယ်။

FlowLayout (int, int, int)

ဆိုပြီး constructor method ထဲက arguments (3) ခုတို့ တွေ ဖြစ်ရမှာပါ။ arguments တစ်ခုချင်းစိတ် ရှင်းပြပါမယ်။

- The alignment which must be FlowLayout.CENTER, FlowLayout.LEFT, or FlowLayout.RIGHT
  - The horizontal gap between components, in pixels
  - The vertical gap, in pixels
- ကဲ အဲဒီ constructor method ကို အသုံးပြုပြီးတော့ code\_line တစ်ခြောင်း ရေးပြပါမယ်။

FlowLayout flo = new FlowLayout  
(FlowLayout.CENTER, 30,10)

ကဲ အဲဒီကို လေ့လာကြည့်မယ်ဆိုရင်

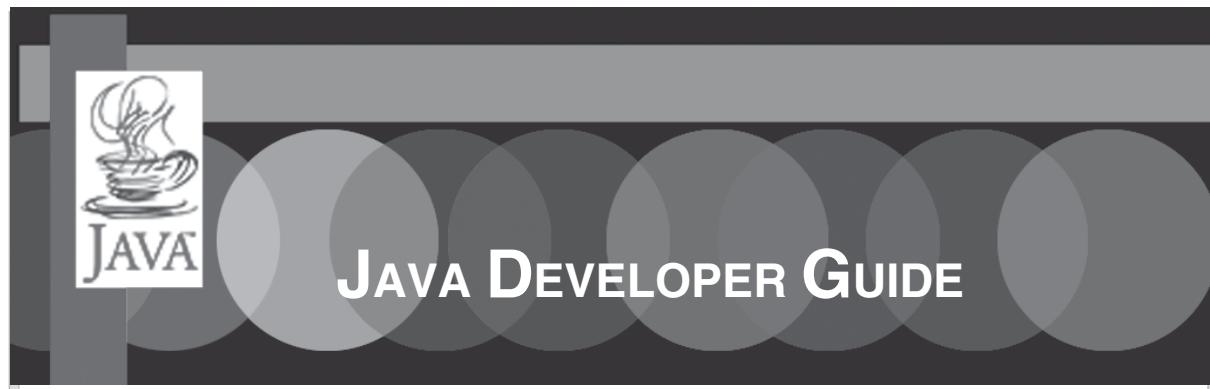
1<sup>st</sup> arguments → flow layout manager ကို JFrame ရဲ့ အလယ်မှာ စိစဉ်ပေးပါလိမ့်မယ်။

2<sup>nd</sup> arguments → Horizontal gap of 30 pixels အနေနဲ့ စိစဉ်ပေးဖို့ပါ။

3<sup>rd</sup> arguments → Vertical gap of 10 pixels အနေနဲ့ စိစဉ်ပေးပါလိမ့်မယ်။

ကဲ ဒီ gap of pixels အနေနဲ့ စိစဉ်တာတွေကို လာမယ့် သင်ခန်းစာဖြစ်တဲ့ GridLayout, BorderLayout အစရိတ် သင်ခန်းစာတွေမှာ ဆက်လက်ပို့ချေပေးပါလိမ့်မယ်။

ကဲ ဒီတော့ index 4.3 Using the GridLayout of Java ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက်ပြီးတော့ လေ့လာကြရအောင်လား။ ဒါမှာသာ Java Language နဲ့ ပတ်သက်တဲ့ Basic interface layout တွေအကြောင်းကို အခြေခံကျကျ နားလည် သဘောပေါက်လာမှာပေါ့။ ကဲ လေ့လာလိုက်ပါပြီး။



Chapter (4)

## 4.3 USING THE GRID LAYOUT OF JAVA

The grid layout manager arranges components into a grid of rows and columns. Components are added first to the top row of the grid, beginning with the leftmost grid cell and continuing to the right. When all the cells in the top row are full, the next components are added to the leftmost cell in the second row of the grid- if there is a second row and so on.

ဒါကတော့ စာဖတ်သူအနေနဲ့ အလွယ်တကူ လေ့လာလို့ရ အောင် Guide Book ထဲကအတိုင်း ရေးသားပေးထားတာပါ။ Simple English ဖြစ်တာကြောင့် Grid Layout အကြောင်းကို စာဖတ်သူ အလွယ်တကူ နားလည်နိုင်မှာပါ။ Gridlayout manager ဟာ components တွေကို arranges လုပ်ပေးပါတယ်။ Components တွေကို added လုပ်တဲ့အခါမှာ first to the top row of the grid ကနေ ထည့်သွင်းမှာပါ။ ထည့်သွင်းတဲ့အခါ မှာ leftmost grid cell ကနေ right ကို ထည့်သွင်းမှာကိုလို့။ တကယ်လို့ top row cell ဟာ fulls ဖြစ်သွားပြီနိုင်ရင် the next component ဟာ ဒုတိယrow ရဲ့ ဘယ်ဘက်အစွန်ဆုံး cell ကနေ ထည့်သွင်းပြီးတော့ ချိန်နှုံးပေးမှာပါပဲ။ တကယ်လို့ ဒုတိယ row ဟာ ပြည့်သွားပြန်ပြန်ရင်လည်း အထက်ပါ နည်းလမ်းအတိုင်းပဲ ဆက်လက်ပြီးတော့ စီစဉ်ပေးမှာပါပဲ။

Grid Layouts නි අන්ති:අලුත්තයේදී GridLayout  
class නි වොට්රියා තෙවූ ඇත්තු:වුගාඩි|| GridLayout ()  
constructor method මා තොග්‍රයිපාටියිං: රෝඳී:රුහිතයි||

1<sup>st</sup> arguments =Rows in the grid

2<sup>nd</sup> arguments = number of columns grid

က ဒီတော့ အောက်မှာရေးပြထားတဲ့ statement ကို လူလာ

ကြည့်မယ်ဆိုရင် GridLayout manager ကို အသုံးပြုစီးတော့ 10 rows and 3 columns အနေနဲ့ ဖန်တီးထားတာပါပဲ။

```
GridLayout gr = new GridLayout(10,3);
```

ကျွန်ုပ်တော်တိ အရှေ့က FlowLayout class ကို ခေါ်ပြီး  
တော့ အသုံးပြုစဉ်တုန်းက arguments မှာ horizontal gap နဲ့  
vertical gap တွေကို components တွေကြားမှာ ထည့်သွင်းဖို့  
အတွက် gap pixels တွေ ထည့်သွင်းဖူးတာကို မှတ်စိပါလိမယ်။  
မူလပါဝင်ပြီးသား Grid Layout ရဲ့ constructor method  
နေရာမှာ နောက်ထပ် 3<sup>rd</sup> & 4<sup>th</sup> arguments တွေ ထပ်ဖြည့်လိုက်ရှု  
ပါပဲ။

3<sup>rd</sup> arguments = horizontal gap of pixels

4<sup>th</sup> arguments = vertical gap of pixels

က အောက်မှာ ရေးပြထားတဲ့ statements ကတော့ အပေါ်  
က code line ကို ပြင်ရေးထားတာပါပဲ။ 10 row and 3 column  
ပါဝင်ဖြိုးတော့ horizontal gap 5 pixels နဲ့ vertical gap က  
8 pixels ခါးထားတဲ့ statements ပါပဲ။ လွှေလာကြည့်လိုက်ပါ။

```

Grid      = new      (10,   3,   5,   8);
Layout grz  GridLayout
             ↑          ↑          ↑          ↑
             Rows in the grid
             number of
             columns grid
             Horizontal gap
             of pixels
             vertical gap
             of pixels

```



ဒါဟာ gap of pixels တွေကို ထည့်သွင်းပြီးတော့ ရေးသားထားတာပါပဲ။ စာဖတ်သူအနေနဲ့ ဘာမျှ ထည့်သွင်းပြီးတော့ မရေးသားဘူးဆိုရင်တော့ default အားဖြင့် grid layout ဟာ vertical နဲ့ horizontal directions ကြားမှာ 0 pixels အနေနဲ့ပဲ စိစဉ်ပေးမှာပါ။ ကဲ ဒီတော့ လက်တွေ့ အနေနဲ့ Listing 4.3 အနေနဲ့ ရေးပြထားတဲ့ Demonstrate the Grid Layout ဆိုတဲ့ သင်ခန်းစာကို လက်တွေ့ လေ့လာကြည့်ကြရအောင်လား။ အလွယ်ကူဆုံး ဖြစ်အောင် အတတ်နိုင်ဆုံး ရှင်းလင်း ပို့ချပေးထားပါတယ်။

#### Listing 4.3 Demonstrate the Grid Layout

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. class GridLayout extends JFrame {
6. JButton a = new JButton ("Kyaw Zayar Lay.");
7. JButton b = new JButton ("Phyu Phyu Kyaw.");
8. JButton c = new JButton ("Kyaw Myo Htet.");
9. JButton d = new JButton ("Zar Zar Oo.");
10. JButton e = new JButton ("Ko Bo Min.");
11. JButton f = new JButton ("Ma Wit Yi.");
12. JButton g = new JButton ("Ko ThiHa");
13. JButton h = new JButton ("Ma Zar Chi");
14. JButton i = new JButton ("Ma Moe ThiDa");
15. GridLayout () {
16. Super ("Demonstrate the GridLayout");
17. SetSize (420,400);
18. SetDefault Close Operation (JFrame.EXIT-ON-
 - CLOSE);
19. JPanel Pane = new JPanel ();
20. GridLayout obj = new GridLayout (3,3,10,10);
21. Pane.setLayout (obj);
22. Pane.add (a);
23. Pane.add (b);
24. Pane.add (c);
25. Pane.add (d);
26. Pane.add (e);
27. Pane.add (f);
28. Pane.add (g);
29. Pane.add (h);
30. Pane.add (i);
31. SetContentPane (pane);
32. SetVisible (true);
33. }
34. Public static void main (string [ ] arguments){
35. JFrame frame = new GridLayout () ;
36. frame. show ();
37. }
```

#### 38. }

#### Analysis

၁။ Grid Layout နဲ့ ပတ်သက်တဲ့ demonstrate program ကို listing 4.3 မှာ ရေးသား ဖော်ပြပေးထားပါတယ်။ ဒီ program မှာလည်း JButton တွေကို အပ်စွဲ။ ခေါ်ယူ အသုံးချထားပါတယ်။ ပြီးတော့မှာ GridLayout class ကို ခေါ်ယူ အသုံးချထားတာပါပဲ။ ဒီတော့ လေ့လာကြည့်ကြရအောင်လား။ Package file (3) ခုကို import & declare လုပ်ထားပါတယ်။ အဲဒီ package file (3) ခုကတော့-import java.awt.\*;
import java.awt.event.\*;
import javax.swing.\*;

စုံပဲဖြစ်ပါတယ်။ ဒီ package file တွေကတော့ အခုလက်ရှိ သင်ခန်းစာတော်တော်များများ ထည့်သွင်း အသုံးချနေတဲ့ package file တွေပါပဲ။ ပြီးတော့ လိုင်းနံပါတ် (၅) မှာ JFrame class ကို extends လုပ်ပြီး ဖန်တီး ရေးသားထားတဲ့ GridLayout ဆိုတဲ့ primary class ကို ဖန်တီး ရေးသားပါတယ်။ အဲဒီနံပါတ် (၅) ခုကနေ လိုင်းနံပါတ် (၁၄) အထိပါပဲ။ အဲဒီတွေကို မြင်အောင် အရင် လေ့လာကြည့်လိုက်ပါဉိုး။

၂။ ကဲ လိုင်းနံပါတ် (၁၅) မှာတော့ GridLayout class ရဲ့ constructor () method ကို ရေးသားထားပါတယ်။ အဲဒီ constructor function ထဲမှာ super () method ကို အသုံးပြုပြီးတော့ JFrame Windows ရဲ့ title text ကို ဖန်တီးရေးသားပါတယ်။ ပြီးတော့ JFrame Class ရဲ့ SetSize () method ကို အသုံးချပြုပြီးတော့ Output Windows ရဲ့ အချက်အအစား width & height ကို ဖန်တီးရေးသားပါတယ်။ Width = 420 နဲ့ height = 400 ပါပဲ။ အဲဒီနံပါတ် SetDefault Close Operation () method ကို အသုံးချထားပါတယ်။ လိုင်းနံပါတ် (၁၈) မှာပါ။ အဲဒီ နံပါတ် လိုင်းနံပါတ် (၁၉) မှာ JPanel Class ကို Object\_Name Pane လို့ ပေးပြီးတော့ ခေါ်ယူပါတယ်။ GridLayout class ကိုတော့ Object\_Name obj လို့ပဲ ပေးပြီးတော့ ခေါ်ယူလိုက်ပါတယ်။ ဒီနေရာမှာတော့ Grid Layout class ရဲ့ constructor method မှာ Row & Column သာမကပါဘူး။ Gap of pixels တွေကို ထည့်သွင်း အသုံးပြုထားပါတယ်။ ဒါကြောင့်



```
GridLayout obj = new GridLayout(3, 3, 10, 10);
    ↑   ↑
    Row &Columns      gap pixels of
                      vertical and horizontal
```

လို့ ရေးသားထားတာပါပဲ။ အဲဒီနောက် JPanel Class ထဲက setlayout ( ) method ထဲကို gridlayout class ကို arguments အနေနဲ့ ထည့်သွင်း အသုံးချဖိတယ်။ Parameter arguments အနေနဲ့ပါပဲ။ ပြီးတော့ လိုင်းနံပါတ် (jj) ကနေ လိုင်းနံပါတ် (20) အထိမှာတော့ အပေါ်က ဖန်တီးခဲ့တဲ့ [global variable အနေနဲ့ declare လုပ်ခဲ့တဲ့ ] JButton တွေကို JPanel Class ထဲမှာ setup လုပ်ပါတယ်။ ဒီတော့ အနုလက်ရှိ constructor ( ) function ထဲမှာ ရေးသားထားတဲ့ code line တွေကတော့ အစွယ်တကူပဲ နားလည် သဘောပေါက်နိုင်ကြမှာပါ။ ဒီတော့ လိုင်းနံပါတ် (24) မှာ ရေးသားထားတဲ့ main () function ကို လေ့လာ ကြည့်ကြရအောင်လား။ ဘာမျှတော့ အထွေအထူး မရှိပါဘူး။ အပေါ်ကဖန်တီးခဲ့တဲ့ gridlayout class ကို object\_Name ပေးပြီး ၏ဦးလိုက်တာပါပဲ။ ပြီးတော့ JFrame Class ကို အမှိုပြုပြီးတော့ gridlayout.java ဆိုတဲ့ program ဟာ အလုပ် လုပ်သွားတာပါပဲ။ လိုင်းနံပါတ် (27) က တော့ main () function ရဲ့ close ရဲ့ close brace } ပါပဲ။ Primary class ရဲ့ close } ကတော့ လိုင်းနံပါတ် (28) မှာ လာဝိတိထားတာပါပဲ။ ဒါဆိုရင်တော့ အခု program ကို နားလည် သဘောပေါက်သွားမှာပါ။

ဒီ gridlayout.java ဆိုတဲ့ program မှာ အရှေ့သင်ခန်းစာ မှာ သင်ကြားပို့ချပေးခဲ့တဲ့ flow layout class ကို အသုံးချထားပဲနဲ့ ပုံစုံ မဖြစ်ဘူးလား။ ပြီးခဲ့တဲ့ chapter index 4.2 ကို ပြန်လုပ် လေ့လာကြည့်စစ်ပါ။ ဒါဆိုရင် flow layout နဲ့ ဘာကွာခြားတာလဲဆိုတာကို စာဖတ်သူအနေနဲ့ အလွယ်တကူ နှိုင်းယူဉ်ပြီးတော့ သဘောပေါက်လာပါလိမ့်မယ်။ ဒဲ ဒီတော့ code\_line တွေကို ရေးသားပြီးတော့ Notepad ထဲမှာ save လုပ်ထားပဲရို့ figure 4.7 အနေနဲ့ ရေးသားပို့ချပေးထားပါတယ်။ File name ကတော့ gridlayout.java ဖြစ်ပြီးတော့ save လုပ်ရမယ့် နေရာကတော့ ထုံးစံအတိုင်းပါပဲ။

C:\j2sdk 1.4.1\_01\bin ရဲ့ အောက်မှာပါပဲ။ ဒီတော့ notepad ထဲကို code\_line တွေ ရှိက်သွင်းပြီးတော့ gridlayout.java လို့ အမည်ပေး တည်ဆောက်ထားတာကို figure 4.7 အနေနဲ့ လေ့လာကြည့်ပါ။

```
gridLayout.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class GridLayout extends JFrame {
    JButton a = new JButton("Kyaw Zayar Lay.");
    JButton b = new JButton("Phyu Phyu Kyaw.");
    JButton c = new JButton("Kyaw Mya Htet");
    JButton d = new JButton("Ma Mo Thiba");
    JButton e = new JButton("Zar Zar Oo");
    JButton f = new JButton("Ko Bo Min");
    JButton g = new JButton("Ma Wit Yi");
    JButton h = new JButton("Ko Thihha");
    JButton i = new JButton("Ma Zarchi");

    GridLayout() {
        super("Demonstrate the GridLayout");
        setSize(420, 400);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        GridLayout obj = new GridLayout(3, 3, 10, 10);
        pane.setLayout(obj);
        pane.add(a);
        pane.add(b);
        pane.add(c);
        pane.add(d);
        pane.add(e);
        pane.add(f);
        pane.add(g);
        pane.add(h);
        pane.add(i);
        setContentPane(pane);
        setVisible(true);
    }

    public static void main(String[] arguments) {
        JFrame frame = new GridLayout();
        frame.show();
    }
}
```

Figure 4.7 Write the program code\_Line in the Notepad, gridlayout.java

က figure 4.7 ကအတိုင်း file ကို ရေးသားပြီးတော့ save လုပ်ပြီးပို့ရင်တော့ Start → Program → Accessories → Command prompt ကို ဖြစ်စေ၊ Start → All programs → Accessories → Command prompt ကို ဖြစ်စေ၊ ဒါမှမဟုတ် Start → Run ကို သွားပြီး Run Dialog Box ကျလာရင် command.com လို့ ရှိက်သွင်းပြီးတော့ command-line interface ကို ဖွင့်ပါ။ တစ်နည်းပြောရှင်တော့ DOS ပဲပေါ်နော်။ အဲဒီ command prompt ကို ဖွင့်ပြီးတော့ gridlayout.java program ကို Compile & Run ထားပဲရို့ figure 4.8 အနေနဲ့ ရေးသားဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 4.8 Compile & Run of the gridlayout.java

က အခုဖော်ပြထားတဲ့ figure 4.8 ကအတိုင်း Compile & Run လုပ်လိုက်ပြီး ဆိုပါစို့။ GridLayout .java ဆိုတဲ့ program



ဟာ စာဖတ်သူရဲ့ computer ရဲ့ Desktop ပေါ်မှာ လာပြီးတော့ အောက်ပါအတိုင်း ဖော်ပြပေးနေမှာပါပဲ။



Figure 4.9 Run the program, gridlayout.java

က ဒါဆိုရင်တော့ GridLayout.java ဆိုတဲ့ Demonstrate program ယောက်လုပ်လုပ်သူးပုဂ္ဂိုရော Compile & Run လုပ်ထားပုဂ္ဂိုရော စာဖတ်သူအနေနဲ့ အလွယ်တကူ နားလည် သဘောပေါက်သွားမှာပါပဲ။

ဆက်လက်ပြီးတော့ Chapter (4) ရဲ့ sub-index 4.9 အနေနဲ့ ရေးသားထားတဲ့ သင်ခန်းစာဖြစ်တဲ့ using the borderlayout ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် လေ့လာကြည့်ကြရအောင်လား။ ဒါဆိုရင်တော့ Basic interface layout တွေနဲ့ ပတ်သက်ပြီးတော့ တော်တော်လေးကို လေ့လာမိလာမှာပါပဲ။ က လေ့လာလိုက်ကြရ အောင်။

#### 4.4 USING THE BORDER LAYOUT

Border Layout ကို ဖန်တီးရမှာလည်း အရှေ့က သင်ခန်း စာတွေလိုပါပဲ။ BorderLayout class ကို အသုံးပြုပြီးတော့ ဖန်တီးရမှာပါပဲ။ BorderLayout ကို ဖန်တီးတဲ့ container calss ထဲမှာ divided into five sections ဆိုပြီးမှ ဖန်တီးထားပါတယ်။ Divide section လုပ်တဲ့အခါမှာလည်း-

- North Section
- South Section
- East Section
- West Section
- Center Section ဆိုပြီးတော့မှ five areas ကို ချုပြီးထားပါတယ်။ ဒါတော့ အဲဒီ section အသီးသီးကို ဘယ်လို အrranged လုပ်ထားသလဲဆိုပြီးတော့ figure 4.10 မှာ ဖော်ပြပေးထားပါတယ်။ အရင်ဆုံး လေ့လာကြည့်ပါ၍။



Figure 4.10

BorderLayout class ကို လေ့လာပြီးတော့ အထက်ပါ အတိုင်း အသုံးချမယ်ဆိုရင် BorderLayout ရဲ့ Result & Arrangement type တွေကို အခြေခံကျကျ နားလည် သဘော ပေါက်ထားရမယ်နော်။ အခြေခံအားဖြင့်တော့ Result ထွက်လေတဲ့ အခါမှာ central components ဟာ ကျနိုင်တဲ့ပတ်ပတ်လည်မှာ ပါဝင်တဲ့ components လေးစုတက်ပိုပြီးတော့ ကြီးမားနေတာကို တွေ့ပြင်ရမယ်နော်။

က ဟုတ်ပြုလေ။ BorderLayout class ကို အသုံးပြုမယ် ဆိုရင် သူရဲ့ constructor function () ကို လေ့လာကြည့်ကြရ အောင်လား။ အခြေခံအားဖြင့်တော့

- BorderLayout()
- BorderLayout(int, int)

ဆိုတဲ့ constructors function () တွေကို တွေ့ပြင်ရမှာပါပဲ။ ဒါဟာ overloading method အရ ရေးသားထားတာပါပဲ။ လေ့လာကြည့်မယ်ဆိုရင် parameter argument မပါဝင်တဲ့ borderlayout () method ကို အသုံးချမယ်ဆိုရင် အရှေ့ သင်ခန်းစာ တွေမှာ ပါဝင်တဲ့ အခြား layout class တွေကအတိုင်း ဖန်တီး လိုက်တဲ့ components အသီးသီးကြားမှာ gap ဟာ မပါဝင်ပါဘူး။ ဒါဟာ အရှေ့က သင်ခန်းစာတွေက layout calss တွေ အတိုင်းပဲနော်။

တကယ်လို့ စာဖတ်သူဟာ second constructor method ကို အသုံးချမယ်ဆိုရင်တော့ စာဖတ်သူအနေနဲ့ horizontal gap နဲ့ vertical gap အသီးသီးကို ထည့်သွင်း ထွက်ချက်ပြီးတော့ အသုံးချလို့ ရပါတယ်။

စာဖတ်သူဟာ BorderLayout ကို အသုံးပြုပြီးတော့ pro-



gram ကို ဖန်တီးမယ်ဆိုရင် container class ထဲက layout manager ဟာ ပါဝင်လာဖြေနော်။ ဒီတော့ ပါဝင် အသုံးပြုထားတဲ့ components တွေကို add () method ကို အသုံးပြုပြီး ရေးသားရမှာပါ။ ဒါလေမယ့် add () method ကို အသုံးချတဲ့အခါမှာ အရှေ့က အသုံးပြုခဲ့တဲ့ add () method နဲ့ ကွာခြားမှု ရှုပါတယ်။ အဲဒါကိုတော့ အထူး သတိပြုပြီးတော့ လေ့လာပါ။ ဒီတော့ add () method ရဲ့ arguments type ကို လေ့လာကြည့်ရအောင်လား။

add (String, Component)

အခု အသုံးပြုထားတဲ့ add () method မှာ ပါဝင်ရမယ့် arguments (2) ရှိမှာ 1<sup>st</sup> arguments ကတော့ BorderLayout ထဲမှာ ပါဝင်ရမယ့် assign လုပ်ထားတဲ့ Swing\_Object\_component ပေါ်မှာ ပါဝင်ရမယ့် string စာသားပါပဲ။ ဥပမာအားဖြင့် အရှေ့က Figure 4.10 မှာ ပါဝင်တဲ့ "North", "South", "East", "West", or "Center" အစရိတာတွေပဲ ဖြစ်ပါတယ်။ မိမိနှစ်သက်တဲ့ (ဖြစ်စေချင်တဲ့၊ ရေးချင်တဲ့ string ကို double quote " " ကြားမှာ ရေးလိုက်ရှုပါပဲ။) 2<sup>nd</sup> argument ကတော့ ဒီ method ကို အသုံးပြုတဲ့အခါ container အတွက် ဘယ်နေရာ မှာ ထည့်သွင်းမယ်ဆိုတာကို ရေးသားတဲ့ အပိုင်းပါပဲ။ တစ်နည်းပြာရရှင်တော့ BorderLayout ရဲ့ portion ကို စိစဉ်ပေးတဲ့အပိုင်းပါပဲ။

ဥပမာအားဖြင့် add ("North", quit button); လို့ ရေးသားလိုက်ရင် quit button လို့၏၏တဲ့ JButton တစ်ခုတဲ့ borderlayout ရဲ့ north portion အပိုင်းမှာ လာရောက်ပြီး တော့ add လုပ်ပေးမှာပါပဲ။ က ဒီတော့ Listing 4.4 အနေနဲ့ BorderLayout ဖန်တီး ရေးသားထားပဲကို program နဲ့တော့ လေ့လာကြအောင်လား။

#### Listing 4.4 Demonstrate the Borderlayout

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. Class borderlayout extends JFrame {
6.     JButton a = new JButton ("North");
7.     JButton b = new JButton ("South");
8.     JButton c = new JButton ("East");
9.     JButton d = new JButton ("West");
10.    JButton e = new JButton ("Center");
11.
12.    borderLayout ( )

```

```

13.    Super ("Demonstrate the borderlayout:");
14.    setSize (240,280)
15.    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
16.    JPanel pane = new JPanel ();
17.    BorderLayout Obj = new BorderLayout ();
18.    pane.setLayout (obj);
19.    pane.add ("North",a);
20.    pane.add ("South", b);
21.    pane.add ("East", c);
22.    pane.add ("West", d);
23.    pane.add ("Center",e)
24.
25.    SetContentPane (pane);
26.
27  }
28.
29. public static void main (String [ ] arguments) {
30.     JFrame frame = new borderlayout ();
31.     frame.show ();
32. }
33. }

```

#### Analysis

၁။ BorderLayout ကို demonstrate လုပ်ပြုတဲ့ programme မှာ ထုံးအတိုင်း package file (3) ရှိကို declare လုပ်ပါတယ်။ ဤတော့ borderlayout ဆိုတဲ့ primary class ကို ဖန်တီး ရေးသားပါတယ်။ ထုံးစံအတိုင်း JFrame class ကို extends လုပ်ထားတယ်နော်။ ဤတော့ JButton (5) ခုတို့ variable name a, b, c, d, e လို့ ပေးပြီးတော့မှ ဖန်တီးထားပါတယ်။ ဤတော့ လိုင်းနဲ့ပါပဲ (၁၂)မှာတော့ borderlayout () ဆိုတဲ့ constructor function ကို ရေးသားထားပါတယ်။ အဲဒါ constructor function ထဲမှာ တော့ Title Bar အတွက် text တွေကို ရေးသားထားပါတယ်။ ဒါကြောင့်

Super ("Demonstrate the borderlayout");  
လို့ ရေးသားထားတာနော်။ တစ်နည်းပြာရရှင်တော့ ဒါဟာ JFrame Class ကို extends လုပ်ထားတဲ့အတွက်ကြောင့် ပါပဲ။ တစ်နည်းပြာရရှင်တော့ extends မလုပ်ဘဲနဲ့ main () function ထဲမှာမှ JFrame class ကို ခေါ်ယူပြီးတော့ အောက်ပါအတိုင်း object ဖွဲ့ပြီးတော့ ရေးရှင်လည်း ရတယ်နော်။

JFrame f = new JFrame ();



F.setTitle ("Demonstrate the borderlayout");  
 လို့ ရေးရင်လည်း ရပါတယ်။ အတူတူပါပဲ။ ဒါပေမယ့် constructor function ထဲမှာ super ()method ကို အသုံးချ တာက ပိုပြီးတော့ အဆင်ပြေပါတယ်။

J။ အဲဒီနောက် လိုင်းနံပါတ် (၁၄)မှာတော့ setSize () method ကို အသုံးပြုပြီးတော့ JFrame Windows ရဲ့ အချက်အစား ကိုပါ ထည့်သွင်းပါတယ်။ အဲဒီနောက် setDefaultCloseOperation Close Operation ( ) method ကို အသုံးချပါတယ်။ ဒါဟာ အရေးသင်ခန်းစာတွေမှာ အသုံးပြုခဲ့တဲ့ type တွေပါပဲ။ လိုင်းနံပါတ် (၁၆)မှာတော့ JPanel class ကို ခေါ်ယူပြီးတော့ Object\_Name pane လို့ အမည်ပေး ဖွဲ့စည်းပါတယ်။ အဲဒီနောက်မှာတော့ JPanel Class ထဲက add () method ကို အသုံးပြုပြီးတော့ borderlayout မှာ အသုံးပြုမယ့် JButton တွေကို ထည့်သွင်း တပ်ဆင်ပါတယ်။ လိုင်းနံပါတ် (၁၉) ကနေ လိုင်းနံပါတ် (၂၂) အထိပါပဲ။ အဲဒီမတိုင်ခင်မှာ လည်း လိုင်းနံပါတ် (၁၇) မှာ borderlayout class ရဲ့ constructor method ကို ခေါ်ပြီးတော့ အသုံးချထားပြီး သားနော်။ ဒါအပြင် setLayout () method ကို အသုံးချပြီး တော့ JPanel Class ထဲကို borderlayout ကို ထည့်သွင်းပြီးသားနော်။ ပြီးတော့ လိုင်းနံပါတ် (၂၉)မှာတော့ main () function ကို ရေးသားလိုက်ပြီးနော်။ ပြီးတော့ အပေါ်မှာ ဖန်တီးလိုက်တဲ့ borderlayout class ကို ခေါ်ယူလိုက်ပါ တယ်။ ပြီးတော့ extends လုပ်ထားတဲ့ JFrame Class ထဲက show () method ကို ခေါ်ယူ အသုံးပြုလိုက်တာပါပဲ။ အဲဒီနောက် main () function ရဲ့ close brace} နဲ့ primary class borderlayout ရဲ့ close brace} တွေကို အသီးသီး လာပြီးတော့ ပိတ်ထားတာပါပဲ။

ဒီ program ကို လေ့လာကြည့်မယ်ဆိုရင်တော့ အရေးက FlowLayout, GridLayout တို့နဲ့ သိပ်ပြီးတော့ အထွေအထူး ကွားမှာ မရှိပါဘူး။ အလွယ်တကူ နားလည် သဘောပါက်နှင့် ကြမှာဝါ။ ကဲ ဒီတော့ ဒီ Code\_Line ကို Notepad ထဲမှာ သွားပြီးတော့ ရိုက်သွင်းပါ။ ပြီးရင် C:\j2sdk 1.4.1\_01\bin ထဲကို သွားပြီးတော့ saveလုပ်ရပါမယ်။ Saveလုပ်ရမယ့် file\_Name ကတော့ borderlayout.java လို့ အမည်ပေးရမှာပါပဲ။

Notepad ထဲမှာ code\_line တွေ ရေးသားပြီးတော့ borderlayout.java လို့ အမည်ပေးပြီးတော့ သိမ်းဆည်း save လုပ်ထားတာကို figure 4.11 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ စာဖတ်သူအနေနဲ့ လေ့လာကြည့်လိုက်ပါဉိုး။

```

borderLayout.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

class borderLayout extends JFrame{
    JButton a = new JButton("North");
    JButton b = new JButton("South");
    JButton c = new JButton("East");
    JButton d = new JButton("West");
    JButton e = new JButton("Center");

    borderLayout(){
        super("Demonstrate the borderLayout");
        setSize(240,280);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        BorderLayout obj = new BorderLayout();
        pane.setLayout(obj);
        pane.add(a,"North");
        pane.add(b,"South");
        pane.add(c,"East");
        pane.add(d,"West");
        pane.add(e,"Center");
        setContentPane(pane);
    }

    public static void main(String[] arguments){
        JFrame frame = new borderLayout();
        frame.show();
    }
}

```

Figure 4.11 Write the program code\_line, borderlayout.java

အခုဖော်ပြထားတဲ့ figure 4.11 ကတော့ notepad program ထဲမှာ code\_line တွေ ရေးသားထားတဲ့ပါပဲ။ ဒီတော့ အဲဒီလိမ့်း code\_line တွေ ရေးသားပြီးသွားရင် command prompt (DOS) ကို ဖွင့်ပါ။ အဲဒီအခါမှာ အောက်ပါ command\_line ကို ရိုက်သွင့်ပြီးတော့ Directory ကို ဖြောင်းပါ။ Java Compiler ထဲကို ဝင်ဖို့ပေါ်နော်။

Cd c:\J2sdk 1.4.1\_01\bin ↵

ဒါဆိုရင်တော့ command windows ထဲမှာရှိတဲ့ prompt သက်တယာ C:\j2sdk 1.4.1\_01\bin ထဲကို ရောက်နေပြီးလို့ လာပြီးတော့ ပြရင် အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac borderLayout.java ↵

က ဒီလိမ့်း compile လုပ်လိုက်လို့ error မရှိဘူးဆိုရင်တော့ အောက်ပါ command နဲ့ တစ်ခါပြန်ပြီးတော့ Run လုပ်လိုက်ပါ၏။

java borderLayout ↵

အခုလိမ့်း command prompt ထဲကို ဝင်ပြီးတော့ Java compiler ထဲမှာ compile & run လုပ်တဲ့ပဲကို figure 4.12 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။ Code\_Line တွေမှာ error မရှိဘဲနဲ့ compile & run လုပ်တဲ့ပဲကို ရှင်းပြပေးထားတာပါပဲ။ အမှန်တကယ်ကတော့ compile လုပ်စဉ်မှာတောင် error တွေ အများကြီး ရှိလာနိုင်ပါတယ်။ အဲဒါတွေ ကို အထပ်ထပ်အခါခါ ရှင်းလိုက်၊ Save As ကို notepad မှာ



ပြန်လည်၊ ပြီးရင် compile ပြန်လည်ပြီးမှ error မရှိမှ run ရမှာပါပဲ။  
က ဒါဆိုရင်တော့ figure 4.12 ကို လေ့လာကြည့်ပါ။

```
C:\> Command Prompt - java borderLayout
C:\>j2sdk1.4.1_01\bin>javac borderLayout.java
C:\>j2sdk1.4.1_01\bin>java borderLayout
```

Figure 4.12 Compile & Run the borderlayout .java in the Command Prompt.

ဒီအခါမှာတော့ figure 4.12 ကအတိုင်း Compile & Run လုပ်လိုက်ရင် အောက်မှာ ဖော်ပြထားတဲ့အတိုင်း program ရဲ့ output console windows ဟာ ဖော်ပြပေးလာမှာပါ။ Desktop screen ရဲ့ upper-left corner မှာပါပဲ။ အစွယ်တက္က တွေ့မြင်နိုင်မှာပါ။ ဒီတော့ output windows ကို figure 4.13 အနေနဲ့ ရေးသားပေးထားတာကို လေ့လာလိုက်ပါဉိုးလား။ အရှေ့၊ က theory & structure ရှင်းစဉ်က DEMO figure ပုံစံပါပဲ။ လေ့လာကြည့်ပါ။



Figure 4.13  
Run the  
program,  
borderlayout.java

က ဒါဆိုရင်တော့ layout type တွေနဲ့ ပတ်သက်တဲ့ အကြောင်းအရာတွေကို အခြေခံလောက်တော့ သဘောပေါက်သွားလောက်ပါပြီ။ ဘယ်လိုနေရာမျိုးမှာ ယဉ်တွဲပြီးတော့ အသုံးချမယ် ဆိုတာကို တွေးဆြြးတော့ အသုံးပြနိုင်မှာပါပဲ။ က ဆက်လက်ပြီးတော့ Index 4.5 Mixing layout Managers and components ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် လေ့လာကြည့်ကြရ အောင်။

KYAW ZAYAR LAY [ IT ENGINEERING ]



### Laptop ကွန်ပျူတာကို အေးစေနို့ရာ

Laptop ကွန်ပျူတာများရဲ့ တည်ဆောက်ပုံအရ အအေးပေးစနစ်များ ထပ်မံ တပ်ဆင်ရန် မလွှယ်ပါ။ တစ်နည်းအားဖြင့် ပန်ကာ အပိုတပ်နှုံးရာ နေရာမရှိပါ။ အပူသက်သာအောင်တော့ သတိပြု သုံးစွဲနိုင်ပါတယ်။ ဥပမာအားဖြင့် မျက်နှာပြင် ညီညာတဲ့ စားပွဲပေါ်မှာတင်ထားတာမျိုး၊ လေဝင်လေထွက်ကောင်းမွန်တဲ့ နေရာမှာ သုံးတာမျိုးပါ။ ဒါပေမယ့် ကွန်ပျူတာ အပိုပစ္စည်း ထုတ်လုပ်သူတွေက အဆိုပါ အပူဒဏ်ကို သက်သာစေနို့ ပစ္စည်းများကို ထုတ်လုပ်ကြပါတယ်။ ငါးတွင် Belkin Cooling stand ဆိုတဲ့ laptop တင်တဲ့ စင်ကလေးလည်း အပါအဝင် ဖြစ်ပါတယ်။ ငါးကို USB port ကတစ်ဆင့် laptop နဲ့ ဆက်သွယ်ထားပြီး ပန်ကာကို လည်းစေပါတယ်။ Belkin Cooling stand ရဲ့ ဂုံးကို အပြားငယ်လေးဖြစ်လို့ လေဝင်လေထွက်အတွက် ကောင်းမွန်စေသလို သယ်ယူရာမှာလည်း အပန်းမကြီးစေပါ။ Laptop ထည့်တဲ့ အိတ်ထဲကို ရောထည့်နိုင်ပါတယ်။ ငါးတွင် တပ်ဆင်ထားတဲ့ ပန်ကာက လေယူတဲ့အား ကောင်းပေမယ့် ပါဝါစွမ်းအင်ကို သိပ်မသုံးဘူးလို့ဆိုပါတယ်။ ပါဝါအတွက် adapter လည်းဆုံးစုံရာမလိုပါ။ Stand တည်ဆောက်ပုံက သင့်အတွက် laptop ကို သက်သက်သာသာနဲ့ သုံးနိုင်ရန် ကျေန်းမာရေးနည်းလမ်းများနဲ့ တွက်ချက် တည်ဆောက်ထားတာပါ။ အသေးစိတ်အချက်အလက်များကို [www.belkin.com](http://www.belkin.com) သို့ ဆက်သွယ် စုစုပေါင်းနိုင်ပါတယ်။



# JAVA DEVELOPER GUIDE

## Chapter (4)

### 4.5 MIXING LAYOUT MANAGERS AND COMPONENTS

အခုလက်ရှိ သင်ကြားပို့ချပေးခဲ့တဲ့ သင်ခန်းစာတွေမှာ Layout Managers အကြောင်းတွေကို အခြေခံကျကျ အသုံးပြုခြုံပါ။ ဒီတော့ သင်ကြားပြီးသမျှ သင်ခန်းစာတွေထက် swing components တွေရယ်၊ layout managers တွေရယ်ကို ယူဉ်တဲ့ အသုံးပြုခြုံတော့ Application programs ထဲလေးတစ်ခု လောက် ဖန်တီးကြည့်ကြရအောင်လား။ ဒိဖန်တီးမယ့် application မှာ JLabel, JTextField အစရိတဲ့ Swing components တွေရယ်၊ ပြီးခဲ့တဲ့ standard Dialog အပ်စုဝင် showInputDialog( ) method, Show OptionDialog( ) method အစရိတဲ့ method တွေရယ်၊ အခု Chapter(4) မှာ ပါဝင်လာတဲ့ GridLayout class တွေရယ်ကို ပေါင်းစပ်ပြုးတော့ program တစ်ခုအနေနဲ့ ဖန်တီးမှာပါပဲ။

Program Flow\_type ကတော့ program Run ပြီဆိုတာ နဲ့ Enter the Site Title: ဆိုတဲ့ ShowInputDialog Box တစ်ခု ပေါ်လာပြီး လေ့လာမယ့် Web\_Site Title & Topics ကို လာပြီးတော့ Input\_Data လုပ်ခိုင်းမှာပါပဲ။ ဟုတ်ပြီ။ Data\_Input လုပ်ပြီးတော့ OK Button ကို နှိပ်လိုက်မယ်ဆိုရင်

Enter the Site Address: ဆိုတဲ့ ShowInputDialog Box တစ်ခုက ထပ်မံပြီးတော့ Data ကို တောင်းလာမှာပါပဲ။ အဲဒီ မှာ သွားပြီးတော့ လေ့လာမယ့် Web-Site Address ကို ရှိက်သွင်းပြီးတော့ OK Button ကို နှိပ်လိုက်မယ်ဆိုကြပါစို့။

အဲဒီအခါမှာ ကျွန်ုတ်တို့ လေ့လာမယ့် website ရဲ့ type အမျိုးအစားပေါ့နော်။ Personal Type လား၊ Commercial Type လား၊ ဒါမှမဟုတ် Unknow Type ပဲလားလို့ မေးလာတဲ့ Site Type ဖြစ်တဲ့ ShowOption Dialog Box ဟာ လာမေးလာမှာပါ။

အဲဒီအခါမှာတော့ စာဖတ်သူလေ့လာမယ့် web-site ရဲ့

Title, Address, Type တွေ ပါဝင်တဲ့ Site Information windows ဟာ စာဖတ်သူရဲ့ ဖြည့်သွင်းလိုက်တဲ့ အချက်အလက် Input\_Data တွေကို အမို့ပြုပြီးတော့ Output ထဲတံ့လာပေးမှာ ပါပဲ။ ဟုတ်ပြီနော်။ ဒီတော့ ဖန်တီးရေးသားမယ့် Application ရဲ့သဘောတရားကို နားလည်သွားရင်တော့ အရင်ဆုံး code\_line တွေကို ချရောပြီးတော့ လေ့လာကြရအောင်။ Listing 4.5 Mixing Layout Managers and Components ဆိုတဲ့ program ကို လေ့လာကြရအောင်လား။

**Listing 4.5 Mixing Layout Managers and Components**

```
1. import java.awt.GridLayout.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. public class info extends JFrame{
6. private JLabel titlelabel = new JLabel("Title:",
7. SwingConstants.RIGHT);
8. private JTextField title;
9. private JLabel addresslabel = new JLabel("Address;",
10. Swing constant RIGHT);
11. private JTextField address;
12. private JLabel typelabel = new JLabel("Type:",
13. swing constants.RIGHT);
14. private JTextField type;
15.
16. public info() {
17. super ("Site Information");
18. setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
```



```

19. setLookAndFeel( );
20.
21. //Site name
22.
23. String response1 = JOptionPane. Show
       Input Dialog (null,
24. "Enter the Site Title:");
25. title = new JTextField (response 1,
   20);
26.
27. //site address
28. String response2 = JOptionPane. ShowInputDialog (null,
29. "Enter the Site Address: ");
30. address=new JTextField (response2, 20);
31.
32. //site type
33.
34 String [] Choices = {"Personal",
   "Commercial", "Unknown"};
35. int response 3 = JOptionPane>ShowOption
       Dialog (null,
36. "What type of site is it?",
37. Site Type",
38. 0,
39. JOptionPane.QUESTION_MESSAGE,
40. null,
41. choices,
42. choices [0]);
43. type = new JTextField (choices
   [response3], 20);
44.
45. JPanel pane = new JPanel ( );
46. pane.setLayout (new GridLayout (3,2));
47. pane.add (titlelabel);
48. pane.add (title);
49. pane.add (addresslabel);
50. pane.add (address);
51. pane.add (typelabel);
52. pane.add (type);
53. SetContent Pane (pane);
54. Pack ();
55. SetLookAndFeel ( );
56. SetVisible (true);
57. }
58. private void SetLookAndFeel ( ){
59. try {
60. UIManager.SetLookAndFeel (
61. UIManager, getSystemLookAndFeel
      ClassName());
62. SwingUtilities- update ComponentTreeUI
      (this);
63. }Catch (Exception e){
64.
65. /* System. err. PrintIn ("Couldn't use
      the System"
66. + "Look and Feel:" + e); */
67.
68. }
69. }
70. public static void main (String [

```

```

    arguments) {
71. Info frame = new Info ( );
72. }
73. }

```

### Analysis

- ၁။ ကဲ code\_line (72) ကြောင်းပါဝင်တဲ့ လက်ရှိ program ကို analysis လုပ်ကြည့်ကြအောင်။ ဒီ program မှာ awt. GridLayout; package ကိုထိပ်ဆုံးမှာ ခေါ်ယူပြီး Declare လုပ်ထားပါတယ်။ ဒါဟာ GridLayout Class Manager ကို အသုံးချချင်လိုအင်။ ကျေနိတဲ့ package file (2) ရှာတော့ အရင် program တွေမှာမြင်တွေဖြူးသား package တွေပါပဲ။ လိုင်းနံပါတ် (၅) မှာတော့ Info ဆိုတဲ့ primary class ကို Declare လုပ် ဖန်တီးပါတယ်။ JFrame Class ကို extendsလုပ်းပေါ်လည်း ဖန်တီးထားတာအင်။ လိုင်းနံပါတ် (၆)ကနေ လိုင်းနံပါတ် (၁၄) အထိမှာတော့ private keyword ကို အသုံးပြုပြီးတော့ swing components တွေကို Variable\_Name အနေနဲ့ ခေါ်ယူ အသုံးပြုပါတယ်။ သက်ဆိုင်ရာ class ရဲ့ constructor method အနေနဲ့ပါပဲ။ လေ့လာကြည့်မယ်ဆိုရင် JLabel (1) ခုကို titlelabel လို့ object\_variable\_name ပေးမယ်။ အဲဒီ JLabel Box မှာပါဝင်ရမယ့် Caption\_text ကို JLabel () constructor method ရဲ့ parameter\_argument နေရာမှာ ထည့်သွင်းရေးသားပါတယ်။ Comma (,) ခံပြီးတော့ Swing constants.RIGHT လို့ ရေးသားထားတဲ့ ဒုတိယ arguments ကတော့ alignment လုပ်ဖို့အတွက်ပါပဲ။ လိုင်းနံပါတ် (၈) မှာတော့ JTextField (1) ခုကို title လို့ object\_variable\_name ပေးပြီးတော့မှ ဖန်တီး ခေါ်ယူပါတယ်။ ဒါကြောင့် code\_line အနေနဲ့
- ```

private JLabel titlelabel = new JLabel
("Title:" Swing Constants.RIGHT);
private JTextField title;

```
- လို့ ရေးသားထားတာပါပဲ။ ဒါဟာ Output JFrame\_Windows မှာ ပါဝင်ရမယ့် JLabel & JTextField တွေ အတွက်ပေါ်အင်။ ဒီတော့ ကျေနိတဲ့ code\_line တွေကို ဆက်လက် လေ့လာကြည့်လိုက်ပါ။ String arguments နေရာမှာ ပြောင်းလဲသွားပေမယ့် အတူတူပါပဲ။ လေ့လာရ လွယ်ကူပါလိမ့်မယ်။
- ၂။ လိုင်းနံပါတ် (၁၆) မှာတော့ လက်ရှိ Info primary class ရဲ့ constructor function ကို ရေးသားလိုက်ပါတယ်။ ဒီတော့ JFrame\_Windows မှာ ပါဝင်ရမယ့် Title Text အတွက်ပါပဲ။ လိုင်းနံပါတ် (၁၇)မှာ ရေးသားထားပါတယ်။



```

    Super ("Site Information");

က ဟုတ်ပြုနော်။ ပြီးတော့ လိုင်းနံပါတ် (၁၈)နဲ့ (၁၉)မှာတော့
SetDefault CloseOperation ()method နဲ့ SetLook
AndFeel () method တွေကို အသုံးချေပေးထားတာပါပဲ။
အဲဒီနောက် လိုင်းနံပါတ် (၂၃)မှာကတော့ ShowInput-
Dialog () method တစ်ခုကို ဖန်တီးပြီးတော့ အသုံးချေပါ
တယ်။ ဒီနေ့ရာမှာတော့ ပြီးခဲ့တဲ့ chapter (၃)မှာတင် သင်
ကြေားနဲ့တာဆိုတော့ အထွေအထူး ရှင်းပြစ်ရာ မလိုတော့ပါ
ဘူး။ လိုအပ်လာရင် Chapter (၃)ကို ပြန်လှန်လေ့လာပါ။
ဒါကြောင့် လိုင်းနံပါတ် (၂၃) (၂၄)မှာ-
String response1=JOptionPane. Show-
Input- Dialog (null, Enter the site
Title:");

လို့ ရေးသားထားတာပါပဲ။ လိုင်းနံပါတ် (၂၅)မှာ ရေးသား
ထားတဲ့

title = new JTextField (responens 1,20);
ကတော့ ShowInputDialog () Box ထဲမှာ လက်ခံလိုက်
တဲ့ String Input ကို JFrame_Windows အနေနဲ့ output ထွက်လာရင် သွားပြီးတော့ ဖော်ပြပေးဖို့အတွက် အပေါ်
က Declare လုပ်ခဲ့တဲ့ လိုင်းနံပါတ် (၈)က title ဆို တဲ့
object_variable_name ကို ခေါ်ယူ အသုံးပြုလိုက်တာပါ
ပဲ။ ထောက်ကြော အဲဒီ title ဘယ်က ဝင်လာတာလဲလို့ complex
ဖြစ်သွားမှုစုံလိုပါ။ သေသေချေချာ လေ့လာကြည့်နော်။
အဲဒီနောက် လိုင်းနံပါတ် (၂၀)မှာတော့ နောက်ထပ် Show-
Input_Dialog () တစ်ခုကို ထပ်မံ ဖန်တီးပါတယ်။ အဲဒီ
Input-Dialog Box ကတော့ site address ကို လက်ခံစိုး
အတွက်ပါပဲ။ ဒါကြောင့်

String response 2 = JOptionPane.
ShowInputDialog (null, "Enter the site
Address:");

လို့ ရေးသားထားတာနော်။ က ဒီလိုစိုရင် လိုင်းနံပါတ် (၂၀)
မှာ ရေးသားထားတဲ့

address=new JTextField (response 2,20);
ဆိုတာကတော့ အပေါ်က လိုင်းနံပါတ် (၁၁)မှာ ဖန်တီးခဲ့တဲ့
JTextField ထဲမှာ ပါဝင်ရမယ့် String အတွက် variable
အနေနဲ့ ညီယူထားတာပါပဲ။ ဟုတ်ပြုနော်။ အပေါ်မှာက
object_variable လုပ်ခဲ့တာကတွဲလို့ တိုက်ရှိက် အသုံးမချေခဲ့
ပါဘူး။ အဲဒီကို သာတော်ပေါက်ပြုနော်။ ပြီးတော့ လိုင်းနံပါတ်
(၂၄) မှာတော့ Demisional type ကို အသုံးပြပြီးတော့ choices ဆိုတဲ့ variable_name ကို ဖန်တီးပါတယ်။ ပါဝင်
တဲ့ Array_Data တွေကတော့

```

Choices [0] = "Personal"

Choices [1] = "Commerical"

Choices [2] = "Unknown" တို့ပဲ ဖြစ်ကြပါတယ်။
ဒါကြောင့် အောက်ပါအတိုင်း ရေးသားထားတာပဲ ဖြစ်ပါ
တယ်။

```

String [] choices = {"Personal",
"Commerical", "Unknown"};

```

၃။ အဲဒီနောက် လိုင်းနံပါတ် (၂၅)မှာတော့ ShowOption
Dialog Box တစ်ခုကို ခေါ်ယူ အသုံးချထားပါတယ်။ အဲဒီ
ShowOption\_Dialog Box မှာ ပါဝင်ရမယ့် arguments
တွေကိုတော့ မရင်းပြတော့ပါဘူးနော်။ လိုင်းနံပါတ် (၂၅)
ကနေ လိုင်းနံပါတ် (၄၂)အတိုင်း ကိုယ့်ပါယူလိုက် တစ်ကြောင်းတည်း အစအဆုံး ရေးမယ်
ဆိုရင်လည်းရတယ်နော်။ ကျွန်ုတ်ကတော့ ရှင်းပြရွယ်ကူ
အောင် Line\_By\_Line ရေးပြပေးထားတာပါပဲ။

လိုင်းနံပါတ် (၄၃) မှာတော့ အပေါ်က Declare လုပ်ခဲ့တဲ့
JText-Field ထဲကို Input လုပ်ဖို့အတွက် ညီယူထားတာပါ
ပဲ။ ဖန်တီးထားတဲ့ variable\_name ကတော့ type နော်။
အဲဒီပြီးတော့ လိုင်းနံပါတ် (၂၅)မှာတော့ JPanel Class
ကို pane ဆိုတဲ့ object\_name ပေးပြီးတော့ ခေါ်ယူပါ
တယ်။ အဲဒီနောက် JPanel Class ထဲက SetLayout ()
method ကို ခေါ်ယူ အသုံးချပါတယ်။ အဲဒီအခါမှာ parameter\_arguments အနေနဲ့

```

java.awt.Gridlayout.*;

```

package ထဲက GridLayout () method ကို ထည့်သွင်း
အသုံးချထားပါတယ်။ အခြားသော Layout Manager
တွေနဲ့ ရောထွေးနိုင်တာမျို့ package file ကို Declare
လုပ်စဉ်ကတည်းက GridLayout class ကို တိုက်ရှိက်
ခေါ်ယူထားတာပါပဲ။ လိုင်းနံပါတ် (၄၇) ကနေ လိုင်းနံပါတ်
(၅၂)အတိုင်းတော့ JPanel class ထဲက add () method
ကို ခေါ်ယူ အသုံးချထားတာပါပဲ။ အဲဒီ add () method
နဲ့ arguments ထဲမှာ အပေါ်မှာ ဖန်တီးထားတဲ့ JLabel
Box တွေရယ်၊ JTextField တွေရယ်ကို ထည့်သွင်း တည်
ဆောက်တာပါပဲ။ အဲဒီနောက်မှာတော့ ဖန်တီးထားတဲ့ object
components တွေဟာ JFrame windows ပေါ်ကို ရောက်
ရှိလာအောင်လို့ SetContentPane () method ရယ်၊ pack
() method ရယ်ကို ခေါ်ယူ အသုံးချပါတယ်။ လိုင်းနံပါတ်
(၅၅) ကတော့ SetVisible\_ () method ကို အသုံးချထား
တာပါပဲ။ လိုင်းနံပါတ် (၅၆) ကတော့ Info () ဆိုတဲ့ constructor
function close brace } ပါပဲ။ ဒီအထိ ပြန်ပြီး
တော့ လေ့လာကြည့်ပါတီး။

၄။ အဲဒီနောက်မှာတော့ အပေါ်မှာ useလုပ်ခဲ့တဲ့ SetLookAndFeel ( ) method ကို လေ့လာကြည့်ရအောင်လား။ အဲဒီ SetLookAndFeel ( ) method ကို လိုင်းနံပါတ် (၅၇) ကနေ လိုင်းနံပါတ် (၆၈) အထိ ရေးသားပေးထားပါတယ်။ ရေးသားရာမှာ Exception Handling အနေနဲ့ ရေးသားတာပဲ ဖြစ်ပါတယ်။

**Note :** Exception Handling နဲ့ ပတ်သက်တဲ့ သင်ခန်းစာ တွေကို မသင်ကြားရသေးတဲ့အတွက် ရှိ chapter တွေမှာပဲ သေသေချာချာ ရှင်းပြပေးပါမယ်။ အလွယ်တကူ ပြောရရင်တော့ အောက်ပါ try-catch block ထဲကို ဝင်ပြီးတော့ အလုပ်လုပ်တယ် လို့ပဲ မှတ်ထားကြတာပေါ့။

#### Sample

```
try {
    Float in = Float.parseFloat(input);
} catch (NumberFormatException info) {
    System.out.println (input + "is not a
valid number.");
}
```

တကယ်လို့ multiple catch blocks တွေကို single try နဲ့လည်း ယူဉ်လွှာပြီးတော့ အသုံးချွန်ပါတယ်။ လေ့လာကြည့်ပါ။

#### Sample

```
try{
    //code that might generate exceptions
}catch (IOException e){
    // handle IO exceptions
}catch (ClassNotFoundException e){
    // handle class not found exceptions
}catch (InterruptedException e) {
    // handle interrupted exceptions
}
```

က ဒီတော့ ကျွန်တော်ကလည်း ဒီ Exception Handling method ကို မသင်သေးဘူး။ ရှင်းလည်း မရင်းပြရသေးဘူးဆိုတော့ ဘယ်လိုလုပ်မလဲ။ လွယ်ပါတယ်။ ရေးထားတဲ့ code\_line တွေကို ကြည့်ပါ။

Code\_line No;(58) ကနေ code\_line No: (69) အထိ ရေးသားထားတဲ့ Private Void SetLookandFeel () method တစ်ခုလုံးကို selection\_color ခြယ်ပြီးတော့ ဖျက်ပစ်လိုက်ပါ။ ဟုတ်ပြုနော်။ { } တွေ အပါအဝင်အပေါ့နော်။ က ပြီးသွားရင် လိုင်းနံပါတ် (၁၉) မှာ ရေးသားထားတဲ့ SetLookAndFeel () method ကိုလည်း selection\_color ခြယ်ပြီးတော့ ဖျက်ပစ်လိုက်ပါပြီး။ ပြီးတော့ လိုင်းနံပါတ် (၅၅) မှာ ရေးသားထားတဲ့ SetLookAndFeel () method ကိုလည်း ဖျက်ပစ်လိုက်ပါပြီး။ အားလုံး သုံးခါတိတိ ဖျက်ပစ်ရမှာပါပဲ။

က SetLookAndFeel ( ) function ကိုလည်း ဖျက်ပြီးတော့ ခေါ်သုံးထားတဲ့ code\_line တွေကိုလည်း ဖယ်ရှားပြီးတော့ ဖျက်ပစ်လိုက်ပါ။ ဒီတော့ code\_line တွေထဲမှာ ပါဝင်တဲ့ SetLookAndFeel ( ) method နဲ့ ဆက်နွယ်နေတဲ့အချက်တွေဟာ ပါဝင်တော့ဘူးဆိုတော့ program ပေါ်မှာ ဘယ်လို အကျိုးသက်ရောက်မျှေးသလဲဆိုတာကို ဆက်လက် လေ့လာကြရအောင်လား။

၅။ လိုင်းနံပါတ် (၇၀)မှာတော့ main () function ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၇၁)မှာတော့ အပေါ်မှာ ရေးသားထားတဲ့ Info ဆိုတဲ့ Class ကို Object \_ Name Frame လို့ပေးပြီးတော့ ခေါ်ယူ လိုက်တာနဲ့ Info Class နဲ့ constructor function ဟာ အလိလိ ဝင်ရောက်လာပါပြီ။ ဒီတော့ program ရဲ့ တည်ဆောက်မှု flow ဟာ ကျွန်တော်တို့ရှုပ်တဲ့အတိုင်း ဖြစ်လာပြီဖော်။ လိုင်းနံပါတ် (၇၂)နဲ့ (၇၃) ကတော့ main () function နဲ့ primary class တို့ရဲ့ close brace } တွေပဲ ဖြစ်ပါတယ်။

က ဒီတော့ code\_line တွေကို မူလအတိုင်းပဲရေးရေး၊ SetLookAndFeel ( ) method နဲ့ ပတ်သက်တာတွေကိုပဲ ဖြေတဲ့ချုပြီးတော့ပဲ ရေးရေး အဲဒီ program ကို Compile & Run လုပ်ပြီး လေ့လာကြရအောင်လား။ ကျွန်တော်ကတော့ code\_line နှစ်ချိုးစလုံးနဲ့ Run လုပ်ပြီးတော့ program တည်ဆောက်ပြထားပါတယ်။ ဘယ်လိုက္ခာခြားမှုရှိတယ်ဆိုတာကို စာဖတ်သူကိုယိုင်ပဲ compare လုပ်ပြီးတော့ လေ့လာကြည့်ပါနော်။

အရင်ဆုံး notepad ထဲကို သွားပြီးတော့ code\_line တွေကို ရေးသားပါ။ Code\_line အရေးအတွက်က အနည်းငယ်များတဲ့ အတွက် အမှားအယွေးမရှိအောင် သတိထားပြီးတော့ ရေးသားပါ။ SetLookAndFeel ( ) method တွေကို မထည့်ချင်ရင် တစ်ခါတည်းထည့်မရေးခဲ့ပါနဲ့။ ကျွန်တော်ကတော့ ထည့်သွင်းရေးသားပြီးတော့ program ကို တည်ဆောက်စေချင်ပါတယ်။ ပြီးတော့မှ တစ်ခါ ပြန်ဖြတ်ချေပြီးတော့မှ ပြန် Run ကြည့်စေချင်ပါတယ်။ ဒါမှ ကွာခြားချက်ကို လက်တွေ့ လေ့လာ သဘောပေါ်တော့မှာ ပျော်နော်။

အဲဒီ code\_line တွေကို ရေးသားပြီးသွားပြီဆိုရင်တော့ File → save ကို တောင်းပါ။ Save Dialog Box ကျလာရင် ပေးရ မယ့် File\_Name ကတော့ Info.java ပါပဲ။ Java Program ရေးရာမှာ အဓိကအချက်ကတော့ File\_Name နဲ့ primary class\_name ဟာ အတူတူဖြစ်ရမှာပါပဲ။

က ဒီတော့ အဲဒီ code\_file ကို ထားရမယ့် Directroy\_path ကတော့ C:/j2sdk 1.4.1\_01 bin > အောက်မှာနော်။ အဲဒီဆိုရင်တော့ စာဖတ်သူအနေနဲ့ အလွယ်တကူ ရေးသားပြီးတော့ save လုပ်နိုင်မှာပါပဲ။ အဲဒီလို့ code\_line တွေ ရေးသားပြီးတော့



save လုပ်ထားပုံကို figure 4.14 မှာ ရေးသား ဖော်ပြပေးထားပါတယ်။ စာဖတ်သူအနေနဲ့ လေ့လာကြည့်ပါ။

**Note :** Code\_line တွေ ပါဝင်တဲ့ program file တွေကို ရှုပ်တယ်ဆိုရင်တော့ ယခုစာအပ်နဲ့အတူ ပါဝင်လာတဲ့ CD ထဲကနေ ပြီးတော့မျပ်ဖြစ်ဖြစ်၏ ကွန်ပူးတာဂျာနယ်မှာပဲဖြစ်ဖြစ် လာပြီးတော့ မှ JAVA DEVELOPER Guide ရဲ့ code\_File CD တွေကို လာရောက် ကူးယူနိုင်ပါတယ်။ သက်ဆိုင်ရာ Chapter အလိုက် Folder လေးတွေ ခွဲပြီးတော့ တည်ဆောက် ဖန်တီးပေးထားပါတယ်။ အလွယ်တကူ လေ့လာနိုင်အောင် ဖန်တီး တည်ဆောက်ပေးထားတာပါပဲ။

```

Info.java - Notepad
File Edit Format View Help
//Listing 4.5 Mixing Layout Managers and Components
import java.awt.GridLayout;
import java.awt.event.*;
import javax.swing.*;

public class Info extends JFrame{
    private JLabel titleLabel = new JLabel("Title: ", SwingConstants.RIGHT);
    private JTextField title;
    private JLabel addressLabel = new JLabel("Address: ", SwingConstants.RIGHT);
    private JTextField address;
    private JLabel typeLabel=new JLabel("Type: ", SwingConstants.RIGHT);
    private JTextField type;

    public Info() {
        setTitle("Enter Information");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setLookAndFeel();
        // Site name
        String response1 = JOptionPane.showInputDialog(null, "Enter the Site Title: ");
        title = new JTextField(response1,20);
        //site address
        String response2 = JOptionPane.showInputDialog(null, "Enter the Site Address: ");
        address = new JTextField(response2,20);
        // Site type
        String[] choices = {"Personal", "Commercial", "Unknown"};
        int response3 = JOptionPane.showOptionDialog(null, "What type of site is it?", "Site Type", 0, JOptionPane.QUESTION_MESSAGE, null, choices, choices[0]);
        type = new JTextField(choices[response3],20);
        JPanel pane = new JPanel();
        pane.setLayout(new GridLayout(3,2));
        pane.add(titleLabel);
        pane.add(title);
        pane.add(addressLabel);
        pane.add(address);
        pane.add(typeLabel);
        pane.add(type);
        setContentPane(pane);
        pack();
        setLookAndFeel();
        setVisible(true);
    }
    private void setLookAndFeel() {
        try {
            UIManager.setLookAndFeel(
                UIManager.getSystemLookAndFeelClassName());
            SwingUtilities.updateComponentTreeUI(this);
        } catch (Exception e){
            /* System.out.println("couldn't use the System "+ "Look and Feel : " + e); */
        }
    }
    public static void main (String[] arguments) {
        Info frame = new Info();
    }
}

```

Figure 4.14 Write the program code\_line in the Notepad, info.java

အချေဖြစ်ပဲခဲ့တဲ့ Figure 4.14 က အတိုင်း code\_line တွေ ရေးသားပြီးသွားပြီးဆုံးရင်တော့ Start → Program → Accessories → Command Prompt (or) Start → All Program → Accessories → Command Prompt (or) Start → Run ကနေ Run Dialog Box ကျလာရင် Command.com လို့ရှိက်သွေးပါ။ Command Line Interface Run လာတဲ့လို့မယ်။ ကျွန်တော်ကတော့ Desktop Screen ပေါ်မှာ Command Prompt ရဲ့ Shortcut\_Icon လေးကို တင်ထားပြီးတော့ အလွယ်တကူ D-click ခေါက်ပြီးတော့ ဖွင့်လိုက်တာပါပဲ။ အဲဒါကတော့ စာဖတ်သူအနေနဲ့ အဆင်ပြုသလိုပေါ်နေပါ။ Command Line Interface ပွင့်လာရင် အောက်ပါ Command နဲ့ Directory ကို change လုပ်ပါ။

cd c:\ j2sdk1.4.1\_01\ bin ↵

ဒါဆိုရင်တော့ command windows ထဲက prompt သက်တဲ့ အောက်ပါအတိုင်း ပေါ်နေမှာပါပဲ။

c:\j2sdk 1.4.1\_01\bin ↵

အဲဒါလိုမျိုး ပေါ်နေစဉ်မှာ အောက်ပါ Command နဲ့ Info.java ဆိုတဲ့ program ကို compile လုပ်လိုက်ပါ။

javac Info.java ↵

အဲဒါလိုမျိုး compile လုပ်လိုက်လို့ error မရှိဘူးလို့ ဆိုကြပါစိုး။ အောက်ပါ Command\_Line နဲ့ တစ်ခါပြန်ပြီးတော့ Run လုပ်လိုက်ပါ။

javac Info.java ↵

အဲဒါလိုမျိုး compile လုပ်လိုက်လို့ error အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

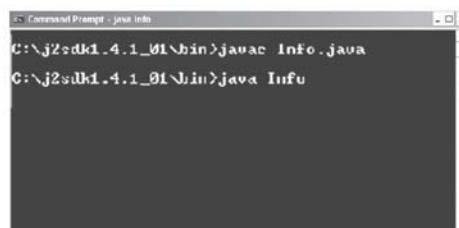
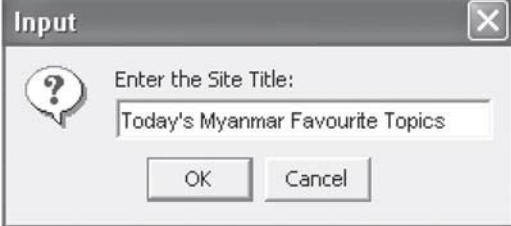


Figure 4.15 Compile & Run of the info.java program

အချေဖြစ်ပဲထားတဲ့ figure 4.15 ကတော့ command prompt ထဲမှာ Info.java ဆိုတဲ့ program ကို Compile & Run လုပ်ပေးတဲ့ ပုံပါပဲ။ ဒါဆိုရင်တော့ Info.java Application

ကွန်ပျူတာသင်ခန်းစာ

ဟာ စတင်ပြီး အလုပ်လုပ်ပါပြီ။ ဒီတော့ ပထမဗြိုးဆုံး error the site title: လို့ မေးလာတဲ့ ShowInputDialog Box ဟာ လာပြီး data\_Input လုပ်နိုင်းပါပြီ။ လေ့လာကြည့်ပါ။



Can't use the SetLookAndFeel ( )method



Using the SetLookAndFeel ( ) method

Site Type

What type of site is it?

Personal    Commercial    Unknown

Using the SetLookAndFeel ( ) method

Figure 4.17 The site address input dialog box

အခါ Figure 4.16 မှာ ဖော်ပြထားတဲ့ Input လို့ Title ပါဝင်တဲ့ Input Dialog Box ထဲမှာ ကျွန်တော် ဖြည့်သွင်းသလို ပဲပေါ့။ Today's Myanmar Favourite Topics လို့ပဲ ဖြည့်လိုက်ပါ။ ဒါမှမဟုတ်လည်း စာဖတ်သူသိချင်တဲ့ အကြောင်းအရာနဲ့ ပတ် သက်တဲ့ Title တစ်ခုခုကိုပေါ်နော်။ Dialog Box ပုံစံ (၂)မျိုး ကွဲနေတာကတော့ စာဖတ်သူ သိတဲ့အတိုင်း SetLookAndFeel ( ) method ကို ရေးသား ထည့်သွင်းတာနဲ့ SetLookAndFeel ( ) method ကို မထည့်သွင်းထားတာပဲကွားပါတယ်။ ကျွန်တော် ကတော့ စာဖတ်သူအနေနဲ့ မျက်စိတ် ရှင်းလင်းသွားအောင် ပုံစံ (၂) မျိုးလုံးကို output ထုတ်ပြထားပါတယ်။ ကဲ ဟုတ်ပြုလော့ အဲဒီလိုမျိုး string data ကို Input လုပ်ပြီးတော့ OK Button နိုင်လိုက်ရင် အောက်မှာဖော်ပြထားတဲ့ Figure 4.17 အတိုင်း Enter the Site Address ဆိုတဲ့ ShowInputDialog Box တစ်ခု ထပ်မံ့မေးလာမှာပါ။ လေ့လာကြည့်ပါ။

Can't use the SetLookAndFeel ( ) method

Can't use the SetLookAndFeel ( ) method

Using the SetLookAndFeel ( ) method

Figure 4.18 The Site Type, Show OptionDialog Box



က အခါလိမျိုး ရွေးချယ်လိုက်ပြီဆိုရင်တော့ အထက်မှာ ရွေးချယ် ဖြည့်စွက်ထားတဲ့ Data တွေ Select လုပ်ခဲ့တဲ့ type တွေ ပါဝင်တဲ့ Site Information တွေတိ ဖော်ပြပေးထားတဲ့ JFrame\_Windows လေးဟာ အောက်ပါအတိုင်း ဖော်ပြပေးလာမှာပါပဲ။ ကျွန်ုတ်တော်ကတော့ ထဲ့စံအတိုင်းပဲ SetLookAndFeel() function ကို အသုံးပြထားတဲ့ program နဲ့ SetLookAndFeel() function ကို အသုံးပြုမထားတဲ့ Program ရယ်ကို နှင့်ယူဉ် ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။ Figure 4.19 အနေနဲ့ပါပဲ။

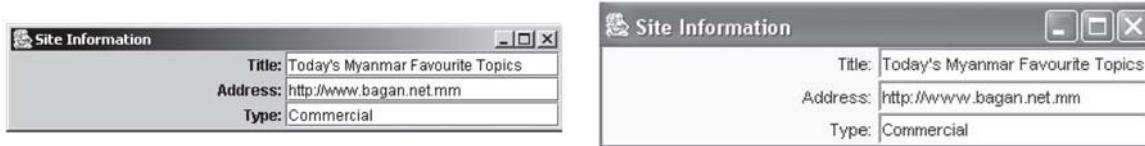


Figure 4.19 The main window of the info application

က ဒီတော့ ကျွန်ုတ်အနေနဲ့ အခု ရှင်းပြနေတဲ့ Layout-Manager တွေနဲ့ ပတ်သက်တဲ့ program ငယ်လေးတွေကို နားလည် သဘောပေါက်သွားပြီဆိုရင်တော့ 4.6 Developing a FrameWork ဆိုပြီးတော့ JFrame နဲ့ JWindows အကြောင်းတွေကို ပို့ချုပ် ထားပါတယ်။ ဆက်လက် လေ့လာကြည့်ပါ။

KYAW ZAYAR LAY [ IT ENGINEERING ]

**သင့် ကွန်ပူးတာတွင် USB1.1 လား၊ USB2.0 လား သိဖို့ရာ**

USB (Universal Serial Bus) port ပေါ်ကာစက USB0.9, 1.0, 1.1 များသာဖြစ်၍ အသုံးလည်း နည်းခါသည်။ ၂၀၀၀ ပြည့်နှစ်တွင် Hewlett-Packard, Intel, Lucent, Microsoft, NEC, Philips ကုမ္ပဏီတို့ စုပေါင်းကာ အချက်အလက် သယ်ပို့ နှင့် ပိုမိုမြန်သော USB 2.0 ကို ထုတ်လုပ်ခဲ့သည်။ သိဖြစ်၍ ကွန်ပူးတာ အပောင်းများတွင် USB port ပါသော်လည်း USB1.1 ဖြစ်နေတတ်သည်။ ငှါးတွင် USB2.0 ကို ဆက်သွယ်တပ်ဆင်လျှင် အဆင်မပြုနိုင်ပါ။ မိမိ၏ ကွန်ပူးတာတွင် မည်သည့် USB port အမျိုးအစားကို တပ်ဆင်ထားသည်ကို သိနိုင်ရန်

- \* My Computer ကို မောက်စ်ညာကာလစ် နှိပ်၍ ပေါ်လာသော pop-up menu မှ Manage ကို ကလစ်လုပ်လိုက်ပါ။
- \* Computer Management (box) ပေါ်လာပြီး ငါးမှ Device Manager (icon) ကို ကလစ်လုပ် ဖွင့်ပါ။
- \* ထိအခါ ညာဘက်ဖြင့်ကွင်းအတွင်း ကွန်ပူးတာတွင် တပ်ဆင်ထားသည့် ဟာ့အုပ်ပစ္စည်းအားလုံး၏ စာရင်းကို မြင်ရပါမည်။ ငါးမှ Universal Serial bus controllers စာရင်းကို ချုံကြည့်လိုက်ပါ။
- \* အဆိုပါ စာရင်းများတွင် Enhanced Host Controller နှင့် (သို့မဟုတ် USB2.0 Root Hub Device ဟူသော အမည် ပါနေပါက ကွန်ပူးတာတွင် USB2.0 port ရှိသည်ဟု မှတ်ယူနိုင်ပါသည်။
- \* အကယ်၍ Universal Host Controller, Open Host Controller နှင့် (သို့မဟုတ်) USB Root Hub (Device) ဟူသော စာသားဖြစ်နေပါက ကွန်ပူးတာတွင် USB1.1 သာ ရှိသည်ဟု မှတ်ယူနိုင်ပါသည်။
- \* ထိအတူ Controller သို့မဟုတ် Root Hub ၏ ရှုံးတွင် ထုတ်လုပ်သူ၏ အမည်ကို တွေ့နိုင်သည်။
- \* မှတ်ရန်မှာ စာရင်းတွင် Universal Serial Bus Controllers ဟူသော အမည် ပေါ်ရှိပါက ကွန်ပူးတာ၏ USB controller သို့မဟုတ် BIOS တွင် USB နှင့်ပတ်သက်သည့် ပြဿနာတစ်ခုခု ရှိနေပြီဟု မှတ်နိုင်သည်။ ငှါးအတွက် ကွန်ပူးတာ ဝယ်ယူခဲ့သည့်ဆိုင်သို့ ဆက်သွယ်မေးမြန်းကြည့်ပါ။



# JAVA DEVELOPER GUIDE

## Chapter (4)

### 4.6 DEVELOPING A FRAMEWORKS

ပြီးခဲ့တဲ့ သင်ခန်းစာတွေကို ပိုချပေးစဉ်မှာ ထူးခြားတာကို သွားတွေ့ရပါတယ်။ အဲဒါကတော့ အရင် programming အခြေခံ တွေကို သင်ကြားပိုချစဉ်ဟန်းက လေ့လာမှ မရှိခဲ့ဘဲနဲ့ Swing & AWT components တွေကို ပိုချပေးတဲ့အခါမှသာ ကြားဖြစ် လေ့လာတဲ့သွားရှိပါတယ်။ အမှန်ကတော့ ဘယ် Language မှာမဆို အခြေခံကစြိုးတော့ လေ့လာမှသာလျှင် အမှန်တကယ် တတ်မြောက်မှာ ဖြစ်ပါတယ်။ ဒါပေမယ့် တော်တော်များများက တော့ သူတို့အတွက် အဆင်ပြေမယ့် အခန်းကဏ္ဍတွေကနေ စတင် ပြီးတော့ လေ့လာနေကြပါတယ်။ အဲဒီလို ကြားဖြစ် လေ့လာသွားတွေကို သက်ဆိုင်ရာ Chapter Index တွေမှာ Code\_File တွေ ရေးနည်း၊ Save & Exit လုပ်နည်း၊ Command\_Line Interface ထဲကို ဝင်ရောက်ပြီးတော့ DIR ပြောင်းနည်း၊ ပြီးတော့ Compile & Run လုပ်နည်း၊ စတော်တွေကို သက်ဆိုင်ရာ အခန်း တိုင်းမှာ ထည့်သွင်း ရေးသားခဲ့ပါတယ်။

ဒီအခါမှာတော့ Java ရဲ့ Swing Package လို့ အော်လုံးများ ပြုလိုကုတ် အတွက် အခြေခံကြားလို့ ထင်ပြီးတော့ ချိန်လုပ်ထားခဲ့တဲ့ JFrame and JWindows အကြောင်းကို မရှင်းလင်းဘဲ ရောတွေးနေကြတာကို တွေ့လာရပါတယ်။ အမှန်ကတော့ ကျွန်တော်ဘက်က ရှင်းလင်းတဲ့အခါမှာလည်း JFrame\_Windows လို့ ခေါ်ပေါ် ရေးသားခဲ့တဲ့အတွက်ပါပဲ။ ဒါကြောင့် အရေး မကြိုးလောက်ဘူးလို့ ထင်ပြီးတော့ ချိန်လုပ်ထားခဲ့တဲ့ JFrame and JWindows အကြောင်းကို ကျွန်တော်အနေနဲ့ သတိထားပြီး တော့ ရှင်းလင်း ပိုချပေးရတော့တာပါပဲ။ အတိအကျပြောရရင် တော့ ကျွန်တော်အနေနဲ့ JFrame\_Windows လို့ ခေါ်ပေမယ့် တကယ်တစ်းကျတော့ JFrame နဲ့ JWindows ဟာ လုံးဝ မတူကပါဘူး။ ဒါကြောင့် အလွယ်အနေနဲ့ sample programs ငယ်လေး နှစ်ခုကို အသုံးပြုပြီးတော့မှ .JFrame နဲ့ JWindows နဲ့ကို ရှင်းလင်း ပိုချပေးပါမယ်။

က အရင်ဆုံး Listing 4.6 အနေနဲ့ ရေးသားထားတဲ့ program ကို အရင် လေ့လာကြရအောင်။

#### Listing 4.6 Demonstrate the SimpleFrame.

java

```
1. import javax.swing.JFrame;
2.
3. public class SimpleFrame extends JFrame
{
4.     public SimpleFrame(){
5.         super("Frame Title");
6.         setSize(300,200);
7.         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
8.         setVisible(true);
9.     }
10.    public static void main(String args[]){
11.        SimpleFrame sf = new SimpleFrame();
12.    }
13.
14. }
```

#### Analysis

၁။ ဒီ program ဟာ သာမန် JFrame တစ်ခုကို တည်ဆောက်တဲ့ program\_code line လေးတွေပါပဲ။ Complex ဖြစ်ပြီးတော့ မှားယွင်းစရာ ဘာမျှ မပါဝင်ပါဘူး။ ဒီ program ရဲ့ ထိပ်မှာ-

```
import javax.swing.JFrame;
```

ဆိုတဲ့ package file ကို import လုပ်ပြီးတော့ ခေါ်ယူပါတယ်။ အဲဒီနောက် SimpleFrame ဆိုတဲ့ primary class

ကို လိုင်းနံပါတ် (၃) မှာ ရေးသားပါတယ်။ JFrame class ကို extends လုပ်ထားတာဖော်။ သတိထားကြည့်ပါ။ တစ်နည်းပြောရရင်တော့ JFrame တစ်ခုကို ဖန်တီးရယူမှာဖြစ်လို့ အမြတ်ဆုံး ဖြစ်ပါဘူး။ အဲဒီနောက် Super () method ကို အသုံးချ ပြီးတော့ JFrame ရဲ့ title bar မှာ ဂါဝင်ရမယ့် စာသားကို ဖန်တီး ရေးသားပါတယ်။ ဒါဟာ JFrame class ထဲက constructor () method ကို အမြဲပြုပြီးတော့ ရေးသားထားတာပါပဲ။ ဟုတ်ပြီးနော်။

- JII ප්‍රි:තො? Setsize () method ගි ඇත්: බුල්පි:තො? JFrame තු වූ Width & Height අවුරුදු නො ගැනීමේ අංකය නො පි තයි. විභාගී මා ප්‍රි:තො? නො ගැනීමේ අංකය නො පි

- Set the size of the frame's windows, in pixels.

လို ဆိတ္တာပါပဲ။ Width & Height ကို ရေးသားထားတဲ့ အခါမှာ Pixels in Size နဲ့ တွက်ချက်ပြီးတော့ ရေးသား တာပါပဲ။ ဒါ programမှာတော့ JFrame ရဲ့ အဆင့်အစားကို 300 x 100 pixels in size နဲ့ တွက်ချက်ပြီးတော့ ဖန်တီး ထားပါတယ်။ ဒါကြောင့် Setsize (300, 100) လို ရေးသား ထားတာပါပဲ။

ထပ်မံဖြေးတော့ ရှင်းပြရမယ့် code\_line ကတော့ SetDefaultCloseOperation () method ပါပဲ။ က ဒါကို မရှင်းပြခဲ့မှာ JFrame နဲ့ ပတ်သက်ဖြိုးတော့ လွှေလာကြရအောင်။ JFrame မှာဆိုရင် maximize, minimize and close button တွေဟာ title bar မှာ user's control အနေနဲ့ ပါဝင်ပါတယ်။ Java Language မှာတော့ လက်ရှိ Run နေတဲ့ Application တစ်ခုကို Run နေရာကနေ ရပ်ပစ်ချင်တယ်ဆိုရင်တော့ ဖော်တီးထားတဲ့ JFrame ကို SetDefaultCloseOperation () method နဲ့ ပေါင်းစပ် ဖြေးတော့ ခေါ်ယူ အသုံးချဖူမှာပါပဲ။ အဲဒီ method ရဲ့ arguments နေရာမှာ အောက်ပါ static variables (4) ခဲ့တဲ့ တစ်ခုခုကို ခေါ်ယူ အသုံးချမှာပါပဲ။

- EEXIT\_ON\_CLOSE
    - Exit the program when the frame is closed.
  - DISPOSE\_ON\_CLOSE
    - Close the frame, dispose of the frame object, and keep running the application.

- DO NOTHING ON CLOSE - Keep the frame open and continue running.

- HIDE\_ON\_CLOSE
    - Close the frame and continue running.

- ၃။ **ပြီးတော့** setvisible (true) method ကို ခေါ်ယူ အသုံးချ ထားပါတယ်။ ဒါဟာ JFarme ကို display လုပ်ပေးနို့ အတွက်ပါပဲ။ **ပြီးတော့** main () functions ကို ရေးသား ထားပါတယ်။ Main () function တဲ့မှာ

**SimpleFrame** sf = new SimpleFrame();  
 ဒီပြီးမှ အပေါက ဖန်တီးခဲ့တဲ့ class ကို object အဖြစ်  
 ပြောင်းလဲ ခေါ်ယူ အသုံးချဖိုက်တာပါပဲ။ လိုင်းနံပါတ် (၁၁)  
 မှာပါ။

କି ଆଜି ଯାଇବାକୁବେଳାକି ହେଉଥିଲା ଏହାରେ Listing 4.6 ଓ Sample Program ବା JFrame ତାଠି ଫନ୍କ୍ଷନ୍‌କୁ ଦିଆଯାଇଛି । ଏହାରେ କୌଣସି କି ଏହାରେ କୌଣସି କି ଏହାରେ କୌଣସି କି ଏହାରେ କୌଣସି କି

```
SimpleFrame.java - Notepad
File Edit Format View Help
//Listing 4.6 Demonstrate the SimpleFrame.java
import javax.swing.JFrame;
public class SimpleFrame extends JFrame{
    public SimpleFrame(){
        super("Frame Title");
        setSize(300,200);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        setVisible(true);
    }
    public static void main(String args[]){
        SimpleFrame sf= new SimpleFrame();
    }
}
```

Figure 4.20 Write the program code\_line in the Notepad, SimpleFrame.java

က ပြီးရင်တော့ Command\_Line interface ထဲကို ဝင်ပြီး  
တော့ Compile & Run လုပ်ကြည့်ကြရအောင်။ Start → All  
programs → Accessories → Command prompt ထဲကို  
ဝင်ပြီးတော့ Compile & Run လုပ်ရမှာပါ။ က Command  
prompt ဖွင့်လာပြီဆိုရင် အောက်ပါ Command နဲ့ DIR ကို  
ပြောင်းပါ။



```
CD C:\j2sdk1.4.1_01\bin
DIR ဟာ C:\j2sdk 1.4.1_01\bin အဖြစ် ပြောင်းလဲသွားရင်
အောက်ပါ command နဲ့ clear screen လုပ်ပါ။
CLS
ဟုတ်ပြောနိုင်၏ အဲဒီအခါမှာ အောက်ပါ command နဲ့ program
ကို compile လုပ်ပါ။ Command ကို ရေးပြထားပါတယ်။
javac SimpleFrame.java
compile လုပ်လိုက်လို့ error မရှိတော့ဘူးဆိုရင် အောက်ပါ command နဲ့ run လုပ်လိုက်ပါ။
java SimpleFrame
အဲဒီလိုမျိုး Compile & Run လုပ်ပုံကို Figure 4.21 အနေနဲ့
ရေးပြထားပါတယ်။ လေ့လာကြည့်ပါ။
```

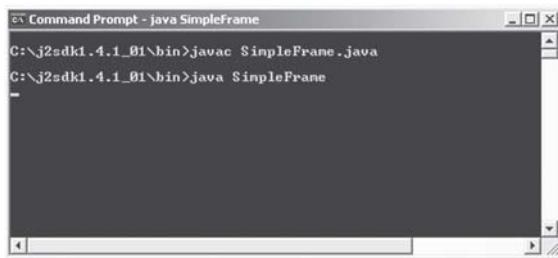


Figure 4.21 Compile & Run of the SimpleFrame.java

ကဲ ဟုတ်ပြော။ Compile & Run လုပ်လိုက်လို့ error မရှိတော့ဘူး ဆိုပါစို့။ အောက်ပါအတိုင်း SimpleFrame.java ဆိုတဲ့ program ဟာ လာပြီးတော့ ဖော်ပြပေးနေမှာပါပဲ။



Figure 4.22 Run the SimpleFrame.java

ကဲ ဒီတော့ JFrame ကို လေ့လာကြည့်ပြီးပြီးဆိုရင် JWindow နဲ့ ပတ်သက်တာကို ဆက်လက် လေ့လာကြည့်ရအောင်လား။ JWindow ဟာ JFrame နဲ့ အလုပ်လုပ်တာ အတူတူနီးပါးပါပဲ။ တူညီတယ်လို့ မပြောဘူးနော်။ ဆင်တူတာပါ။ အဲဒီ JWindow ဟာလည်း Swing Component ပါပဲ။ သူကို ဖန်တီး အသုံးပြု မယ်ဆိုရင်

```
import javax.swing.JFrame;
```

ဆိုတဲ့ package file ကို import & Declare လုပ်ရမှာပါပဲ။ ဒါပဲ မယ့် JFrame မှာလိုမျိုး titles, closing a window အစရိတဲ့ features တွေကတော့ JFrame မှာ မပါဝင်ပါဘူး။ ဒီတော့ Listing 4.23 ကို ရေးသားပြီးတော့ သင်ကြားပိုချေပေးပါမယ်။ Program ကို run လိုက်ရင် window တစ်ခုဟာ command\_line windows ထဲမှာ လာပြီးတော့ open လာလုပ်ပါမယ်။ အဲဒီ အခါမှာ 10,000 integers တွေဟာ command\_line window ထဲမှာ အရင်လာပြီး ပြပါလိမ့်မယ်။ ပြီးရင်တော့ ဖန်တီးထားတဲ့ JWindow application ဟာ ပိတ်သွားမှာပါ။ ဒီတော့ JWindow နဲ့ ပတ်သက်တဲ့ application တစ်ခုကို ဖန်တီးကြည့်ကြရ အောင်လား။ ဒီတော့ Listing 4.7 အနေနဲ့ ရေးသား ဖော်ပြထားတဲ့ sample program ကို လေ့လာကြည့်ကြရအောင်။

**Listing 4.7 Demonstrate the SimpleWindow.**

```
java
1. import javax.swing.JFrame;
2.
3. public class simpleWindow extends
4. public simpleWindow(){
5.     super();
6.     setBounds(250, 225, 300, 150);
7. }
8.
9. public static void main (String []
10. simpleWindow SW = new simpleWindow();
11. sw.setVisible (true);
12. for (int, i = 0, i < 1000, i + +)
13.     System.out.print (i + " ");
14.     sw.setVisible(false);
15.     System.exit(0);
16. }
17.
18. }
```

### Analysis

၁။ ဒါ program မှာ import javax.swing.JFrame; ဆိုပြီး Declare လုပ်ထားပါတယ်။ ပြီးတော့ SimpleWindow ဆိုပြီး primary class ကို ဖန်တီး ရေးသားပါတယ်။ ရေးသားရာမှာ swing package ထဲမှာပါဝင်တဲ့ JWindows class ကို extends လုပ်ထားတာပါ။ လိုင်းနံပါတ် (၄) မှာ တော့ SimpleWindow () constructor function ကို ရေးသားပါတယ်။ ပြီးတော့ super () method ကို ခေါ်ယူပြီးတော့ အသုံးချပါတယ်။ အခု အသုံးချတဲ့ super () method ဟာ JWindow class ထဲမှာ ပါဝင်တဲ့ super () method ပါပဲ။ ကဲ ပြီးတော့ SetBounds () method ကို ခေါ်ယူပြီးတော့ အသုံးချပါတယ်။ ဒါကြောင့်



```

SetBounds (250,      225,      300,      150);
           ↑          ↑          ↑          ↑
           x          y
position position width Height
လို ရေးထားတာပါပဲ။
အခု ကျွန်တော် ရေးပြခဲ့သလိုပါပဲ။ အရှေ့က
arguments 1 = x
arguments 2 = y
arguments 3 = Width
arguments 4 = Height ဆိုပြီးမှ သတ်မှတ်ထားတာ ဖြစ်ပါ
တယ်။ စာဖတ်သူ ဖန်တီးချင်တဲ့ JWindow အရွယ်အစားအတိုင်း
ဖန်တီးရယူလို့ ရပါတယ်။

```

**Note :** C & C++ programmer တွေကတော့ အောက်ပါ ပုံစံအတိုင်း Windows ရချင်တယ်ဆိုရင်ပေါ့နော်။ အောက်မှာ code\_line လေးတွေနဲ့ ဖန်တီးရယူလို့ရနိုင်တာကို သိပြီးသားဖြစ် မှာပါ။ လေ့လာကြည့်ပါ။

```

#include(<conio.h>
main ( )
{ window (20, 10, 60, 20); textbackgroundcolor(7); clrscr( );
getch ( );
}

```

အဲဒီ code\_line လေးကို အသုံးပြုပြီးတော့ run လိုက်မယဆိုရင် ပေါ့။ ပူးတွဲဖော်ထားတဲ့ figure က ပုံအတိုင်းပဲ command.line window ထဲမှာ windows ငယ်လေး လာပြီးတော့ ဖော်ပြုပေးနေ မှာပါပဲ။ ထားပါတော့။ ဒါကတော့ ဥပမာအနေနဲ့ နှိမ်းယူပြတာပါပဲ။ ဒါပေမယ့် C & C++ Programmer တွေအန္တကလည်း နှင့် ယဉ်လို့ရအောင် ရေးပြတာပဲ ဖြစ်ပါတယ်။ တကယ်လို့ အခု ရှင်းပြ နေတဲ့ Simple Window.java program ထကအတိုင်း For Looping ပတ်ပြီးတော့ ရေးရင်လည်း ရတာပါပဲ။ ဒါပေမယ့် ဥပမာပေးပြီးတော့ ရေးပြတာပဲ ဖြစ်ပါတယ်။ ဒီတော့ နောက် အားလုံး integer ကိန်းတွေကြားမှာ run လိုက်တာတွေ မပါဝင်ဘူးပေါ့နော်။

J) For Looping ကိုတော့ လိုင်းနံပါတ် (၁၂)နဲ့ (၁၃)မှာ ရေး သားထားပါတယ်။ Int iဟာ 0 ကနေစတယ်။ iဟာ 10,000 ထက်တော့ ငယ်မယ်။ i ကို post fix increment order နဲ့တိုးမယ်။ ဆိုပြီးတော့မှ Looping ကို ပတ်ထားတာပါပဲ။ ပြီးတော့ SetVisible() method ကို ခေါ်ယူ အသုံးချထား ပါတယ်။ Arguments ကိုတော့ False အနေနဲ့ ရေးထားတာ နော်။ ပြီးတော့ system.exit(0); method ကို ခေါ်ယူအသုံး ချပြီးတော့ ဖန်တီးထားတဲ့ JWindow ကို exit လုပ်ထားပါ တယ်။

က အခု ရေးသားရင်းပြခဲ့တဲ့ code\_line တွေကို Notepad ထဲမှာ သွားပြီးတော့ ရိုက်သွင်းပါ။ အဲဒီလို့ code\_line တွေ ရေး သားထားပုံကို အောက်က Figure 4.23 မှာ ရေးသား ဖော်ပြထား ပါတယ်။

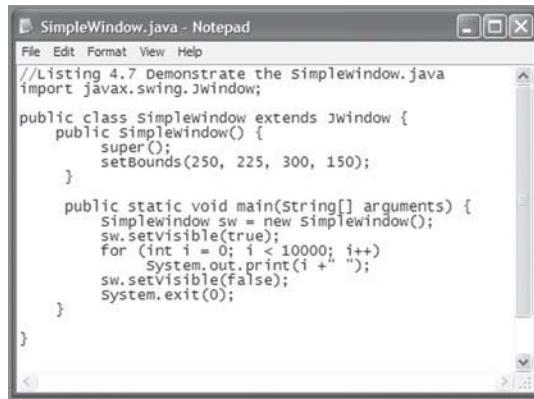


Figure 4.23 Write the program Code\_Line in Notepad, SimpleWindow.java

Notepad ထဲမှာ code\_line တွေ ရိုက်သွင်းပြီးသွားရင် C:\j2sdk1.4.1\_01\bin အောက်မှာ SimpleWindow.java လို့ အမည်ပေးပြီးတော့ save လုပ်ပါ။ အဲဒီတော့ file ကို save လုပ်ပြီးသွားရင် command\_line window ထဲကို ဝင်ပြီးတော့ အောက်ပါ အတိုင်း DIR ပြောင်းပါ။

CD C:\j2sdk1.4.1\_01\bin ↵

က DIR ပြောင်းသွားပြီးရင် အောက်ပါ command နဲ့ compile လုပ်ပါ။

javac simplewindows.java ↵

က error မရှိတော့ဘူးဆိုရင် အောက်ပါ command line နဲ့ run လုပ်ပါ။

java simplewindow ↵

အဲဒီဆိုရင်တော့ အောက်မှာမြင်ရတဲ့အတိုင်း simplewindow.java ဆိုတဲ့ program ဟာ လာပြီးတော့ run လုပ်နေမှာပါပဲ။

| 1 Command Prompt - java SimpleWindow                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 3200 3201 3202 3203 3204 3205 3206 3207 3208 3209 3210 3211 3212 3213 3214 3215 3216 3217 3218 3219 3220 3221 3222 3223 3224 3225 3226 3227 3228 3229 3230 3231 3232 3233 3234 3235 3236 3237 3238 3239 3240 3241 3242 3243 3244 3245 3246 3247 3248 3249 3250 3251 3252 3253 3254 3255 3256 3257 3258 3259 3260 3261 3262 3263 3264 3265 3266 3267 3268 3269 3270 3271 3272 3273 3274 3275 3276 3277 3278 3279 3280 3281 3282 3283 3284 3285 3286 3287 3288 3289 3290 3291 3292 3293 3294 3295 3296 3297 3298 3299 3300 3301 3302 3303 3304 3305 3306 3307 3308 3309 3310 3311 3312 3313 3314 3315 3316 3317 3318 3319 3320 3321 3322 3323 3324 3325 3326 3327 3328 3329 3330 3331 3332 3333 3334 3335 3336 3337 3338 3339 3340 3341 3342 3343 3344 3345 3346 3347 3348 3349 3350 3351 3352 3353 3354 3355 3356 3357 3358 3359 3360 3361 3362 3363 3364 3365 3366 3367 3368 3369 3370 3371 3372 3373 3374 3375 3376 3377 3378 3379 3380 3381 3382 3383 3384 3385 3386 3387 3388 3389 3390 3391 3392 3393 3394 3395 3396 3397 3398 3399 3400 3401 3402 3403 3404 3405 3406 3407 3408 3409 3410 3411 3412 3413 3414 3415 3416 3417 3418 3419 3420 3421 3422 3423 3424 3425 3426 3427 3428 3429 3430 3431 3432 3433 3434 3435 3436 3437 3438 3439 3440 3441 3442 3443 3444 3445 3446 3447 3448 3449 3450 3451 3452 3453 3454 3455 3456 3457 3458 |

Figure 4.24 Compile & Run of the SimpleWindow.java



က အခဲ့ဖြထားတဲ့ Figure 4.24က command prompt ပေါ်မှာ run ပြနေတာက JWindow လေးပါပဲ။ Swing components ထဲမှာ ပါဝင်တဲ့ features of component ပဲရော့။ ဒီတော့ C & C++ Programmer တွေအနိုကလည်း အခု သင် ခန်းစာဖြစ်တဲ့ JFrame နဲ့ JWindow အကြောင်းကို လေ့လာတဲ့ အခါမှာ အလွယ်တကူ compare လုပ်ပြီးတော့ လေ့လာလို့ ရနိုင် တာပေါ့နော်။ C & C++ အနေနဲ့ ရေးပြထားတဲ့ code\_line အနေနဲ့ကတော့ ပြည့်စုံလာတယ် မဟုတ်ပါဘူး။ ဒါပေမယ့် sample code\_line အနေနဲ့ ရေးပြထားတာပါပဲ။ အောက်မှာ ဖော်ပြထားတဲ့ Figure 4.25 ကို လေ့လာကြည့်လိုက်ပါလား။

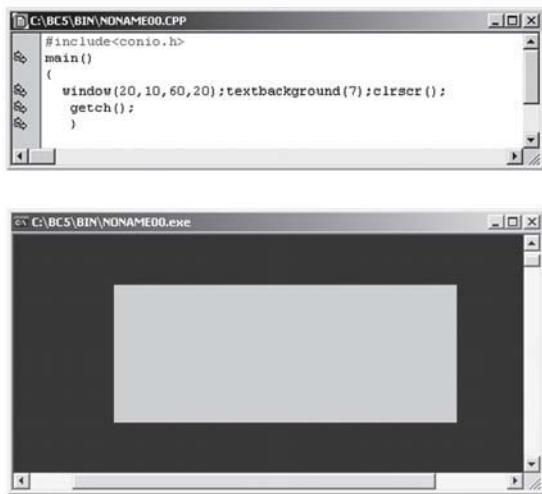


Figure 4.25 Write the program Code\_Line, Compile & Run of the C & C++ programming

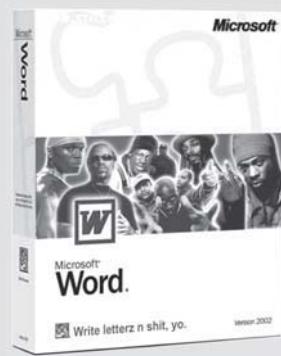
က အခု ရေးသားထားတဲ့ program ဟာ C & C++ Language မှာ JAVA Language ရဲ့ JWindow သဏ္ဌာန် Window တစ်ခု ဖန်တီးထားတာကို ရေးသား ရှင်းပြထားတာပဲ ဖြစ်ပါတယ်။ က ဒီတော့ ဆက်လက်ပြီးတော့ Chapter (5) Arranging Components on a User Interface ဆိုတဲ့ သင်ခန်းစာအပိုင်းကို ဆက်လက် လေ့လာကြည့်ကြပါမို့။ JAVA Developer Guide သင်ခန်းစာ VOL-I ရဲ့ နောက်ဆုံး Chapter ဖြစ်တဲ့အတွက် Certification Practice ဆိုတဲ့ အခန်းကဏ္ဍပါ ပါဝင်ပါတယ်။ သေသေချာချာ လေ့လာ ဖြေဆိုနိုင်ပြီးဆုံးရင်တော့ JAVA Developer Guide Volume - II ကို ဆက်လက် လေ့လာလို့ အခြေခံကောင်းတွေပဲ ဖြစ်ပါတယ်။ က လေ့လာကြပါမို့။

KYAW ZAYAR LAY [ IT ENGINEERING ]

### Microsoft Word သမိုင်းကြောင်း

၁၉၇၅ ခုနှစ်တွင် တစ်ဦးချင်း အသုံးပြနိုင်သည့် ကွန်ပူးတာ ထောက်တင်းအဖြစ် MITS မှ Altair ကို မိတ်ဆက် ထုတ်လုပ်ခဲ့သည်။ ၁၉၇၆ ခုနှစ်မှ စတင်ကာ Steve Jobs နှင့် Steve Wozniak တို့သည် Apple ကုမ္ပဏီကို တည်ထောင်ခြင်းဖြင့် ကွန်ပူးတာနယ်ပယ်သို့ ဝင်ရောက်လာခဲ့သည်။ ထိုအတူ Bill Gates နှင့် Paul Allen တို့သည်လည်း မိုက်ကရှိဆော်ဖွုမ္မာနယ်ပယ်ကို တည်ထောင်ခဲ့ကြကာ တစ်ဦးချင်းသုံး ကွန်ပူးတာနယ်ပယ်ကို အလျင်အမြန် တိုးတက်စေခဲ့သည်။ ၁၉၇၈ ခုနှစ်တွင် စာစီဆော်ပဲများ အဖြစ် WordMaster, Electric Pencil II, Apple Writer စသည်တို့ ပေါ်ထွက်လာခဲ့သည်။ ၁၉၇၉ ခုနှစ်တွင် WordStar စာစီဆော်ပဲကို ထုတ်လုပ်ခဲ့သည်။ ၄င်းသည် မြန်မာနိုင်ငံ ကွန်ပူးတာနယ်ပယ်တွင် ထင်ရှားခဲ့သည့် စာစီဆော်ပဲတစ်ခုလည်းဖြစ်သည်။ ထိုစဉ်က ပြင်ဘက်ဆော်ပဲတစ်ခုလည်း WordPerfect ဖြစ်သည်။

မိုက်ကရှိဆော်ဖွုမ္မာနယ်သည် ဝင်းဒီဇိုင်စနစ် ဆော်ဖြင့် ကွန်ပူးတာနယ်ပယ်တွင် ထင်ရှားလာခဲ့သည်။ Microsoft Word သည် စာစီဆော်ပဲတစ်ခုဖြစ်၍ မိုက်ကရှိဆော်ဖွုမ္မာနယ်မှ ထုတ်လုပ်သည့် စာစီလုပ်ငန်းသုံးဆော်ပဲ ဖြစ်သည်။ Word ကို ၁၉၈၃ ခုနှစ် ဖေဖော်ဝါရီလတွင် စတင် ရေးသားထုတ်လုပ်ခဲ့ပြီး ထိုစဉ်က Multi-Tool Word အမည်နှင့် ဖြစ်သည်။ ၁၉၈၃ ခုနှစ် အောက်တို့ ဘာလတွင် Microsoft Word အမည်သို့ ပြောင်း၍ ထုတ်လုပ်ခဲ့ပြီး Word 2, Word 3, Word 4, Word 5, Word 5.1, Word 5.5, Word 6.0 version များတို့ ၁၉၉၃ ခုနှစ်အထိ နှစ်စဉ် တိုးခဲ့၍ ထုတ်ဝေခဲ့သည်။ ၁၉၉၅ ခုနှစ်တွင် Word 7 ကို Word for Windows 95 ဟု အမည်သံဖြင့် ထုတ်ဝေခဲ့ပါသည်။ ထိုအတူ ၁၉၉၇ ခုနှစ် Word 8 (Word 97)၊ ၁၉၉၉ ခုနှစ် Word 9 (Word 2000)၊ ၂၀၀၁ ခုနှစ် Word 10 (Word 2002)၊ ၂၀၀၃ ခုနှစ် Word 11 (Word 2003) နှင့် ၂၀၀၆ ခုနှစ် Word 12 (Word 2007) ဟူ၍ ထုတ်ဝေခဲ့ပါသည်။ ၄င်း Word ကို Microsoft Office suite အဖော်ဖြစ်၍ Excel, PowerPoint, Access စသည် ဆော်ပဲများနှင့်အတူ စုပေါင်း ထုတ်လုပ်ထားခြင်းဖြစ်သည်။





# JAVA DEVELOPER GUIDE

## Chapter (5)

### 5.1 ADDING THE SWING COMPONENTS OF JAVA

#### Slider (OR) JSlider

သင်ခန်းစာထဲမှာ သင်ကြားပို့ချမှတ်ရင်းနဲ့ ကုန်ခဲ့တဲ့ Swing Components တွေကို ယခုလက်ရှိ၍ Chapter [5] မှာ ထပ်မံဖြည့်စွက် သင်ကြားပေးမှုဖြစ်ပါတယ်။ က အခုသင်ခန်းစာအနေနဲ့ ကတော့ slider နဲ့ ပတ်သက်ပြီးတော့ သင်ယူကြရအောင်လား။ Slider ဟာ Swing Components ထဲမှာ ပါဝင်တဲ့ တစ်စိတ်တစ်စိုင်းပဲ ဖြစ်ပါတယ်။ သူဟာ JSlider Class ကို ခေါ်ယူ အသုံးပြုပြီးတော့ ဖန်တီး အသုံးချလို့ရပါတယ်။ Slider ရဲ့ သဘောတရားကတော့ သုံးစွဲသူ user ဟာ slider မှာ ပါဝင်တဲ့ Minimum and Maximum Range တွေကြားထဲမှာ set လုပ်ထားတဲ့ number ထွက်ရှိ sliding control ကို အသုံးပြုပြီးတော့ ထိန်းချုပ်လို့ ရပါတယ်။ Slider ကို အသုံးများတဲ့ type ကတော့ text field ထဲမှာ input လုပ်လိုက်တဲ့ Data\_Range ကို အမို့ပြုပြီးတော့ အလုပ်လုပ်ကြတာ များပါတယ်။ ဒါပေမယ့် သူရဲ့ လက်ခံနိုင်တဲ့ Range ဟာ Minimum\_Range and Maximum\_Range ဆိုပြီးမှ ရှိပါတယ်။ က အောက်က Figure 5.1 မှာတော့ JSlider နဲ့ ပတ်သက်တဲ့ Figure ကို ဥပမာအနေနဲ့ ဖော်ပြုပေးထားပါတယ်။

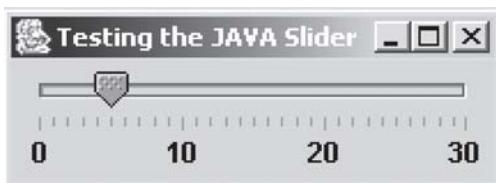


Figure 5.1 A JSlider Component

Slider ကို အသုံးပြုရာမှာ အများအားဖြင့် horizontal အနေနဲ့ပဲ အသုံးများပါတယ်။ အေဒီကြောင့်လည်း Slider are

used, horizontal default. လို့ ပြောကြတာပေါ့နော်။ ဒါပေမယ့်လည်း လိုအပ်လာရင် Orientation\_type မှာ စိတ်ကြိုက် ပြောင်းလဲပြီးတော့ ဖန်တီး ရယူလို့ ရပါတယ်။ ဘာကြောင့်လဲဆိုတော့ Swing-Constants interface အနေနဲ့ HORIZONTAL or VERTICAL ဆိုတဲ့ Constants\_Type တွေ ပါဝင်ပြီးသားပါပဲ။ က ဒီတော့ JSlider Class ကို အသုံးပြုမယ်ဆိုရင် ပါဝင်တဲ့ constructor method တွေကို အရင်ဆုံး လေ့လာကြရအောင်လား။ မူရင်းအတိုင်း ရေးသားပေးထားပါတယ်။ Exam တွေ ဖြစ်ရာမှာ အထောက်အကြုံ ရနိုင်အောင်လိုပါပဲ။ JSlider () constructor method (3) မျိုးကို ရှင်းပြေားထားပါတယ်။

JSlider ( int,int ) - A Slider with the specified minimum value and maximum value

JSlider (int, int, int) - A Slider with the specified minimum values, maximum, and starting value.

JSlider (int, int, int, int) - A Slider with the specified orientation, minimum value, maximum , and starting value.

(ယခု သင်ကြားမည့် program နဲ့ အထက်ပါ method ကို အသုံးပြုထားသည်။)

က အခု ရေးပြုခဲ့တဲ့ constructor method တွေကို ရေးသားဖော်ပြုပေးထားပါတယ်။ စာဖတ်သူအနေနဲ့ minimum value နဲ့ maximum value ပါဝင်ရမယ်။ အဲဒီ value နှစ်ခုကြားမှာ different value ကိုလည်း ထည့်သွင်းရပါမယ်။ တစ်နည်းပြောရရင်တော့ Step\_Value ပေါ့နော်။ ဟုတ်ပြုလား။

အထက်ဖော်ပြပါ method တွေအတွက် မှုလ default အတိုင်းကတော့

minimum value = 0,  
maximum value = 100,  
starting value = 50,

Horizontal orientation အနေနဲ့ default ပေးထားပါတယ်။ အဲဒါကိုမှ စာဖတ်သူအနေနဲ့ စိတ်ကြောက်ပြန်ပြီးတော့မှ ဖန်တီးလို့ ရနိုင်ပါတယ်။ ဒါဆိုရင်တော့ စာဖတ်သူအနေနဲ့ JSlider() constructor method တွေထက် ကိုက်ညီမယ့်၊ အသုံးဝင်မယ့် constructor method ကို ခေါ်ယူ အသုံးချုနိုင်ပါတယ်။ ဒီအခါမှာ JSlider class ကနေပြီးတော့ constructor overloading method အနေနဲ့ နားလည် လက်ခံပြီးတော့ အသုံးပြုခွင့်ပေးထားပါလိမ့်မယ်။ ကဲ ဒီတော့ ဆက်လက်ပြီးတော့ JSlider Class ထဲမှာ ပါဝင်တဲ့ အခြား method တွေကို လေ့လာကြရအောင်လား။ ဘာကြောင့်လဲဆိုတော့ JSlider Components နဲ့အတူ တွဲဖက်ပြီးတော့ အသုံးချုရအောင်လိုပါ။ ကဲ လေ့လာကြည့်ရအောင်လား။ JSlider Class ထဲမှာ ပါဝင်တဲ့ Serval methods တွေကို အောက်မှာ ရေးသားဖော်ပြပေးထားပါတယ်။

- SetMajorTickSpacing (int) - This method separates major tick marks by the specified distance. The distance is not in pixels, but in values between the minimum and maximum values represented by the slider.
- SetMinorTickSpacing (int) - This method separates minor tick marks by the specified distance. Minor ticks are displayed as half the height of major ticks.
- SetPaintTicks (boolean) - This method determines whether the tick marks should be displayed (a true argument) or not (a false argument).
- SetPaintLabels (boolean) - This method determines whether the numeric label of the slider should be displayed (true) or not

(flase).

ကဲ အခုရေးပြထားတဲ့ method တွေကို JSlider constructor() method နဲ့ ယုံ့တွဲပြီးတော့ အသုံးပြုနိုင်ပါတယ်။ ဒီတော့ ကျွန်တော်တို့အနေနဲ့ JSlider နဲ့ပတ်သက်ပြီးတော့ သင်ကြား ပို့ချေပေးထားတဲ့ Listing 5.1 ကို လေ့လာကြရအောင်လား။ Code\_Line ဧရ (၂၅) ကြောင်းပါဝင်တဲ့ program ထံလေးတစ်စုံပါပဲ။ လေ့လာကြည့်ပါ။ အဲဒီ program code\_line တွေကို Analysis လုပ်တဲ့အခါကျေမှု အပေါ်က method တွေကို မြန်မာလို လေ့လာကြတာပေါ်နော်။ အခုတော့ အကြိုးဖျင်းပဲ ရှင်းပြပေးခဲ့ပါမယ်။

- SetMajorTickjSpacing (int) method ကတော့ JScrollbar ပေါ်မှာ Ticks အမှတ်တွေအနေနဲ့ Major မှတ်အနေနဲ့ ထည့်သွင်းတာပဲ ဖြစ်ပါတယ်။ ဒီနေရာမှာ Major Ticks က 10 လို့ ထားခဲ့ရင် 0, 10, 20, 30 အစဉ်သြားဖြင့် ဖော်ပေးထားပါမှာ။ ဒါဟာ ဂိန်းကဏ္ဍးအနေနဲ့ ဖော်ပေးနိုင်သလို အစိတ်အကျွဲမှတ်တွေ အတွက်ပါပဲ။ ဒီ method ကတော့ နောက်ကွယ်က သတ်မှတ်ချက်ပါပဲ။ မဖော်ပြပေးနိုင်ပါဘူး။
- SetMinorTickSpacing (int) method ကတော့ JScrollbar ပေါ်မှာ သတ်မှတ်ထားတဲ့ Major မှတ်တွေရဲ့ ကြားမှာ Minor မျဉ်းစိတ်ငယ်လေးတွေကို သတ်မှတ်တဲ့ အတိုင်းအတာအတိုင်း ဖော်ပေးနိုင်ဖို့ သတ်မှတ်ပေးထားတာပါပဲ။ ဥပမာ Major Tick က 0, 10, 20 ဆိုပြီးတော့ ပေါ်အောင် သတ်မှတ်ပေးထားခဲ့ရင် အဒီကြားက Minor မျဉ်းစိတ်ကို SetMinorTickSpacing (5) လို့ သတ်မှတ်ထားရင် Major Tick ကြားမှာ မျဉ်းစိတ်ဟာ (5) စိတ်ပဲ ခြားပေးထားမှာပါ။ တကယ်လို့ parameter arguments အနေနဲ့ (10) လို့ ရေးထားခဲ့ရင် မျဉ်းစိတ်ဟာ (10) စိတ်ပဲ ခြားပေးထားမှာပါ။ ဒီ method ဟာလည်း သတ်မှတ်ချက်ပဲနော်။ မဖော်ပြပေးနိုင်ပါဘူး။
- SetPaintTicks (boolean) method ဟာ JSlider Bar ပေါ်က အစိတ်အကျွဲ မျဉ်းမှတ်ကို ဖော်ပြရန်အတွက် ထည့်သွင်း အသုံးချုတဲ့ method ပါပဲ။ ဒီ method ကတော့ JSlider ပေါ်မှာ မျဉ်းစိတ်ဖြင့်ရအောင် လုပ်ပေးတဲ့ method ပါပဲ။
- SetPointLabels (boolean) method ဟာ JSlider Bar ပေါ်က မျဉ်းဂိန်းကဏ္ဍး သတ်မှတ်ချက်ကို ထည့်သွင်းကြည့်ရနိုင်ရန်အတွက်ပဲ ဖြစ်ပါတယ်။ ဒီ method ကို ဓမထည့်သွင်းခဲ့ရင် ဂိန်းကဏ္ဍး သတ်မှတ်ချက်တွေ မပါဝင်နိုင်တော့ပါဘူး။ ဒီ method ကတော့ ဖော်ပြပေးနိုင်တဲ့ JSlider ပေါ်မှာ မြင်ရစေတဲ့ method ပါပဲ။ က အခု ရေးသား ရှင်းပြခဲ့တဲ့ method တွေကို အခြေခံပြီး



တော့ Listing 5.1 Testing the JAVA Slider ဆိုတဲ့ program ကို လေ့လာကြည့်ပါ။

#### Listing 5.1 Testing the JAVA Slider

```

1. import java.awt.event.*;
2. import javax.swing.*;
3.
4. public class slider extends JFrame{
5.
6. public slider(){
7. super ("Testing the JAVA Slier");
8. setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
9. JSlider pickNum = new JSlider(JSeparator.HORIZONTAL, 0, 30, 5);
10. pickNum.setMajorTickSpacing(10);
11. pickNum.setMinorTickSpacing(1);
12. pickNum.setPaintTicks(true);
13. pickNum.setPaintLabels(true);
14. JPanel pane = new JPanel();
15. pane.add(pickNum)
16.
17. setContentPane(pane);
18. }
19.
20. public static void main(String[] args){
21. Slider S = new Slider();
22. s.pack();
23. s.setVisible (true);
24. }
```

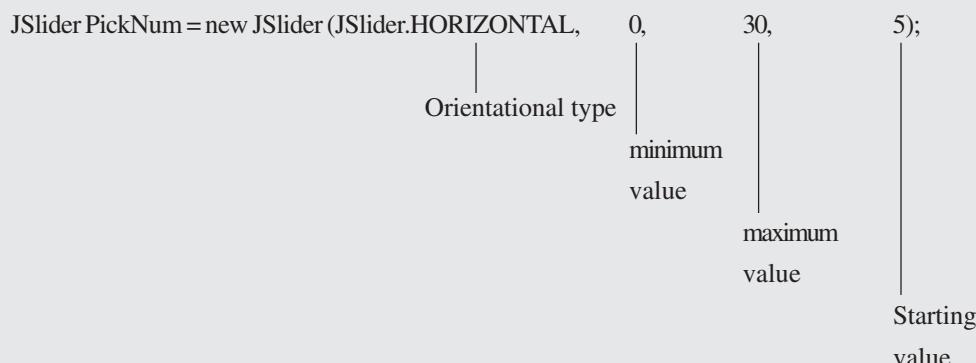
25. }

#### Analysis

၁။ လိုင်းနံပါတ် (၁) နဲ့ (၂) မှာ java.awt.event.\*;javax.swing.\*; ဆိုတဲ့ package file နှစ်ခုကို ခေါ်ယူထားပါတယ်။ အဲဒီနောက် လိုင်းနံပါတ် (၃) မှာမဲ့ slider ဆိုတဲ့ primary class ကို ဖန်တီးပါတယ်။ အဲဒီ slider class ရဲ့ အပေါ်မှာ constructor function ကို လိုင်းနံပါတ် (၆) မှာ ရေးသားထားပါတယ်။ အဲဒီ constructor function ထဲမှာမဲ့ JFrame Windows ရဲ့ Title\_Bar Text အနေနဲ့ ပါဝင်ရမယ့် စာသားကို အောက်ပါအတိုင်း ရေးသားထားပါတယ်။

`Super ("Testing the JAVA Slider");`

အဲဒီနောက်မှာတော့ setDefaultCloseOperation () method ကို ခေါ်ယူ အသုံးချထားပါတယ်။ ဒါကတော့ JFrame\_Windows ကို close လုပ်နိုင်ဖို့ပါပဲ။ ကဲ အဲဒီနောက် လိုင်းနံပါတ် (၅)မှာတော့ JSlider Class ရဲ့ Object\_Name PickNum လို့ အမည်ပေးပြီးတော့ ခေါ်ယူဖန်တီးပါတယ်။ ဒါ နေရာမှာတော့ JSlider Class ရဲ့ constructor method ထဲမှာ parameter\_arguments အနေနဲ့ ထည့်သွင်း ရေးသားထားပုံကို လေ့လာကြည့်ကြရအောင်။



ကဲ အဲဒီလို ရေးသားထားတာဟာ arguments\_type တွေနဲ့ ယျဉ်ပြီးတော့ လေ့လာကြရအောင်လား။

ကဲ အခု ရေးပြထားတဲ့ JSlider Class က constructor function ကို လေ့လာလိုက်ရင်တော့ ရေးထားတဲ့ code\_line စာကြောင်းကို သဘောပေါက်သွားမှာပါပဲ။ JSlider.HORIZONTAL ဆိုတာကတော့ Slider ကို JFrame\_Windows ပေါ်မှာ အလျော့လိုက်ပဲ ထားမယ်လို့ ခိုလိုတာပါပဲ။ Default type အတိုင်းပဲပေါ့နော်။ ပြီးတော့ ဒုတိယ arguments မှာ mini-

imum\_value ကို (0) အနေနဲ့ ထားထားပါတယ်။ တတိယ arguments အနေနဲ့ maximum\_value ကို (30) အဖြစ် ထားထားပါတယ်။ နောက်ဆုံး argument မှာ (5) လို့ ရေးသားထားတာ ကတော့ starting value ကို ထားထားတာပါပဲ။

၃။ အဲဒီနောက်မှာတော့ လိုင်းနံပါတ် (၁၀) မှာ PickNum.setMajorTickSpacing (10); လို့ ရေးထားတာကို တွေ့မြင်ရမှာပါပဲ။ ဒါဟာ JSlider Component ပေါ်မှာ သတ်မှတ်ထားတဲ့ အကွာအဝေး distance တစ်ခုကတော့ ရှိမှုပါပဲ။



အဲဒီအကွာအဝေးမှာ value တစ်ခုကို MajorTickSpacing အနေဖြင့် သတ်မှတ်ထားတာပါပဲ။ သတ်မှတ်ချင်တဲ့ Tick-Spacing ကို method ရဲ့ parameter.arguments နေရမှာ ထည့်ရေးသားလိုက်ရှုပါပဲ။ ဒါ code လိုင်းမှာတော့ 10 လို့ သတ်မှတ်ပေးထားပါတယ်။ အဲဒီကြောင့် JSlider Components ပေါ်မှာ အောက်ပါအတိုင်း 10 စီ space မြားပြီး တော့ ပေါ်နေတာပါပဲ။ လေ့လာကြည့်ပါ။ ဒါဟာ Major မျဉ်းစီစြားဖိုပါပဲ။

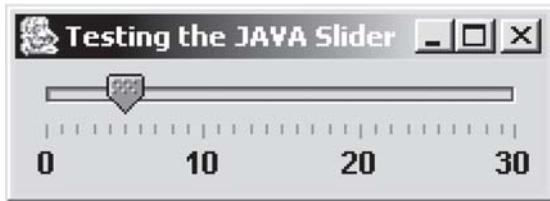


Figure 5.2 Major Spacing Of The JSlider

က ဒါဆိုရင် အဲဒီ Major Spacing 10 စီ မြားပြုတဲ့ကြေားမှာ အစိတ်ငယ်လေးတွေက 10 စီတဲ့ ပါဝင်တယ်နော်။ အဲဒီက တော့ လိုင်း နံပါတ် (၁၁) မှာ ရေးသားထားတဲ့ PickNum. SetMinorTickSpacing (1); လို့ ရေးသားထားလိုပါပဲ။ အပေါက်ပုံမှာ လေ့လာကြည့်ရင် တွေ့ဖြင့်နိုင်ပါတယ်။ အဲဒီ arguments ကြားမှာ 2 လို့ ပြင်ရေးလိုက်ရင် 0 နဲ့ 10 ကြားမှာ Minor Tick ဟာ 5 စီတဲ့ပဲ ရှိမှာပါ။ ဒါဆိုရင်တော့ သဘော ပေါက်လောက်ပါဖြစ်နော်။ က ဒီတော့ လိုင်းနံပါတ် (၁၂) မှာ ရေးသားထားတဲ့ PickNum. SetPaintTick (true); လို့ ရေးသားထားတာကို တွေ့ဖြင့်ရှိမှာပါ။ ဒါဟာ အရှေ့မှာ ရှင်းပြု ခဲ့သလိုပါပဲ။ JScrollPane ပေါ်မှာ သတ်မှတ်ထားတဲ့အပေါ်မှာ ပါဝင်စေတဲ့ Method ပါပဲ။ ဒါဟာ မျဉ်းစီစိအကျေအနေနဲ့ ပါဝင်လာအောင် ရေးသားထားတာပါပဲ။ လိုင်းနံပါတ် (၁၃) မှာ ရေးသားထားတဲ့ PickNum. SetPaintLabels (true); က တော့ JScrollPane ပေါ်မှာ ကိန်းကဏ္ဍားတွေ ပါဝင်လာအောင် ရေးသားထားတာပါပဲ။ တစ်နည်းပြုရှုရင် ကိန်းကဏ္ဍား စာသား Labels တွေကို ပါဝင်လာအောင် ရေးသားထားတာ ပါပဲ။ အဲဒီနောက် လိုင်းနံပါတ် (၁၄) မှာတော့ JPanel Class ကို Object\_Name pane လို့ အမည်ပေးပြီးတော့ ခေါ်ယူလိုက်ပါတယ်။ ပြီးတော့ အပေါက် ဖန်တီးရယူခဲ့တဲ့ PickNum လို့ ခေါ်တဲ့ JSlider Class ရဲ့ set\_method တွေကို JPanel Class ထဲကို ထည့်သွင်း အသုံးချပါတယ်။ ဒါဟာ Slider Class ရဲ့ constructor method ထဲမှာ ထည့်သွင်း ရေးသား အသုံးပြုထားတဲ့ code\_line တွေပါပဲ။ သေသေချာချာ နားလည် သဘောပေါက်အောင် လေ့လာကြည့်ပါ။

၄။ ဒါ program ရဲ့ main () function ကိုတော့ လိုင်းနံပါတ် (၂၀) မှာ ရေးသားထားတာကို တွေ့ဖြင့်ရမှာပါ။ အဲဒီနောက် လိုင်းနံပါတ် (၂၁) မှာ အပေါက် ဖန်တီးခဲ့တဲ့ Slider Class ကို Object\_Name 5 လို့ ပေးပြီးတော့ ခေါ်ယူ အသုံးချပါတယ်။ သတ်မှတ်ထားတဲ့ ရေးသားခဲ့တဲ့ method တွေ အတွက် pack () method ကို ရေးသား အသုံးချပါတယ်။ အဲဒီနောက် JFrame\_Windows ကို တွေ့ဖြင့်နိုင်အောင် SetVisible () method ကို ခေါ်ယူ အသုံးချထားပါတယ်။ အဲဒီနောက် main () function ရဲ့ close brace } နဲ့ primary class ရဲ့ close brace } ကို အသီးသီး လာပိတ်ထားပါတယ်။

```

public class slider extends JFrame implements Runnable {
    public slider() {
        super("Testing the JAVA Slider");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JSlider picknum = new JSlider(JSlider.HORIZONTAL, 0, 30, 5);
        picknum.setMajorTickSpacing(10);
        picknum.setMinorTickSpacing(1);
        picknum.setPaintTicks(true);
        picknum.setPaintLabels(true);
        JPanel pane = new JPanel();
        pane.add(picknum);
        setContentPane(pane);
    }

    public static void main(String [] args) {
        slider s = new slider();
        s.pack();
        s.setVisible(true);
    }
}

```

Figure 5.3 Write the program code\_line in the Notepad, slider.java

အခုလက်ရှိ ရေးသားခဲ့တဲ့ program မှာ JSlider Class ရဲ့ method () အသစ်တွေ ပါဝင်လာတဲ့အတွက် စာဖတ်သူအနေနဲ့ သတ်ထားပြီးတော့ လေ့လာကြည့်ပါ။ က ဒီတော့ စာဖတ်သူအနေနဲ့ Note Pad ထဲမှာ သွားပြီးတော့ code\_line တွေ ရေးသားကြရ အောင်လား။ Code\_Line တွေ ရေးသားထားပဲကို Figure 5.3 အနေဖြင့် လေ့လာကြည့်ပါ။

က အခု ဖော်ပြန့်ခဲ့တဲ့ Figure 5.3 ကအတိုင်း code\_line တွေ ရေးသားပြီးရင် အဲဒီ file ကို C:\j2sdk 1.4.1\_01\bin အောက်မှာ သွားပြီးတော့ save လုပ်ပါ။ Save လုပ်ရမယ့် File\_Name ကတော့ Slider.java ပါပဲ။ က ဒီတော့ အဲဒီ file ကို compile & run လုပ်ကြရအောင်။ Command\_Line interface (DOS) ကို ဖွင့်ပါ။ ပွင့်လာတဲ့ DOS windows ထဲမှာ အောက်ပါ command နဲ့ compile လုပ်ပါ။ အ ဒါပေမယ့် Command Prompt ထဲမှာ ပေါ်နေရမယ့် prompt သက်တကတော့ C:j2sdk1.4.1\_01\bin>လို့ ပေါ်နေရမှာနော်။ က ဟုတ်ပြေလေ။ အဲဒီအချိန်မှာ compile လုပ်ပါ။

```
javac slider.java ↵
```

ကဲ compile လုပ်လိုက်လို့ error မရှိတော့ဘူးဆိုရင်အောက်ပါ command နဲ့ run လုပ်လိုက်ပါ။ အောက်မှာ ရေးပြေးထားပါတယ်။

```
java slider ↵
```

အချင်းမျိုး compile & run လုပ်လိုက် Figure 5.4 အနေဖြင့် ရေးသား ဖော်ပြေးထားပါတယ်။ လေ့လာကြည့်ပါ။



Figure 5.4 Compile & Run of the slider.java in the command prompt

ကဲ အချေရေးပြခဲ့သလိုမျိုး compile & run လုပ်ပြီးတော့ program ကို တည်ဆောက်လိုက်ရင် Figure 5.5 မှာ ဖော်ပြထားတဲ့အတိုင်းပဲ JSlider ကို လာရောက်ပြီးတော့ ဖော်ပြပေးနေမှာ ပါပဲ။

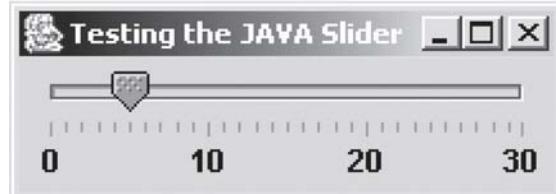


Figure 5.5 Run the program, slider.java

ကဲ JSlider နဲ့ ပတ်သက်တဲ့ သင်ခန်းစာအပြီးမှာ ဆက်လက်ပြီးတော့ 5.2 Using the JProgressBar (OR) ProgressBar ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် ပို့ချေပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

KYAW ZAYAR LAY [ IT ENGINEERING ]

နှင်းဆီကုန်းဘိုးဘွားရိပ်သာသမိုင်းနှင့် ပတ်သက်သော စာရွက်စာတမ်း  
တတ်ပုံများပေးပို့ပျော်ကြန် မေတ္တာက်ခံခြင်း

ဘွားအောင်းစီးသာ

(က) နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာ (၇၅) နှစ်ခေါ် စိန်ရတုအတိမီးအမှတ်စာတမ်း ဖြစ်ပြောက်ရေးကို ပိုလုပ်မှုကြေး စောမြင် (၆၅%)၊ ဦးလှမြိုင် (ကိုယောင်း)၊ ဦးရှုံးသန် (မောင်ဝရှင်)၊ ဦးချိစံဝင်း စသည် စာရေးဆရာတွေ့များက ကြိုးကြပ် စောင်ရွက်လျက် ရှိပါသည်။

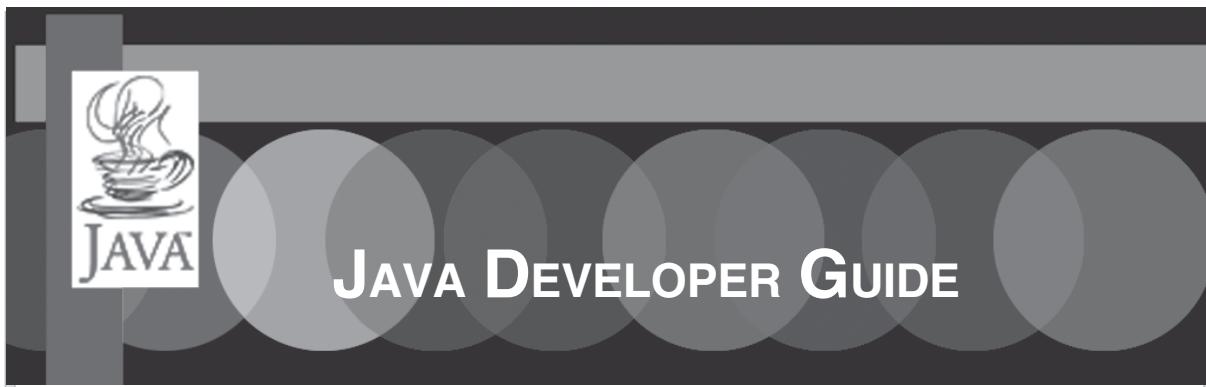
(ခ) စတင်တည်ထောင်ခဲ့သူ "ဘွားအောင်းစီးနှင့် နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာသမိုင်း" ကို စာရေးဆရာတွေ့း တွေ့ဖို့ သိလိုက်လိုပါသည်။

(ဂ) နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာကို ခုတ္တယ ဆက်လက်တည်ထောင်ခဲ့သူ "မြန်မာအလင်း ဦးတင်နှင့် နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာသမိုင်း" ကို စာရေးဆရာတွေ့း ဒေါက်တာ မတိုင်ငြင်က ဂုဏ်ပြုရေးသားမည် ဖြစ်ပါသည်။

(ဃ) နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာ၏ (၇၅) နှစ် သမိုင်းကို ဇွန်ကြားရေးမှုးချုပ် (အွေးသန်း) က ဂုဏ်ပြုရေးသားမည် ဖြစ်ပါသည်။

(ဃ) နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာ၏ (၇၅) နှစ် သမိုင်းကို ဇွန်ကြားရေးမှုးချုပ် (အွေးသန်း) က ဂုဏ်ပြုရေးသားမည် ဖြစ်ပါသည်။

အထောကဖော်ပြပါ သမိုင်းတွင်မည့် စာအုပ်များရေးသားရာတွင် အထောကအကျပ်မည့် အချက်အလက်များ၊ စာရွက်စာတမ်း စာတ်ပုံများကို အမှုဆောင်အဖွဲ့က ဖော်လုပ်ခြင်း အတွက် အတွင်းရေးမှုး စိုင်မှုးအောင်သန်း (၆၅%) (ဥရောပေါ် ၅၄၆၁၀၈၈) တံသိ လူမှုပါန်း၌ဖြစ်စေ စားရမ်း၌ဖြစ်စေ ကူညီကြပါရန် နှင်းဆီကုန်း ဘိုးဘွားရိပ်သာ အမှုဆောင်အဖွဲ့က ဖော်လုပ်ခြင်း ပါသည်။



## Chapter (5)

### 5.2 USING THE JPROGRESSBAR (OR) PROGRESSBAR

စာဖတ်သူဟာ ဂါရင့် programmer တစ်ယောက် မဟုတ်ခဲ့ရင်တောင် သာမန် user တစ်ဦးအနေနဲ့ ProgressBar တွေကို တွေ့မြင်ဖူးဖူးပါ။ အနီးစပ်ဆုံးသုပမာအနေနဲ့ ပေးရန်တော့ စာဖတ် သူရဲ့ computer ထဲကို software တွေ install လုပ်တဲ့ အချိန်ပဲ ပေါ့နော်။ အဲဒီအချိန်မှာ တွေ့မြင်ရတဲ့ install time မှာ 0% to 100% အထိ တစ်ဆင့်ချင်း တိုးပြီးမှ ပြတေတွေဟာ progress bar ကို အသုံးပြုပြီးတော့ ဖုန်တီးထားတာပါပဲ။ ဒါ ပြုပါတယ်။



Figure 5.6 A Program bar in a frame

Progress Bar တွေကို အသုံးချဖို့အတွက်ဆုံးရင်တော့ Swing Package ထဲမှာ ဂါဝင်တဲ့ JProgressBar Class ကို implemented လုပ်ပြီးတော့ အသုံးချရပါလိမ့်မယ်။ တကယ်လို့ များ JProgressBar Class ကို အသုံးချထားတဲ့ Program ငယ်တစ်ခုကို ဖော်ပြပေးရမယ်ဆုံးရင်တော့ အောက်မှာ ဖော်ပြထားတဲ့ Figure 5.6 ကိုသာ လေ့လာကြည့်လိုက်ပါ။

Progress Bar တွေကို အသုံးပြုတဲ့အခါမှာ အများအားဖြင့် တော့ minimum and maximum vlaue တွေနဲ့ represent လုပ်ထားတတ်ကြပါတယ်။ အမိုာယ်ကတော့ အရှုံးက သင်ကြား ပို့ချခဲ့တဲ့ JSlider လိုပါပဲ။ Beginning and ending အတွက် ပါပဲ။ JSlider နဲ့ ကွာခြားတာကတော့ progress bar ဟာ သတ်

မှတ်ထားတဲ့ time limiting အတွင်းမှာ အလိုအလျောက် အလုပ် လုပ်တာပါပဲဖြစ်ပါတယ်။

ဥပမာအနေနဲ့ပေးရရင်တော့ software တစ်ခုကို installation လုပ်တာနဲ့ပဲ ရှင်းပြရမှာပါပဲ။ ဒါ ပြုပါတယ်လုပ်ရမယ်ဆိုပါစို့။ အဲဒီအခါမှာ progress bar အနေနဲ့ task လုပ်ရမယ့် maximum ကတော့ 0 ဖြစ်ပြီးတော့ maximum value ကတော့ 335 ပဲပေါ့နော်။ ဒါပေမယ့် ကျွန်ုတ်တို့အနေနဲ့ install component time အနေနဲ့ တွေ့မြင်ရမှာကတော့ 0 % to 100 % ပဲ ဖြစ်ပါတယ်။ ဒါဆုံးရင်တော့ JProgressBar နဲ့ ပတ်သက်တဲ့ အချက်အလက် တွေကို အခြေအားဖြင့်တော့ သဘောပေါ်သွားမှာပါပဲ။

က ဒါဆုံးရင်တော့ JProgressBar class ကို လေ့လာကြည့်ကြရအောင်။ JProgressBar class ရဲ့ constructor method တွေကို အရှင်လေ့လာကြတာပေါ့။ အောက်မှာ ရေးပြပေးထားပါတယ်။

- JProgress () - Creates a new progress bar
- JProgress (int, int) - Creates a new progress bar with the specified minimum value and maximum value
- JProgressBar (int,int,int) - Creates a new progress bar with the specified orientation, minimum value and maximum value.

JProgress ကို အသုံးပြုတဲ့အခါမှာတော့ orientation\_

type အနေနဲ့ကတော့ Swing Constants.VERTICAL နဲ့ Swing Constants.HORIZONTAL ဆိတဲ့ class constants type နှစ်ခုရှိ ပေးထားပါတယ်။ JSlider Class မှာတစိပါပဲပဲ။ ဘာမျှ parameter.arguments ကို ထည့်ရေးမထားရင် progress bar ကို horizontal by default အနေနဲ့ ထားပေးမှာပါပဲ။

JProgress Bar တွေကို အသုံးပြုတဲ့အခါမှာ minimum value နဲ့ maximum value တွေကို program bar မှာ ထည့်သွင်းတဲ့အခါမှာ

```
SetMinimum (int);
SetMaximum (int);
```

အစရိတဲ့ method တွေကို အသုံးပြုပြီးတော့ သက်ခိုင်ရာအလိုက် indicate လုပ်လို့ ရပါတယ်။

တကယ်လိုများ စာဖတ်သူဟာ progress bar ကို updateလုပ်ယူခိုင်ပဲ့။ SetValue (int)method ကို အသုံးပြုပြီးတော့ မှ စိတ်ကြိုက်ဖန်တီးလိုလည်း ရပါတယ်။ အဲဒီ value ဟာ progress bar မဲ့ minimum value and maximumကြားမှာ အသုံးပြုနိုင်ဖွေ့က်ပါပဲ။ ဒဲ ဒါဆိုရင် example အနေနဲ့ install လုပ်တဲ့ပဲ့၏ progress bar ကို အသုံးပြုမယ်ဆိုရင်ပေါ့။ အဲဒီအခါ မှာ installation files တွေဟာ upload လုပ်ဖို့အတွက် အောက်ပါအတိုင်း ရေးသားပါလိမ့်မယ်။

```
int FilesDone = getNumber Of Files( );
install.sevalues (fileDone);
```

ဒီအပေါ် code\_line examplesမှာဆိုရင် getNumber OfFiles ( ) method ဟာ install လုပ်တဲ့ software ထဲက how many files have been copied ပေါ့နော်။ Installation လုပ်တဲ့အထဲက files အရေအတွက် ဘယ်လောက်ကို ကူးယူခဲ့ရင် ပေါ့နော်။ အဲဒီအခါမှာ SetValue () method ဟာ progress bar ပေါ့မှာ လာရောက်ပြီးတော့ passed လုပ် ခွင့်ပေးရမှာပေါ့။ မဟုတ်ဘူးလား။ ဒါဟာ အပေါ် code\_line ကို ဥပမာေးပြီး တော့ ရှင်းပြတာပါပဲနော်။

Progress bar မှာဆိုရင် graphic အနေနဲ့ empty box တစ်ခုဟာ text label အနေနဲ့ ပါဝင်ပါတယ်။ သူကတော့ progress bar အလုပ်လုပ်ပဲ့ကို fill up ဖြစ်သွားပဲ့နဲ့ ဖော်ပြုပေးမှာပါပဲ။ အဲဒီ label မှာ displays ဖြစ်ခဲ့ကတော့ percentage % အနေနဲ့ပါပဲ။ 100% ဖြစ်သွားပြီဆိုရင်တော့ progress bar အလုပ်လုပ်တာဟာ completed ဖြစ်သွားတာပေါ့နော်။ အဲဒီလိုဖြစ်အောင်လို့ SetStringPainted (boolean)method ကို ခေါ်ယူ အသုံးပြုရတာပါပဲ။ အဲဒီ method မဲ့ parameter\_arguments နေရာ မှာ boolean type ကို true အနေနဲ့ ရေးရမှာပါပဲ။ တကယ်လို့

arguments ကို false အနေနဲ့ ထားခဲ့မယ်ဆိုရင် label ဟာ ဖော်ပြုပေးခြင်းမရှိပါဘူး။ Off ဖြစ်နေမှာပါပဲ။ ကဲ ဒီတော့ progress bar နဲ့ ပတ်သက်တဲ့ program ထဲတစ်ပုဒ်ကို လက်တွေ့ရေးပြီးတော့ လေ့လာကြရအောင်လား။ Listing 5.2 အနေနဲ့ ရေးသားပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

**Listing 5.2 Demonstrate the Progress bar of JAVA**

```
1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. public class program extends JFrame {
6.
7. JProgress Bar current;
8. JTextArea out;
9. JButton find;
10. Thread runter;
11. int num = 0;
12.
13. public progress () {
14. super ("Progres");
15.
16. setDefaultCloseOperation (JFrame.EXIT_ON_CLOSE);
17. JPanel pane = new JPanel ();
18. pane.setLayout ( );
19. current = new JProgressBar (0,2000);
20. current. SetValue (0);
21. current.SetStringPainted (0);
22. Pane.add (current);
23. SetContentPane (pane);
24. }
25.
26.
27. public void iterate ( ){
28. while (num < 2000){
29. current.SetValue (num);
30. try {
31. Thread.Sleep (1000);
32. } catch (InterruptedException o) { }
33. num += 95
34. }
35. }
36.
37. public static void main (string [ ] arguments){
38. Progress Frame = new Progress ();
39. frame.pack ( );
40. frame.setVisible (true);
41. frame.iterate ( );
42. }
43. }
```



## Analysis

၁။ ဒါ program ကို မရင်းပြခင်မှာ အရင်ပြောချင်တာက လိုင်းနံပါတ် (၈)(၉)(၁၀)မှာ ရေးသားတဲ့ JTextArea, JButton, Thread တွေကို selection color ခြယ်ပြီးတော့ ဖျက်ပစ် လိုက်ပါ။ ဘာကြောင့်လဲဆိတော့ အဲဒီ swing components တွေကို Declose လုပ်ထားပေမယ့် program flow မှာ ထည့်သွင်း အသုံးမပြုထားပါဘူး။ ဒါကြောင့် အဲဒီ code\_line တွေကို ထည့်ပြီးပဲ ရေးရေး ဖျက်ပြီးထားထား အတူတူပါပဲ။ လိုင်းနံပါတ် (၁) ကနေ လိုင်းနံပါတ် (၃)အထိ java.awt.\*; java.awt.event.\*; javax.swing.\*; ဆိတ့် package file (၃) ခုကို Declose လုပ်ပါတယ်။ ကဲ Progress ဆိတ့် Primary class ကို ရေးသားပါတယ်။ အဲဒီ primary class ထဲမှာမဲ့ JProgressbar class ကို current လို့ variable\_name ပေးပြီးတော့ object အဖြစ် ဖော်တီးခဲ့ပါတယ်။ ကဲ ကျွန်ုတ် လိုင်းနံပါတ် (၈) (၉) (၁၀)ကိုတော့ မထည့်နဲ့တော့ပေါ့နော်။ ထည့်ပြီး ရေးသားလည်း ထည့်သွင်းပြီးမှ အသုံးမပြုတာဘဲ။ ဟုတ်ပြီးနော်။ အဲဒီနောက်မှာမှာ Progress class ရဲ့ constructor function () ကို လိုင်းနံပါတ် (၁၃)မှာ ရေးသားထားပါတယ်။ အဲဒီ နောက် title\_bar\_text အနေနဲ့ ပါဝင်လာအောင် super () method ကို အသုံးပြုပြီးတော့ ရေးသားထားပါတယ်။ ဒါဟာ Inheritance method ကို အသုံးပြုထားတာဆိုရင်လည်း မမှားပါဘူး။

၂။ အဲဒီနောက် JFrame\_Windows နဲ့ ပတ်သက်တဲ့ SetDefault CloseOperation (JFrame.EXIT\_ON\_CLOSE); ဆိတ့် method ကို ရေးသားထားပါတယ်။ အဲဒီ နောက် JPanel Class ကို object\_name pane လို့ အမည်ပေးပြီးတော့ ခေါ်ယူပါတယ်။ လိုင်းနံပါတ် (၁၉) မှာ တော့ JProgress Bar တစ်ခုကို variable\_Object တစ်ခု အနေနဲ့ current လို့ ပေးလိုက်ပါတယ်။ ဒါကြောင့်

```
current = new JProgressBar (0,2000);
လို့ ရေးသားထားတာ ဖြစ်ပါတယ်။ အဲဒီနောက် JProgressBar နဲ့ ပတ်သက်တဲ့ SetValue () method နဲ့ SetStringPainted () method အစရှိတာတွေကို ခေါ်ယူ အသုံးချပါတယ်။ ဒါကြောင့်
```

```
current.setValue(0);
current.setStringPainted(0);
```

ဆိုပြီးတော့ လိုင်းနံပါတ် (၂၀)နဲ့ (၂၁)မှာ ရေးသားထားတာ ပေါ့နော်။ဒီ method() တွေကို JProgressBar class ကို

ရေးသား ရှင်းပြခဲ့တဲ့နဲ့က ပါပြီးသားမို့ ထပ်မံ မရင်းပြတော့ ပါဘူး။ ပြီးတော့ ဖန်တီးထားတဲ့ JProgressBar ကို JPanel Class ထဲကို ထည့်သွင်းလိုက်ပါတယ်။ ဒါမှသာ JFrame\_Windows ပေါ်မှာ component အနေနဲ့ ပါဝင်လာမှာပေါ့။ အဲဒီကြောင့် လိုင်းနံပါတ် (၂၂)နဲ့ (၂၃)မှာ

```
pane.add (current);
SetContentPane (pane);
```

လို့ ရေးသားထားတာပဲဖြစ်ပါတယ်။ လိုင်းနံပါတ် (၂၄) က တော့ constructor function () ဖြစ်တဲ့ progress ရဲ့ close brace } ပဲ ဖြစ်ပါတယ်။

၃။ လိုင်းနံပါတ် (၂၅) ကနေ လိုင်းနံပါတ် (၂၅) အထိ ရေးသားထားတဲ့ iterate () method ကတော့ subfunction () တစ်ခုပါပဲ။ ထည့်ပြီးပြောစရိတ္တာကတော့ အဲဒီ iterate () method ထဲမှာ ရေးသားထားတာဟာ Exception Handling အနေနဲ့ Exception Class ကို အသုံးပြုပြီးတော့ ရေးသားထားတယ်ဆိုတာပါပဲ။ အဲဒီအတွက် ကြောင့် အဲဒီ code\_line တွေကို အကြမ်းဖျင်းအားဖြင့်ပဲ ရှင်းပြပါတော့ မယ်။ အဲဒီ method ထဲမှာက while (စဉ်မှာ) num ဟာ 2000 ထက် ငယ်စဉ်မှာပဲနော်။ Num ကို +95 အနေနဲ့ increases လုပ်ပြီး တိုးပါမယ်။ အဲဒီလို့ တိုးနိုင်ဖို့အတွက် လိုင်းနံပါတ် (၂၂) မှာ အောက်ပါအတိုင်း ရေးသားထားပါတယ်။

```
num += 95;
```

အဲဒီ code\_line ကို လေ့လာကြည့်လိုက်ရင် num ဟာ self assign အနေနဲ့ ရေးသားပြီး +95 ကို လာပြီးတော့ ပေါင်းထားတာ ပါပဲ။ ဟုတ်ပြီးနော်။ အဲဒီအခါမှာ progress bar's ရဲ့ SetValue () method ကို လိုင်းနံပါတ် (၂၆)မှာ While Loop ကို တစ်ခါ လာပတ်တိုင်း ဝင်ပြီးတော့ အလုပ်လုပ်ပါတယ်။ အဲဒီ SetValue () method ထဲမှာ num <+ 95 တိုးနေတဲ့ကိုနဲ့>ဟာ arguments အနေနဲ့ ထည့်ထားတာနော်။ အဲဒီအခါမှာ JProgressBar ကို သွားပြီးတော့ အလုပ်လုပ်မှာပါပဲ။ ဒါပေမယ့် ဒီလိုအလုပ်လုပ်တာ ဟာ ဆက်တိုက်တော့ အလုပ်မလုပ်ပါဘူး။ လိုင်းနံပါတ် (၂၁)မှာ ရေးသားထားတဲ့ try block ထဲမှာ ပါဝင်တဲ့

```
Thread.Sleep (1000);
```

ဆိုတဲ့ Thread Class ထဲမှာ ပါဝင်တဲ့ Sleep () method ကြောင့် object ဟာ [JProgressBar Component ဟာ] စဉ်နဲ့ အနည်းငယ် ခြားပြီးတော့ sleep ဖြစ်နေမှာပါပဲ။

**Sleep () method :** The current running Thread object to sleep for the specified amount of time (mili second). The sleeping Thread object reatains whatever locks it may process.

အခု ရေးပြုခဲ့တာကတော့ sleep ( ) method ကို အခြေခံ ရှင်းပြတာပဲ ဖြစ်ပါတယ်။ ရှေ့လာမယ့် သင်ခန်းစာတွေမှာ သေသေ ချုံချုံ ထုတ်မဲ့ ရှင်းပြပေးပါမယ်။ ကဲ ဒီတော့ main () function ကို ဆက်လက် လေလာကြရအောင်။

၄။ Main( ) function ကိုလိုင်းနံပါတ် (၃၇)မှာ ရေးသားထားပါတယ်။ အပေါ်မှာ ဖော်တီးခဲ့တဲ့ Progress ဆိုတဲ့ Class ကို object\_name frame လို ပေးပြီးတော့ ၁၅ယူပါတယ်။ အဲဒီအခါမှာ လိုင်းနံပါတ် (၁၃) မှာ ရေးသားထားတဲ့ constructor function ဟာ အထိအလျောက် ဝင်ရောက်လာမှာပါပဲ။ ဒီတော့ အဲဒီ function () ထဲမှာ ပါဝင်တဲ့ JFrame ကို ဖော်တီးတာတွေ Jprogress Bar ကို တပ်ဆင်တာတွေ အစရိုတဲ့ constructor function () ထဲမှာ ပါဝင်တဲ့ code\_line တွေဟာ လာပြီးတော့ အလုပ်လုပ်နေပြီးလေ့နေ၏။ ဒီအခါမှာ JProgress Bar နဲ့ ပတ်သက်တဲ့ JFrame ပေါ်လာပြီးတော့ အလုပ်လုပ်အောင်

```
Frame.pack();  
Frame.setVisible(true);
```

အစရိတ္တာ method တွေကို အသုံးချထားပါတယ်။ ဒါပေမယ့်  
လိုင်းနဲ့ပါတ် (၄)မှာတော့

```
Frame.iterate();
```

ခိုတဲ့အပေါ်မှာ ရေးသားခဲ့တဲ့ iterate ( ) function ကို  
ခေါ်ယူလိုက်ပါတယ်။ Iterate ( ) function ဟာ progress  
class ရဲ့ member function ဖြစ်တာကြောင် । Progress

class ကို object\_name frame လို အမည်ပေးထားတာ  
ကြောင့် object\_name(.)dot ခံပြီးတော့ ၁။၂။ထားတာ  
ပါပဲ။ အဒီအခါမှာ JProgress Bar ပေါ်မှာ ပါဝင်တဲ့ အလုပ်  
လာရောက်လုပ်ရမယ့် looping structure နဲ့ thread class  
ထဲက method တွေဟာ ဝင်ရောက်အလုပ်လုပ်ပါပြီ။ ဒါဆို  
ရင်တော့ ရင်းပြုနော်။

ကဲ code\_line တွေကို Analysis လုပ်ပြီးသွားရင်တော့  
အဒီ code\_line တွေကို Notepad ထဲမှာ သွားပြီးတော့ ရှိက်  
သွင်းလိမ်ပါ။ အဒီ code\_line တွေကို ရေးသားပြီးသွားရင်တော့  
file ကနေ save ကို တောင်းပါ။ Save Dialog Box ကျလာရင်  
တော့ ပေးရမယ့် file\_name ကတော့ အခု program ရဲ့ class\_  
name အတိုင်း progress ဆိုတဲ့ အမည်နဲ့ progress.java ဆိုတဲ့  
အမည်ကိုပေးပြီးတော့ save လုပ်ပါ။ ဒါဆိုရင်တော့ C:\j2sdk  
1.4.1\_01\bin ဆိုတဲ့ folder အောက်မှာ progress.java ဆိုတဲ့  
code\_file ဟာ တည်ဆောက်ပြီးသွားပါ။

အဲဒိုလိုပြီး Notepad ထဲမှာ code\_line တွေ ရေးသားပြီး  
တော့ save လုပ်ထားပုဂ္ဂိုလ် Figure 5.7 အနေနဲ့ပဲ ရေးသားဖော်ပြ  
ပေးထားပါတယ်။ လျေလာကြည်ပါ။

```

Progress.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;

public class Progress extends JFrame {
    JProgressBar current;
    JTextArea out;
    JButton find;
    Thread runner;
    int num = 0;

    public Progress() {
        super("Progress");
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JPanel pane = new JPanel();
        pane.setLayout(new FlowLayout());
        current = new JProgressBar(0, 2000);
        current.setValue(0);
        current.setStringPainted(true);
        pane.add(current);
        setContentPane(pane);
    }

    public void iterate() {
        while (num < 2000) {
            current.setValue(num);
            try {
                Thread.sleep(1000);
            } catch (InterruptedException e) {
                num += 95;
            }
        }
        Progress frame = new Progress();
        frame.pack();
        frame.setVisible(true);
        frame.iterate();
    }
}

```

Figure 5.7 Write the program code\_line in the Notepad, Progress.java

ကဲ အပေါ်မှာ ဖော်ပြခဲ့တဲ့ code\_file အနေနဲ့ ရေးသားထားပုံ  
ကို လေ့လာပြီးရင် Start → All programs → Accessories  
→ Command Prompt ကို ဖွေ့ပါ။ ကျွန်ုတ်တော့ Command Prompt ကို Desktop Screen ပေါ်မှာ shortcut အနေနဲ့  
တင်ထားပါတယ်။ အဲဒီတော့ Program Compile & Run လုပ် ခါနီးရင် လွယ်ကူတာပေါ့နော်။ ကဲ Command Prompt (DOS)  
Windows ပွဲလာခဲ့ရင်ပေါ့။ အောက်ပါ command\_line နဲ့  
directory ကို ပြောင်းပါ။

**CD C:\j2sdk1.4.1\_01\bin ↵**

ကဲ အဲဒီအခါမှာ command\_line windows ထဲမှာ ပေါ်နေ  
တဲ့ သက်တာဟာ အောက်ပါအတိုင်း ပြောင်းလဲသွားပါ၍။

**C:\j2sdk1.4.1\_01\bin>**

ဟုတ်ပြော်နော်။ အဲဒီအချိန်မှာ အောက်ပါ command ကို ရိုက်  
သွေးပြီးတော့ အပေါ်မှာ ဖန်တီးရေးသားခဲ့တဲ့ progress.java  
ဆိုတဲ့ program ကို compile လုပ်လိုက်ပါ။

**java Progress.java ↵**

အဲဒီအခါမှာ compile လုပ်လို့ error ရှိခဲ့ရင်တော့ လိုင်း  
နံပါတ်နဲ့တက္က လာရောက်ပြီးတော့ ဖော်ပြုပေးမှာပါ။ အဲဒီ Notepad  
ထဲမှာပဲဖြစ်ဖြစ် Save As လုပ်ပြီးတော့ နောက်ထပ် ပြန်ပြီး  
တော့ compile ပြန်လုပ်ပေါ့နော်။ အကယ်၍ error မရှိတော့ဘူး

ဆိုရင်ပေါ့နော်။ Prompt သက်တာဟာ အောက်ပါအတိုင်းပဲ ပြန်  
ပေါ်လာမှာပါပဲ။ လွှဲလာကြည့်ပါ။

**C:\j2sdk1.4.1\_01\bin>**

အဲဒီလို ပြန်ပေါ်လာရင်တော့ အောက်ပါ command နဲ့ run  
လုပ်လိုက်ပါလား။ Program Output ထွက်လာအောင်ပေါ့နော်။

**java Progress ↵**

ကဲ အဲဒီလိုမျိုး command\_line windows ထဲမှာ compile  
& run လုပ်လိုက် Figure 5.8 အနေနဲ့ ဖော်ပြုပေးထားပါတယ်။  
လွှဲလာကြည့်ပါ။



Figure 5.8 Compile & Run of the Progress.java

ကဲ အခုဖော်ပြုပေးထားတဲ့ Figure 5.8 ကတော့ compile  
& run လုပ်နေတဲ့ ပုံပါပဲ။ အဲဒီ Figure ထဲကအတိုင်း compile  
& run လုပ်လိုက်ရင်တော့ အောက်ပါအတိုင်း JFrame\_Windows  
ကို တွေ့မြင်ရမှာပါပဲ။



Figure 5.9 Run the Program, Progress.java

ကဲ အခုဖော်ပြထားတဲ့ Figure 5.9 ကတော့ Listening  
5.2 က code\_line တွေကို အသုံးပြုပြီးတော့ တည်ဆောက်ထား  
တဲ့ program ထုတေသနပါပဲ။ ကဲ ဒီတော့ ဆက်လက်ပြီးတော့  
Chapter (5) ရဲ့ sub\_index အနေနဲ့ ပို့ချေပေးထားတဲ့ 5.3 Using  
the JMenus ဆိုတဲ့ သင်ခန်းစာကို ဆက်လက် လွှဲလာကြည့်ကြရ  
အောင်လား။

KYAW ZAYAR LAY [ IT ENGINEERING ]



# JAVA DEVELOPER GUIDE

## Chapter (5)

### 5.3 USING THE J MENUS

Menu Bar တွေနဲ့ပတ်သက်တဲ့ program လေးတွေကို စာဖတ်သူတို့ အသုံးပြုဖူးကြောပါ။ Visual Basic 6.0မှာလောက်တော့ မလွယ်ကူတာ အမှန်ပါပဲ။ ဒါပေမယ့် ခက်ခဲလှတာတော့ လည်း မဟုတ်ပါဘူး။ ဒီတော့ Menu တွေကို ဖန်တီးရာမှာ Supportလုပ်ပေးတဲ့ components တွေကို လေ့လာကြရအောင်လား။ ဒီ components တွေဟာ အလုပ်လုပ်ရာမှာ တစ်ခုနဲ့တစ်ခုဟာ ဆက်နှစ်ယြိုင်းတော့ အလုပ်လုပ်ကြတာပါပဲ။ တစ်နည်းပြာရရင်တော့ components that work in conjunction with each other လို့ ပြောရင်လည်း ရှာဘပြုဖော်။ ကဲ အောက်မှာ ရေးသားထားတာကို လေ့လာလိုက်ပါပြီး။

- JMenuItem - an item on a menu
- JMenu - a drop-down menu that contains one or more JMenuItem components, other interface components, and separators, lines that are displayed between items
- JMenuBar - a container that holds one or more JMenu components and displays their names

ကဲ အချင်းပြုခဲ့တဲ့ components(၃) ခုကို လေ့လာကြည့်ရအောင်လား။ ဒါမှာသာ programရေးတဲ့အခါမှာ လက်တွေ့ထည့်သွင်း အသုံးချလို့ရမှာပေါ့ဖော်။

#### • JMenuItem

JMenuItem component ကတော့ JButtonနဲ့ဆင်တူပါတယ်။ ဒီတော့ JButton တွေ ဖန်တီးတဲ့အခါမှာ constructor method ထဲမှာ ထည့်သွင်းရေးသားရသလိုပါပဲ။ JMenuItem

တွေ ဖန်တီးတဲ့အခါမှာ အောက်ပါ method ကို ၇၅ယူရပါမယ်။ Constructor method မှာကိုက (၃) မျိုး ကွဲပြားနေတယ်ဖော်။

- JMenuItem (String) for a text item
  - JMenuItem (Icon) for an item that displays a graphic file
  - JMenuItem (String, Icon) for both
- ဆိုပြီးတော့ constructor overloading အနေနဲ့ ရေးသားဖန်တီးလို့ ရပါတယ်။ ကဲ ဒီတော့ အောက်ပါအတိုင်း JMenuItem နဲ့ ဖန်တီးထားတာကို လေ့လာကြည့်ကြရအောင်လား။

```
JMenuItem j1 = new JMenuItem("New");
JMenuItem j2 = new JMenuItem("Open ....");
JMenuItem j3 = new JMenuItem("Close");
JMenuItem j4 = new JMenuItem("Save ....");
JMenuItem j5 = new JMenuItem("Save As Template..");
JMenuItem j6 = new JMenuItem("Import ....");
JMenuItem j7 = new JMenuItem("Export ....");
JMenuItem j8 = new JMenuItem("Page Setup ....");
JMenuItem j9 = new JMenuItem("Print ....");
JMenuItem j10 = new JMenuItem ("Use As Default Message Style");
JMenuItem j11 = new JMenuItem("Exit");
```

ဒါဟာ JMenuItem () constructor method ကို အသုံးပြုပြီးတော့ သက်ဆိုင်ရာ Menu တွေကို ဖန်တီးပြထားတာပါပဲ။

ဒီ JMenuItem တွေဟာ program ကို အသုံးပြုစဉ်အခါမှာ အသုံးချမယ့် command တွေကို ထည့်သွင်း ရေးသားထားတဲ့ ဖြစ်ပါတယ်။ အခု ရေးသားခဲ့တဲ့ command တွေဟာ သက်ဆိုင် ရာ JMenu ကို 1-Click နိုင်လိုက်ရင် drop-down အနေနဲ့ ကျေလာမှာပေါ်နော်။ ကဲ ဟုတ်ပြုလေ။ ဒီတော့ JMenu ကို ဖန်တီးထည့်ကြရအောင်လား။

### • JMenu

JMenu ကို ဖန်တီးရာမှာတော့ အပေါ်က ဖန်တီးခဲ့တဲ့ JMenuItem တွေကို drop-down menu အနေနဲ့ ဖန်တီးထည့်သွင်းရာမှာ အသုံးချဖို့ပါပဲ။ အသုံးချရမယ့် method ကတော့ JMenu (String) Constructor method ပါပဲ။ Menu ခေါင်းစဉ်အနေနဲ့ ပါဝင်ချချင်တဲ့ name (Menu\_Name) ကို အဲဒီ constructor method ပဲ arguments နေရာမှာ ထည့်သွင်းရမှာ ပါပဲ။ အဲဒီ name ဟာ menu bar ရဲ့ အပေါ်မှာ လာပြီးတော့ ပေါ်ပေးနေရာပါပဲ။ Program ကို ရေးသားစဉ်မှာ JMenu container ကို အသုံးချပြီးရင်တော့ add (JMenuItem) method ကို ခေါ်ယူအသုံးချရမှာပါပဲ။

အဲဒီ add ( ) method ကို ခေါ်ယူ အသုံးချပြီးဆိုရင်တော့ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ JMenuItem တွေရဲ့ Object\_Name ကို add ( ) method ရဲ့ parameter\_arguments နေရာမှာ ထည့်သွင်းရေးသားရပါမယ်။ ဒါဟာ တစ်နည်းပြောရရင်တော့ အားလုံးအတွက် Group\_Name ပါပဲ။ သူမှာ ကိုယ်ပိုင် Menu\_item ကို မပိုင်ဆိုင်ပါဘူး။ ဒါဆိုရင်တော့ JMenu ကို အသုံးပြု ရေးသားထားတဲ့ method အကြောင်းကို အခြေခံအားဖြင့်တော့ နားလည်သေားပေါ်သွားမှာပါ။

ဂဲ အမိကအချက်တစ်ချက် ကျော်နေတာကတော့ ကျွန်တော်တို့ ဖန်တီးမယ့် Menu ထဲမှာ add Separator ( ) method ကို ထည့်သွင်း အသုံးပြုရမှာပါပဲ။ ဘာကြောင့်လဲဆိုတာကို ကျွန်တော်ရှင်းပြပါမယ်။

**Note :** Visual Basic Language ကို အသုံးပြုစဉ်တုန်းက ဖန်တီးထားတဲ့ Menu ထဲမှာ Separators တွေကို ထည့်သွင်းဖန်တီးချင်ရင် Menu DialogBox ရဲ့ Caption နေရာမှာ - မိမိထည့်သွင်းချင်တဲ့ Menu\_Name Name နေရာမှာ - mnu1/ mnu2/ mun3... အစရှိသဖြင့် ထည့်သွင်း ရေးသားလိုက်ရုံးပါပဲ။ ဒါဟာ Visual Basic Language က programmer တွေအတို့ အဓိုကတဲ့ အသုံးချဖူးမှာပါ။ အမှန်ကတော့ Separators ကို အသုံးပြုတော့ ဘာ menu အပ်စုတဲ့မှာ ရောထွေးမနေအောင် line separators လေးတွေကို ထည့်သွင်းထားတာပါပဲ။ ဘယ် language နဲ့ ပဲ

program ကို တည်ဆောက်ပါ။ Menu တွေ ဖန်တီးပြီ ဆိုတာနဲ့ line separators တွေကို ထည့်သွင်းဖော်တယ်။ သက်ဆိုင်ရာ language ရဲ့ method အရပေါ်နော်။ ကဲ လေ့လာ ကြည့်ရအောင်။

အခု ကျွန်တော်က Java Language ကို အသုံးပြုပြီးဆိုရင် JMenu ကို တည်ဆောက်မယ်ဆိုတော့ Visual Basic မှာလို့မျိုး Name & Caption တွေကို သီးသန် ဖန်တီးစရာ မလိုတော့ဘူး ပေါ့။ AddSeparator ( ) method ကို ခေါ်ယူ အသုံးချယ်ကို ရုံးပါပဲ။ ဒါဆိုရင်တော့ ကျွန်တော်တို့ ဖန်တီးမယ့် Menu ထဲမှာ line separators တွေ ပါဝင်ပေးရမှာပါပဲ။ ဒါပေမယ့် Objetc\_Name [JMenu နဲ့ Object\_Name] ခံရေးရမှာနော်။ ကဲပြီးတော့ အရှေ့မှာ JMenuItem အနေနဲ့ ဖန်တီးထားတဲ့ menu\_label အနေနဲ့ ရေးသားထားတဲ့ string တွေကို add (String) method ကို ခေါ်ယူ အသုံးချရမှာပါပဲ။ ဒီတော့ ရေးသားရမှာ text as an arguments အနေနဲ့ အရှေ့မှာ ဖန်တီးခဲ့တဲ့ JMenuItem ရဲ့ Object\_Name တွေကို ထည့်သွင်း ရေးသားရမှာပေါ်နော်။ အဲဒီကြေားထဲက addSeparator ( ) method ကိုလည်း ထည့်သွင်းရေးသားရမှာပါ။ ကဲ ဒီတော့ လေ့လာကြည့်ကြရအောင်လား။ အောက်မှာ sample ရောဖြေပေးထားပါတယ်။

```
JMenu m1 = new JMenu ("File");
JPanel pane= new JPanel ();
m1.add(j1);
m1.add(j2);
m1.addSeparator ();
m1.add(j3);
m1.addSeparator ();
m1.add(j4);
m1.add(j5);
m1.addSeparator ();
m1.add(j6);
m1.add(j7);
m1.addSeparator ();
m1.add(j8);
m1.addSeparator ();
m1.add(j9);
m1.add(j10);
m1.addSeparator ();
m1.add(j11);
```

ကဲ အခု ရေးပြခဲတာကတော့ အပေါ်က JMenuItem နဲ့ ဖန်တီးထားတဲ့ Menu item တွေကို JMenu ရဲ့ အောက်ကို ထည့်သွင်းလိုက်ပြုနော်။ ဆက်လက်ပြီးတော့ JMenuBar ကို ဆက်လက်အသုံးချကြည့်ကြရအောင်။ JMenuBar ကတော့ JMenu အနေနဲ့

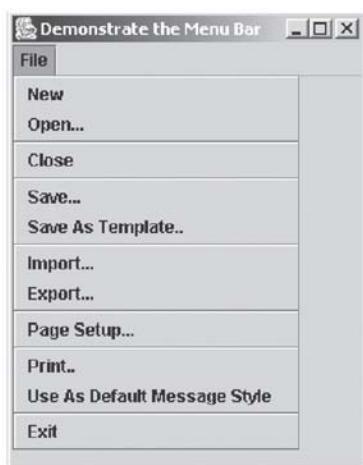
အခု ဖန်တီးလိုက်တဲ့ file ဆိုတဲ့ Menu ကို JMenuBar container ထဲကို ထည့်သွင်းရမှာပါပဲ။ JMenuBar ထဲကို JMenu တွေ တစ်ခုထက်မက အမြဲးကို ထည့်သွင်းလို့ ရနိုင်ပါတယ်။ ဥပမာ အသုံးချု application program ငယ်တစ်ခုမှာ File Edit View Help အစရိသဖြင့် JMenu တွေ အများအပြား ပါဝင်နေတာ ကို တွေ့ဖြင့်ရမှာပေါ်နေ၏။ တစ်နည်းပြောရရင်တော့ JMenu နဲ့ ဖန်တီးထားတဲ့ Menu တွေကို menu bar [application's title bar] အနေနဲ့ တပ်ဆင်ပေးမှာပါပဲ။

ကဲ JMenuBar ကို ဖန်တီးမယ်ဆိုရင် JMenuBar ( ) constructor method ကို arguments ထည့်သွင်းရေးသားစရာ မလိုဘဲ ရေးသားရမှာပါပဲ။ ဤဦးတော့ အပေါ်မှာ ခုတီယာအဆင့်အနဲ့ ဖန်တီးခဲ့တဲ့ JMenu ကို add ( ) method နဲ့ ထည့်သွင်းပြီး တော့ ရေးသားရမှာပါပဲ။

**Note :** ဒီတော့ JMenuItem တွေ ဖန်တီးမယ်။ JMenu ထဲကို ထည့်သွင်းပြီး Group\_Name လေးမယ်။ အဲဒါတွေကို JMenuBar နဲ့ တပ်ဆင်မယ်။ ဒါပြီးဂျင်တော့ frame\_windows ပေါ်မှာ တပ်ဆင်ဖို့အတွက်ကတော့ SetJMenuBar( )method ကို ဆက်လက်ပြီးတော့ အသုံးချုရမှာပဲဖြစ်ပါတယ်။ အခု ဆက်လက် ရေးပြုမယ့် sample code\_line တွေဟာ အပေါ်က ပထမအဆင့် ခုတီယာအဆင့် ရေးသားတာကို ဆက်လက်ပြီးတော့ ရေးသားပြုထားတာပါပဲ။ လေ့လာကြည့်ပါ။

```
JMenuBar bar = new JMenuBar ( );
bar.add (m1);
SetJMenuBar (bar);
```

ကဲ အဲဒီလိုသာ ရေးသားလိုက်ရင်တော့ အောက်မှာ တွေ့ဖြင့်ရမယ့် Figure 5.10 ကအတိုင်း JMenuBar တစ်ခုဟာ အလွယ်တကူ ဖန်တီးလို့ရတာကို တွေ့ဖြင့်ရမှာပါပဲ။ လေ့လာကြည့်လိုက်ပါး။



က အခု အပေါ်မှာ ရေးပြုခဲ့တဲ့ code\_line တွေကို အခြေခံပြီးတော့ ဖော်ပြထားတဲ့ Figure အတိုင်း Menu ကို ဖန်တီးရယူလို ရပါတယ်။ ဒါပေမယ့် အချုပ်နှင့်တီးလိုက်တဲ့ Menu တွေထို ရွေးချုပ်လိုက်ပေမယ့် nothing happens in response ပေါ်နေ၏။ ဘာမျှ ပြန်ပြီးတော့ Response ပြန်မပေးပါဘူး။ ဒါက Menu ကို ဖန်တီးပဲ ဖန်တီးထားတာကိုး။ အထဲက code\_line တွေကို ရေးမထားတာပဲ။ အဲဒီလို Program တွေကို တည်ဆောက်ပြီးတော့ အမှန်တက်ယ် အလုပ်လုပ်အောင်ကတော့ ရရှိလာမယ့် သင်ခန်းစာ တွေမှာ Programming Theory တွေနဲ့အတူ "Responding to User Input" ဆိုတဲ့ သင်ခန်းစာမှာ လေ့လာကြည့်ကြတာပေါ့။ အခုတော့ အရောက sample code\_line တွေနဲ့ အတူ Program ရေးပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။ Listing 5.3 အနေနဲ့ ရေးပြထားတဲ့ program ကို လေ့လာကြည့်ကြရအောင်လား။

**Listing 5.3 Demonstrate the Menu Bar in JAVA**

```
1. import javax.swing.*;
2. import java.awt.*;
3. public class Menus extends JFrame
4. {
5.     public Menus ( )
6.     {
7.         super ("Demonstrate the Menu Bar");
8.         setSize(250,320);
9.         setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
10.
11.        JMenuItem j1=new JMenuItem ("New");
12.        JMenuItem j2=new JMenuItem ("Open...");
13.        JMenuItem j3=new JMenuItem ("Close");
14.
15.        JMenuItem j4 = new JMenuItem ("Save...");
16.        JMenuItem j5 = new JMenuItem ("Save As Template..");
17.        JMenuItem j6 = new JMenuItem ("Import...");
18.        JMenuItem j7 = new JMenuItem ("Export....");
19.        JMenuItem j8 = new JMenuItem ("Page Setup....");
20.        JMenuItem j9 = new JMenuItem ("Print....");
21.        JMenuItem j10 = new JMenuItem ("Use As Default Message Style");
22.        JMenuItem j11 = new JMenuItem ("Exit");
23.        JMenu m1 = new JMenu ("File");
24.        JPanel pane = new JPanel ( );
25.        m1. add (j1);
26.        m1. add (j2);
27.        m1. add (j3);
28.        m1. addSeparator( );
```

```

29. m1.add(j4);
30. m1.add(j5);
31. m1.addSeparator();
32. m1.add(j6);
33. m1.add(j7);
34. m1.addSeparator();
35. m1.add(j8);
36. m1.addSeparator();
37. m1.add(j9);
38. m1.add(j10);
39. m1.addSeparator();
40. m1.add(j11);
41. JMenuBar bar = new JMenuBar();
42. bar.add(m1);
43. SetJMenuBar(bar);
44. Show();
45. }
46.
47. public static void main(String arg[])
48. {
49. Menus m = new Menus();
50. }
51. }
```

### Analysis

၁။ ဒီ program မှာကတော့

```

import javax.swing.*;
import java.awt.*;

ဆိတဲ့ package flie( ) ခုကို import & declareလုပ်ပြီး
ခေါ်ယူထားပါတယ်။ အဲဒီနောက် Menus ဆိတဲ့ primary
class ကို ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၃) မှာပါ။
Extends လုပ်ထားတာကတော့ JFrame class ကိုနော်။
အဲဒီနောက် Menus ( ) Constructor function ကို
ရေးသားပါတယ်။ အဲဒီနောက် JFrame Windows ရဲ့ Title
Bar မှာ ပါဝင်ရမယ့် စာသားအတွက်
Super ("Demonstrate the Menu Bar");
လို့ ရေးသားပေးထားပါတယ်။ အဲဒီနောက် JFrame
Windows ရဲ့ အချယ်အစားအတွက် width=250, height
=320 နဲ့ ဖြစ်အောင်
setSize (250,320);
ဆိပ်ပြီးတော့ ရေးသားထားပါတယ်။ ပြီးတော့ JFrame
windows ကို close လုပ်လို့ရအောင်လို့
SetDefaultCloseOperation (JFrame.EXIT_
ON_CLOSE);
method ကို ခေါ်ယူ အသုံးချပေးထားပါတယ်။ အဲဒီနောက်
မှာတော့ Sample Code_line အနေနဲ့ ရေးပြုခဲ့တဲ့ JMenuItem
Item ကိုအသုံးချပ်ပြီးတော့ ရေးသားထားတဲ့ text_label &
JMenuItem တွေကို လိုင်းနံပါတ် (၁၁) ကနေ လိုင်းနံပါတ်
```

(၂၁)အထိ ရေးသားထားပါတယ်။ အဲဒီ code\_line တွေကို sample ရေးပြုပြီးသားမို့ ပြန်ရေးမပြတော့ပါဘူး။ ဒါဟာ ကျွန်ုတ်တို့ ဖန်တီးမယ့် Menu အတွင်းမှာ ပါဝင်မယ့် Menu တွေပါပဲ။ ကဲ အဲဒီပြီးတော့ လိုင်းနံပါတ် (၂၂) မှာမဲ့

JMenu m1 = new JMenu ("File");
လို့ ရေးသားထားပါတယ်။ ဒါဟာ ဖန်တီးမယ့် JMenu ပါပဲ။ Menu\_Name ကတော့ file ပေါ်နော်။ ရှင်းပြီးနော်။
၂။ ကဲ ပြီးတော့ JPanel Class ကို ခေါ်ယူ အသုံးပြုရအောင်လို့ လိုင်းနံပါတ် (၂၃) မှာ အောက်ပါအတိုင်း ရေးသားပေးထား ပါတယ်။

```

JPanel pane = new JPanel ();
ဒါဟာ JPanel Class ကို object_name pane လို့ ပေးပြီး
ခေါ်ယူထားတာပါပဲ။ အဲဒီရဲ့ နောက်မှာမဲ့ ခေါ်ယူ ဖန်တီးထား
တဲ့ JMenu နဲ့ ယူဉ်တွဲပြီးတော့ add () method နဲ့ ထည့်
သွင်း ရေးသားပါတယ်။ ဒါကြောင့် လိုင်းနံပါတ် (၂၄) ကနေ
လိုင်းနံပါတ် (၂၀) အထိ ရေးသားထားတာပါပဲ။ ဒီ code_
line တွေကြားမှာလည်း addSeparator( ) method
ကိုလည်း ထည့်သွင်း ရေးသား အသုံးပြုထားပါတယ်။ လိုင်း
နံပါတ် (၂၆)၊ (၂၁)၊ (၃၁)၊ (၃၇)၊ (၃၉)၊ (၃၉) စတဲ့ လိုင်း
တွေမှာ တွေ့မြင်ရမှာပါ။ ဒါတွေကို ကျွန်ုတ်အနေနဲ့ ရေးမပြ
တော့ဘူး။ Sample code_line အနေနဲ့ ရေးပြုပေးထား
ပါတယ်။ ပြီးတော့ လိုင်းနံပါတ် (၂၁) မှာ
```

JMenuBar bar = new JMenuBar();
ဆိပ်ပြီးမဲ့ JMenuBar ကို ခေါ်ယူ အသုံးချထားတာကို တွေ့ရ
မှာပါ။ အဲဒီလိုင်းနံပါတ် (၂၁) ကနေ (၂၅) အထိကိုလည်း
အရေးမှာ sample code\_line အနေနဲ့ ရေးသား ဖော်ပြုပြီး
သားပါ။ ပြန်ပြီး လေ့လာကြော်ပါ။ အကျယ်ချဲ့ မရှင်းပြတော့
ပါဘူး။ အဲဒီနောက် လိုင်းနံပါတ် (၂၅) မှာ show( )
method ကို ခေါ်ယူ အသုံးချထားပါတယ်။ အဲဒီနောက်
Menu( ) constructor method ရဲ့ close brace } ကို
လိုင်းနံပါတ် (၂၅)၏ လာပိတ်ထားပါတယ်။ အဲဒီနောက်
main () function ကို လိုင်းနံပါတ် (၂၇) မှာ ရေးသားထား
ပါတယ်။ အဲဒီ main () function ထဲမှာမဲ့ အပေါ်က class
Menus ကို object\_name က ထိပေးပြီးတော့ ခေါ်ယူလိုက်
ပါတယ်။ ဒါဆိုရင် အပေါ်က ဖန်တီး ရေးသားခဲ့တာတွေ
အားလုံး ပါဝင်လာပြုပေါ်နော်။ ပြီးတော့ main () function
ရဲ့ close brace } နဲ့ primary class ရဲ့ close brace
တွေကို လာပိတ်ပေးထားပါတယ်။

ကဲ အခုရေးပြထားတဲ့ code\_line တွေကို Analysis လုပ်ပြီး
သွားရင်တော့ စာဖတ်သူအနေနဲ့ အောက်မှာ ဖော်ပြထားတဲ့ Figure



5.11 အတိုင်း Note\_Pad ထဲမှာ သွားပြီးတော့ ရိုက်သွင်းပါ။  
ပြီးရင်တော့ C:\j2sdk1.4.1\_01\bin ထဲမှာ save လုပ်ရပါမယ်။  
Save လုပ်ရမယ့် File\_Name ကတော့ Menus.java လိုပေးရမှာ  
နော်။ ကဲ code\_line ထွေ ရေးသားထားတာကို လေ့လာကြည့်ပါ။

```

Menus.java - Notepad
File Edit Format View Help
import java.awt.*;
import javax.swing.*;
public class Menus extends JFrame
{
    public Menus()
    {
        super("Demonstrate the Menu Bar");
        setSize(250, 320);
        setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        JMenuItem[] m1 = new JMenuItem[10];
        JMenuItem[] m2 = new JMenuItem[10];
        JMenuItem[] m3 = new JMenuItem[10];
        JMenuItem[] m4 = new JMenuItem[10];
        JMenuItem[] m5 = new JMenuItem[10];
        JMenuItem[] m6 = new JMenuItem[10];
        JMenuItem[] m7 = new JMenuItem[10];
        JMenuItem[] m8 = new JMenuItem[10];
        JMenuItem[] m9 = new JMenuItem[10];
        JMenuItem[] m10 = new JMenuItem[10];
        m1[0] = new JMenuItem("New..."); m1[1] = new JMenuItem("Open..."); m1[2] = new JMenuItem("Close..."); m1[3] = new JMenuItem("Save..."); m1[4] = new JMenuItem("Save As Template..."); m1[5] = new JMenuItem("Import..."); m1[6] = new JMenuItem("Export..."); m1[7] = new JMenuItem("Page Setup..."); m1[8] = new JMenuItem("Print..."); m1[9] = new JMenuItem("Use As Default Message Style");
        m2[0] = new JMenuItem("Exit");
        menu1 = new JMenu("File");
        JPanel pane = new JPanel();
        m1[0].addActionListener(pane);
        m1[1].addActionListener(pane);
        m1[2].addActionListener(pane);
        m1[3].addActionListener(pane);
        m1[4].addActionListener(pane);
        m1[5].addActionListener(pane);
        m1[6].addActionListener(pane);
        m1[7].addActionListener(pane);
        m1[8].addActionListener(pane);
        m1[9].addActionListener(pane);
        menu1.add(m1[0]);
        menu1.add(m1[1]);
        menu1.add(m1[2]);
        menu1.add(m1[3]);
        menu1.add(m1[4]);
        menu1.add(m1[5]);
        menu1.add(m1[6]);
        menu1.add(m1[7]);
        menu1.add(m1[8]);
        menu1.add(m1[9]);
        menu1.add(m2[0]);
        menuBar = new JMenuBar();
        bar.add(menu1);
        setMenuBar(bar);
        show();
    }
    public static void main(String args[])
    {
        Menus m = new Menus();
    }
}

```

Figure 5.11 Write the program code\_line in the Notepad, Menus.java

ကဲ အရှေ့က Figureမှာ ဖော်ပြထားတဲ့အတိုင်း code\_line ထွေ ရေးသားပြီးသွားရင် Start → All programs → Accessories → Command prompt ထဲမှာ ဖွင့်ပြီးတော့ DIR ကို ဖြောင်းပါ။ ဒါမှမဟုတ်ရင် Start → Run ကနေ Run Dialog Box ကျလာ ရင် command.com ကို ရိုက်သွင်းပြီး Run ရင်လည်း ရတယ် နော်။ DIR ပြောင်းကြရအောင်။

CD C:\j2sdk1.4.1\_01\bin ↵

ဒါဆိုရင် DIR\_path ဟာ အောက်ပါအတိုင်း ပြောင်းလဲသွားမှာ ပါ။

C:\j2sdk1.4.1-01\bin>

က ဒီတော့ အောက်ပါ command နဲ့ compile လုပ်ပါ။

java c Menus.java ↵

Error မရှိဘူးဆိုရင်တော့ Run လုပ်ရပဲပေါ့။ Error ရှိခဲ့ရင်တော့ အောက်မှာပါလာတဲ့ Error လိုင်းနဲ့ပါတဲ့အတိုင်း လိုက်ရှုပြီး Error ကို ပြင်ပြီးတော့ Save As လုပ်ပါ။ ပြီးရင် ဖြန်ပြီးတော့

java c Menus.java ↵

ဆိုပြီးတော့ compile လုပ်ပေါ့နော်။ တကယ်လို့ error မရှိတော့ ဘူးဆိုရင် အောက်ပါအတိုင်း run လုပ်ပါ။

#### java Menus ↵

အဲဒီလိမ့်း Compile & Run လုပ်ပုံကို Figure 5.12 အနေနဲ့ ရေးပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

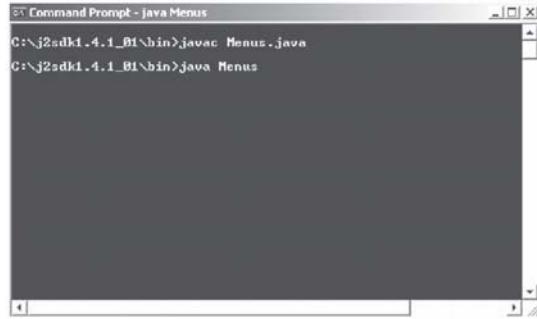


Figure 5.12 Compile & Run of the Menus.java

က အပေါ်က ဖော်ပြထားတဲ့အတိုင်း Complie & Run လုပ်လိုက်ရင် အရှေ့မှာ Sample ပြထားခဲ့တဲ့အတိုင်း Demonstrate the Menu Bar ဆိုတဲ့ program ဟာ လာပြီးတော့ Run နေမှာပါပဲ။ Figure 5.13 အနေနဲ့ ရေးသား ဖော်ပြပေးထားပါတယ်။ လေ့လာကြည့်ပါ။

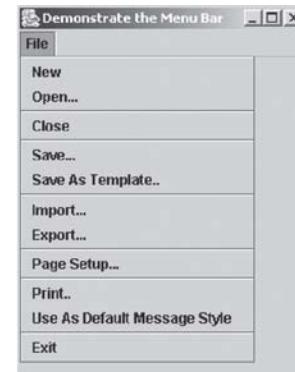


Figure 5.13 Run the program, Menus.java

က ဒါဆိုရင်တော့ အခါ ပို့ချပေးခဲ့တဲ့ JMenuBar နဲ့ ပတ်သက်တာအပြီးမှာ JAVA နဲ့ Swing Components တစ်ခုဖြစ်တဲ့ Tabbed အကြောင်းကို ဆက်လက် လေ့လာကြရအောင်လား။ 5.4 Using the Tabbed of JAVA ဆိုပြီးမှ ရေးသား ပို့ချပေးထားပါတယ်။ လေ့လာကြည့်ပါ။



# JAVA DEVELOPER GUIDE

## Chapter (5)

### 5.4 USING THE TABBED PANES

Tabbed Panes ဆိတ်ကတော့ panels တွေကို စစည်းထားတာပါပဲ။ ဒါပေမယ့် တစ်ကြိမ်မှာ panel တစ်ခုတို့သာလျှင် ဖြင့်ရအောင် ဖန်တီးထားတာပါပဲ။ Tabbed Panes တွေကို ရယူချင်ရင် JTabbed Panes Class ကို implement လုပ်ပြီးတော့ ပေါ်ယူအသုံးချင့်ပါပဲ။

ကဲ တကယ်လို့ Tabbed Panes ကို ဖန်တီးရေးသားလိုက်လို့ panel တစ်ခုချင်းစီကို ခေါ်ယူမယ်ဆိုရင်ပေါ့နော်။ စာဖတ်သူဟာ သူရဲ့ Tab တွေကို နှိပ်ပြီးတော့ Tab Panel တွေကို ပြောင်းလဲ ကြည့်ရှုလိုက်ရှုပါပဲ။ Tabs တွေကို ဖန်တီးပဲအခါမှာ Hori-zontally across the top or bottom မှာ ဖန်တီးလို့ရာသလို Verticaly along the left or right side မှာလည်း ဖန်တီးလို့ရာပါတယ်။

Tabbed Panes တွေကို ဖန်တီးတဲ့အခါမှာ အောက်ပါ အတိုင်း constructor method (3) ခုရှိပါတယ်။

- JTabbedPane() - Create a vertical tabbed pane along the top that does not scroll.
- JTabbedPane(int) - Create a tabbed pane that does not scroll and has the specified placement.
- JTabbedPane(int, int) - Create a tabbed pane with the specified placement (first argument) and scrolling policy (second argument).

Tabbed Panes တွေကို ထားတဲ့ အနေအထားကလည်း tabs တွေကို panel ပေါ်မှာ ဆက်နွယ် ဖော်ပြပေးထားတာနဲ့ အတူတူပါပဲ။ အောက်ပါ four class arguments တဲ့က တစ်ခုခု ကိုတော့ အသုံးပြုရမှာပါပဲ။

JTabbedPane.TOP, JTabbedPane.BOTTOM,  
JTabbedPane.LEFT, JTabbedPane.RIGHT

တကယ်လို့ စာဖတ်သူဟာ tabbed pane တွေကို ဖန်တီးပြီးရင် အဲဒီ component တွေကို ထည့်သွင်းနိုင်ပြီး addTab () method ကို ခေါ်ယူ အသုံးချင့်ပါပဲ။ သူရဲ့ method type က အောက်ပါအတိုင်းပါပဲ။

```
addTab(String, Component)
      ↑           ↑
1st arguments   2nd arguments
```

1<sup>st</sup> argument ကတော့ ဖန်တီးထားတဲ့ Tabbed တွေရဲ့  
Label အနေနဲ့ ဝါဝင်ရမယ့် text ပဲ ဖြစ်ပါတယ်။

2<sup>nd</sup> argument ကတော့ components အနေနဲ့ ဖန်တီးထားတဲ့ Tab တွေကို ဖန်တီးမယ့် Tab panel ပေါ်မှာ ထည့်သွင်းတာပဲ ဖြစ်ပါတယ်။ အဲဒါတွေကို ဖန်တီးအသုံးချုပဲအခါမှာတော့ JPanel Class ကို Object အနေနဲ့ ဖန်တီး ခေါ်ယူ အသုံးချုတာပါပဲ။

ကဲ အခု အောက်မှာဖော်ပြထားတဲ့ statements တွေကို လေ့လာကြည့်ပါ။ Panel အရေအတွက် (၅) ခုကို ဖန်တီးပြီးတော့ Tabbed Pane ထဲကို ထည့်သွင်းပြထားတာပဲ ဖြစ်ပါတယ်။ လေ့လာကြည့်ပါ။

```
JPanel mainsettings = new JPanel();
JPanel advancedsettings = new JPanel();
JPanel privacysettings = new JPanel();
JPanel emailsettings = new JPanel();
JPanel securitysettings = new JPanel();
JTabbedPane tabs = new TabbedPane();
tabs.addTab ("Main", mainsettings);
```



```

1 tabs.addTab ("Advanced", advancedsettings);
2 tabs.addTab ("Privacy", privacysettings);
3 tabs.addTab ("E.mail", emailsettings);
4 tabs.addTab ("Security", securitysettings);
5 အချဖော်ပြေားခဲ့တဲ့ code_line တွေအတိုင်း panels ထဲကို
6 ပေါင်းထည့်ပြီးတော့ tabbed pane ကို ဖန်တီးလိုက်မယ်ဆိုရင်ပေါ့
7 နော်။ Figure 5.14 အနေနဲ့ ဖော်ပြေားထားတဲ့ Tabs တွေ ပါဝင်တဲ့
8 JFrame Windows ဟာ လာရောက် ဖော်ပြေားလာမှာပါပဲ။

```



Figure 5.14 A tabbed pane with five tabs displayed along the top edge

ကဲ ဒီတော့ အရင်ဆုံး Listing 5.4 အနေနဲ့ program code\_line တွေကို ရေးသားပြီးတော့ လေ့လာကြည့်ကြရအောင်လား။

#### **Listing 5.4 Demonstrate the Tabbed in Java**

```

1. import java.awt.*;
2. import java.awt.event.*;
3. import javax.swing.*;
4.
5. public class Tabbed extends JFrame {
6.
7. public Tabbed() {
8.
9. super ("Demonstrate the Tabbed Program
Setting");
10. setSize (400,150);
11.
12. setDefaultCloseOperation (JFrame.EXIT_
ON_CLOSE);
13. JPanel mainsettings = new JPanel();
14. JPanel advancedsettings=new JPanel();
15. JPanel privacysettings=new JPanel();
16. JPanel emailsettings = new JPanel();
17. JPanel securitysettings=new JPanel();
18.
19. JTabbedPane tabs = new JTabbedPane
(JTabbedPane.TOP);
20.
21. tabs.addTab ("Main".mainsettings);

```

```

22. tabs.addTab ("Advanced", advanced-
settings);
23. tabs.addTab ("Privacy", privacy-
settings);
24. tabs.addTab ("E-mail", emailsettings);
25. tabs.addTab ("Security", security-
settings);
26. show ();
27. }
28.
29. public static void main(String[] argu-
ments) {
30. Tabbed t = new Tabbed ();
31. }
32. }

```

#### **Analysis**

၁။ ဒီ program ကေးမှာတော့

```

java.awt.*;
java.awt.event.*;
javax.swing.*;

```

ဆိုတဲ့ package file (3) ခုရှိ ခေါ်တင်ထားပါတယ်။ ဒီတော့ ဆက်လေ့လာကြည့်ရင် JFrame Class ကို extends လုပ်ထားတဲ့ Tabbed ဆိုတဲ့ primary class ကို ရေးသားပါတယ်။ အဲဒီ class ထဲမှာ Tabbed () ဆိုတဲ့ constructor function ကို ရေးသားထားပါတယ်။ အဲဒီ constructor function ရဲ့ လိုင်းနံပါတ် (၉) မှာမှ

```

super ("Demonstrate the Tabbed Program
Setting");

```

ဆိုပြီးတော့ JFrame\_Windows ရဲ့ title\_bar text ကို ရေးသားထားပါတယ်။ အဲဒီနောက် JFrame\_windows ရဲ့ အရွယ်အစားကို Width=400, Height=150 ဖြစ်အောင်လို့

```
setSize (400,150);
```

ဆိုပြီးမှ ရေးသားထားပါတယ်။ လိုင်းနံပါတ် (၁၂) မှာတော့ setDefaultCloseOperation (JFrame.EXIT\_ON\_CLOSE);

method ကို ခေါ်ယူ အသုံးချယားပါတယ်။ ဒီအထိကတော့ အခုအချိန်အထိ သင်ကြားခဲ့တဲ့ sample program တွေမှာနဲ့ အတူတူပါပဲ။ သိပ်ပြီး ထူးခြားတာတော့ မရှိလေးပါဘူး။

၂။ လိုင်းနံပါတ် (၁၃) မှာတော့ JPanel Class ကို Object\_Name တွေ ပေးပြီးတော့ ခေါ်ယူ အသုံးချပါတယ်။ တစ်ခု ပြောစရာရှိတာကတော့ JPanel Class ကိုပဲ object\_name



ပေးထားတာနော်။ ဒါကြောင့် လိုင်းနံပါတ် (၁၃)ကနေ လိုင်းနံပါတ် (၁၇) အထိ

```
JPanel mainsettings = new JPanel ();
JPanel advancedsettings = new JPanel ();
JPanel privacysettings = new JPanel ();
JPanel emailsettings = new JPanel ();
JPanel securitysettings = new JPanel ();
```

ဆိုပြီးတော့ ရေးသားထားတာပါပဲ။ ဒီနေရာမှာ အကျယ်ချုပြီး တော့ ရင်းဖြစ်ရာ မလိုတော့ပါဘူး။ အရှေ့က Tabbed အကြောင်းကို ရှင်းပြန်က sample code\_line တွေနဲ့အတူ ရှင်းပြခဲ့ပြီးသားပါ။ ဟုတ်ပြောနော်။ က ပြီးတော့ Java ရဲ့ Swing components ဖြစ်လဲ JTabbedPane ကို object\_name tabs လို့ အမည်ပေးပြီး ခေါ်ယူလိုက်ပါတယ်။ လိုင်းနံပါတ် (၁၉)မှာပါ။ အဲဒီနောက်မှာမှ အရှေ့က ဖန်တီးခဲ့တဲ့ string components တွေကို container ထဲကို ထည့်သွင်းပါတယ်။ ထည့်သွင်းရာမှာတော့ addTab() method ကို ခေါ်ယူ အသုံးချထားပါတယ်။ ဒါကြောင့် လိုင်းနံပါတ် (၂၁) ကနေ (၂၅) အထိ

```
tabs.addTab ("Main".mainsettings);
tabs.addTab ("Advanced", advancedsettings);
tabs.addTab ("Privacy", privacysettings);
tabs.addTab ("E-mail", emailsettings);
tabs.addTab ("Security", securitysettings);
```

ဆိုပြီးမှ ရေးသားထားပါတယ်။ ဒါ code\_line တွေနဲ့ ပတ်သက်ပြီးတော့ အရှေ့သင်ခန်းစာမှ ရှင်းပြပေးခဲ့ပြီးသားပါပဲ။ ပြီးတော့ show() method ကို ခေါ်ယူလိုက်ပါတယ်။ ဒါတော့ ဟာ constructor functions ထဲမှာ ရေးသားထားတဲ့ code\_line တွေအကြောင်းပါပဲ။

လိုင်းနံပါတ် (၂၉) မှာတော့ main() function ကို ရေးသားထားပါတယ်။ အဲဒီ main() function ထဲမှာမှ အပေါ်က ဖန်တီးခဲ့တဲ့ Tabbed ဆိုတဲ့ class ကို object\_Name t အဖြစ် ပြောင်းလဲပြီးတော့ ခေါ်ယူလိုက်တာပါပဲ။ အဲဒီတော့ အပေါ်မှာ ဖန်တီးခဲ့တဲ့ code တွေဟာ လာရောက်ပြီးတော့ အလုပ် လာလုပ်တော့တာပါပဲ။ က ဒါဆိုရင်တော့ Listing 5.4 အနေနဲ့ ရေးသားထားတဲ့ program ကို နားလည့် သဘောပေါက်သွားမှာပါ။

က ဆက်လက် လေ့လာကြည့်ကြရအောင်လား။ အပေါ်က ရှင်းပြခဲ့တဲ့ Listing 5.4 က program code\_line တွေကို

NotePad ထဲမှာ သွားပြီးတော့ code တွေ ရှိက်သွင်းပါ။ Figure 5.14 အနေနဲ့ ရေးသားဖော်ပြုပေးထားပါတယ်။

```
Tabbed.java - Notepad
File Edit Format View Help
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
public class Tabbed extends JFrame{
public Tabbed(){
    super("Demonstrate the Tabbed Program Setting");
    setSize(400,150);
    setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
    JPanel mainsettings = new JPanel();
    JPanel advancedSettings = new JPanel();
    JPanel privacySettings = new JPanel();
    JPanel emailSettings = new JPanel();
    JPanel securitySettings = new JPanel();
    JTabbedPane tabs = new JTabbedPane(JTabbedPane.TOP);
    tabs.addTab("Main", mainsettings);
    tabs.addTab("Advanced", advancedSettings);
    tabs.addTab("Privacy", privacySettings);
    tabs.addTab("E-mail", emailSettings);
    tabs.addTab("Security", securitySettings);
    setContentPane(tabs);
    show();
}
public static void main(String[] arguments){
    Tabbed t = new Tabbed();
}
```

Figure 5.14 Write the program code\_line in the Notepad, Tabbed.java

NotePad ထဲမှာ code\_line တွေ ရေးသားပြီးသွားရင် အဲဒီ file\_name ကို Tabbed.java လို့ အမည်ပေးပြီးတော့မှ C:\j2sdk1.4.1\_01\bin ထဲမှာ သွားရောက်ပြီးတော့ saveလုပ်ပါ။ ပြီးရင်တော့ Start → All programs → Accessories → Command prompt ကို ဖွင့်လိုက်ပါ။ ပြီးရင် အောက်ပါ command နဲ့ DIR ကို ပြုပြုပါ။

```
CD C:\j2sdk1.4.1_01\bin ↵
```

ဒါလိုရင်တော့ DIR ပြောင်းသွားပြီးဆိုရင်တော့ programs ကို အောက်ပါ command\_line နဲ့ compile လုပ်လိုက်ပါ။

```
javaC Tabbed.java ↵
```

compile လုပ်လိုက်လို့ Error မရှိတော့ဘူးဆိုရင်တော့ အောက်ပါ command\_line နဲ့ ဆက်လက်ပြီးတော့ Run လုပ်လိုက်ပါ။

```
java Tabbed ↵
```

က အခုရေးပြခဲ့သလိုမျိုး compile & run ကို command prompt ထဲမှာ လုပ်သလိုမျိုးကို Figure 5.15 အနေနဲ့ ရေးသားဖော်ပြုပေးထားပါတယ်။ လေ့လာကြည့်လိုက်ပါ။



```
Command Prompt - java Tabbed
C:\j2sdk1.4.1_01\bin>javac Tabbed.java
C:\j2sdk1.4.1_01\bin>java Tabbed
```

Figure 5.15 Compile &amp; Run of the Tabbed.java

က အခုလိုမျိုး Compile & Run လုပ်လိုက်ပြီဆိုပါ၏။ အောက်မှာ ဖော်ပြထားတဲ့အတိုင်း Demonstrate အနေနဲ့ ရော့ပြထားတဲ့ Tabbed.java ဆိုတဲ့ program လေးဟာ ဖော်ပြပေးနေမှာပါပဲ။ လေ့လာကြည့်ပါ။



Figure 5.16 Run the program, Tabbed.java

အခုရေးသားပို့ချခဲ့တဲ့ သင်ခန်းစာကို နားလည် သဘော ပေါက်သွားပြီဆိုရင်တော့ ဆက်လက်ပြီးတော့ 5.5 Certification Practice ဆိုတဲ့ အခန်းကဏ္ဍကို ရေးသား ပို့ချပေးထားပါတယ်။ အခု လက်ရှိ သင်ကြား ပို့ချပြီးခဲ့တဲ့ JAVA Developer Guide VOL - I အနေနဲ့ ရေးသား ပို့ချပြီးပါပြီ။ ဒီနေရာမှာ သင်ခန်းစာ အားလုံးကို ရော်ပြီးတော့ Certification Practice အနေနဲ့ ပို့ချပေးထားပါတယ်။

JAVA Developer Guide VOL - II ကို မသင်ယူခေါင်မှာ သင်ကြား လေ့လာပြီးတဲ့ အခန်းကဏ္ဍ အားလုံးကို သေသေချာချာ တတ်ဖြောက်အောင် လေ့လာ သင်ယူချေပါတယ်။

လုပ်ငန်းခွင်က programmer တွေ ကျောင်းသားနဲ့ JAVA Language ကို လေ့လာ သင်ယူနေတဲ့သူတွေ အားလုံးအတွက် အသုံးဝင်တဲ့ Chapter Index ဖြစ်မှာလို့ ယုံကြည့်ပါတယ်။

KYAW ZAYAR LAY [ IT ENGINEERING ]



ကယ်လာနေတဲ့ PC ကွန်ပျူတာများ

ကွန်ပျူတာနှင့် မီဒီယာနယ်ပယ်တွင် လူသုံးကွန်ပစ္စည်းများက သေးသည်ထက်သေးပြီး ငယ်သည်ထက်ငယ်လာကြသည်။ သို့ရာတွင် အရည်အသေးက မင်္ဂလာပါ။ ယခု OQO (www.oqo.com) မှ ထုတ်လုပ်သည့် OQO Ultra Portable PC ကို မိတ်ဆက်ပေးလိုသည်။ OQO Model01+ ကို ၂၀၀၆ ခုနှစ် နှစ်ဦးပိုင်းကတည်းက ထုတ်လုပ်ခဲ့ခြင်းဖြစ်သည်။ ယခုနှစ်ဦးတွင် OQO Ultra Portable PC model02+ သို့ အဆင့်မြှင့်ကာ လုပေသာ ဒီဇိုင်းဖြင့် ဈေးကွက်သို့ ရောက်ရှိခဲ့ပြီ။ ဒင်းကို Windows XP Tablet သို့မဟုတ် handheld PC ဟူ၍လည်း ခေါ်ဆိုနိုင်ပြီး ခရီးဆောင် ကွန်ပျူတာတစ်လုံးသမ္မတ အသုံးပြုနိုင်သည်။ အနက်ရောင် အမိန့်ကိုယ်ထည်တွင် မီးခိုးရောင်ကိုသူတ်ကို အရောင် လိုက်ဖက်လို့စွာ တည်ဆောက်ထားပြီး ၅ လက်မအချယ် LCD (800 x 480 pixel display) မြင်ကွင်းက တောက်ပူရှိသည်။ 1.5GHz VIA processor, 60GB hard drive, 1GB of DDR2 RAM တို့က PC တစ်လုံးအဖြစ် ကောင်းစွာ သုံးနိုင်သည် အနေအထားဖြစ်သည်။ Bluetooth 2.0, 802.11a/b/g WiFi, integrated EV-DO, HDMI-outport တို့လည်း ပါဝင် တပ်ဆင်ထား၍ ဆက်သွယ်မှုများအတွက် အဆင်ပြေ ပေါ်သည်။ DVD+-RW drive ကို လွယ်ကူစွာ ချိတ်ဆက် သုံးနိုင်ပြီး USB 2.0 port, audio jack, microphone, VGA out များလည်းပါဝင်သည်။ ဘက်ထရီဖြင့် ၃ နာရီအကြာ သုံးနိုင်ပြီး docking station ပါသောကြောင့် စားပွဲပေါ်တွင် desktop ကွန်ပျူတာတဲ့သို့ တင်ပြီးသုံးနိုင်သည်။ ပုံမှန် laptop တစ်လုံးသည် ၁၃x ၁၁ လက်မနှင့် အလေးချိန် ၆ ပေါင်ဝန်းကျင်ဖြစ်၍ ယခု OQO ၏ အရွယ်အစား ၅.၆ x ၃.၃ x ၁.၈ လက်မ၊ အလေးချိန် တစ်ပေါင်နှင့် ပိုင်းယုံပါက မည်မျှကွာ ခြားမည်ကို ခန့်မှန်းကြည့်နိုင်သည်။ တန်ဖိုး ၁၄၉၉၉ ဒေါ်လာဖြစ်၍ www.oqo.com/products သို့ ဆက်သွယ် စံစမ်းနိုင်ပါသည်။

