**அத்தியாயம் 2**

**Hello World**

C, C++, Python என்று எந்த கணினி மொழியாக இருந்தாலும், முதலில் ஒரு எளிமையான ‘Hello World’ program-ஐ எழுதுவதுதான் வழக்கம். அதாவது, நாம் எழுதும் Program, ‘Hello World’ என்ற ஒரே ஒரு வரியினை திரையில் காட்டும். அந்த மரபை ஒட்டி, நாமும் Java மொழியில், முதலில் ‘Hello World’ program-ஐ எழுதப் போகிறோம்.

அதற்கு, HelloWorld.java என்ற file-ஐ உருவாக்குவோம். அதில், கீழே கொடுக்கப்பட்டுள்ள code-ஐ எழுதுவோம்.

|  |
| --- |
| */\**  *\* HelloWorld.java*  *\* Program to print ‘Hello World’.*  *\*/*  *// HelloWorld class*  *public class HelloWorld {*  *public static void main(String[] args) {*  *System.out.println(“Hello World”); // This statement will print ‘Hello World’*  *}*  *}* |

இப்பொழுது, இந்த program-ஐ ‘Run’ செய்வதற்கு ‘Shift+F10’-ஐ அழுத்தவும்(அல்லது, Run -> Run ‘Hello World’-ஐ அழுத்தவும்). இப்பொழுது, உங்கள் திரையில், ஒரு புது window தோன்றி, அதில் ‘Hello World’ என்ற வாக்கியம் தோன்றும்.

வாழ்த்துக்கள்!!!! நீங்கள் Java மொழியில் உங்கள் முதல் program-ஐ எழுதி, அதை வெற்றிகரமாக execute (அல்லது Run) செய்து விட்டீர்கள். இப்பொழுது, நீங்கள் என்ன எழுதி இருக்கிறீர்கள் என்று பார்ப்போம்.

# **Comments**

முதலில், இந்த வாக்கியங்களை பார்ப்போம்.

*/\**

*\* HelloWorld.java*

*\* Program to print ‘Hello World’.*

*\*/*

இந்த வரிகளை, ‘Comments’ என்று சொல்லுவோம். இவை, உங்கள் program பற்றி நீங்கள் எழுதும் விளக்கங்கள். இவற்றில், நீங்கள் எழுதும் விஷயங்கள் உங்கள் program-ஐ எந்த விதத்திலும் பாதிக்காது. இவற்றை எழுதாமல் விட்டாலும், உங்கள் program எந்த விதத்திலும் பாதிக்கப்படாது. பின் இவை எதற்கு?

இந்த program, மிகவும் எளிமையானது. எப்பொழுது இதை பார்த்தாலும், இந்த program என்ன செய்கிறது என்று உடனே புரிந்து கொள்ள இயலும். ஆனால், பின்னாட்களில் நீங்கள் மிகவும் பெரிய, மிகவும் சிக்கலான program-களை எழுத வேண்டி வரும். அப்பொழுது, நீங்கள் எழுதும் program-ஐ மற்றவர்களோ அல்லது எழுதி சில நாட்கள் கழித்து நீங்களே பார்த்தால் கூட உடனே புரிய வேண்டும். அதற்கு, இந்த ‘Comments’ உதவும். உங்கள் program என்ன செய்கிறது என்று இந்த comments-இல் நீங்கள் விரிவாக எழுதலாம்.

## **Multi-Line comments**

Java-வில் மூன்று வகையான comments-கள் உண்டு. முதல் வகை:

/\*

......

......

\*/

அதாவது, ‘/\*’ என்பதில் ஆரம்பித்து ‘\*/’ என்பதில் முடியும். இந்த இரண்டு குறீயிடுகளுக்கு (symbols) நடுவில், நீங்கள் எழுதும் அத்தனையும் comments-தான். அது உங்கள் program-ஐ எந்த வகையிலும் பாதிக்காது. மேலும், இந்த வகையான comments, பல வரிகளுக்கு எழுதலாம். இவற்றை, ‘Multi-Line’ comments என்று சொல்வோம்.

### **உதாரணங்கள்**

1. /\* This is a sample program to print ‘Hello World’

This is written by a beginner.

\*/

மூன்று வரிகளில் எழுதப் பட்ட சரியான comment. ‘/\*’-இல் ஆரம்பித்து, ‘\*/’-இல் முடிகின்றது.

அதற்கு நடுவில் உள்ள அனைத்தையும், உங்கள் Java compiler புறக்கணித்து விடும்.

1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* This is a sample program to print ‘Hello World’ \*

\* This is written by a beginner. \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

நான்கு வரிகளில் எழுதப் பட்ட சரியான comment. இதற்கும், மேலே (1)-இல் பார்த்த comment-க்கும் அர்த்தத்தில் எந்த வித்தியாசமும் இல்லை. உங்கள் Java compiler-ஐ பொறுத்த அளவில், ‘/\*’ -இல் ஆரம்பித்து. ‘\*/’ வரை உள்ள அனைத்தையும் புறக்கணித்து விடும். நடுவில் உள்ள, பல ‘\*’-கள், உங்கள் program-ஐ அழகு படுத்த பயன்படும். அவ்வளவுதான்.

1. /\* This is a sample program to print ‘Hello World. This is written by a beginner \*/

ஒரே ஒரு வரியில் எழுதப்பட்டாலும், இது சரியான comment.

1. /\* This is a sample program to explain ‘Multi line’ comments that

start with ‘/\*’ and ends with ‘\*/’.

This is written by a beginner

\*/

இது ஒரு தவறான(invalid) comment. ஏன்? நன்றாக கவனித்து பாருங்கள். உங்கள் Java compiler, ‘/\*’ மற்றும் ‘\*/’ என்ற symbol-களை மட்டும் பார்த்து, அதற்கு நடுவில் இருப்பவற்றை புறக்கணிக்கும். அதன்படி, முதலில் ‘/\*’ – இல் ஆரம்பித்து, இரண்டாம் வரியில் உள்ள ‘\*/’ symbol-இல் முடித்து விடும். அதன் பிறகு உள்ள வரிகளை:

’.

This is written by a beginner

\*/

சரியான ‘statements’-களாக எண்ணி process செய்ய முயற்சி செய்து, error கொடுக்கும்.

## **Single-Line comments**

Java-வின் இரண்டாவது வகையான comments:

//.........

அதாவது, ‘//’ என்ற symbol-க்கு பிறகு – அந்த வரியில் வரும் அத்தனையும் comments-தான். இந்த வகையான comments, ஒரே ஒரு வரியில் மட்டுமே எழுத வேண்டும். இவற்றை, ‘Single-Line’ comments என்று சொல்வோம்.

### **உதாரணங்கள்**

1. // Hello World class

இது ஒரு சரியான Single Line comment.

1. System.out.println(“Hello World”);//Line to print Hello World

இதில், *System.out.println(“Hello World”);* என்பது வரை சரியான statement. அதற்கு, பிறகு ‘//’-இல் ஆரம்பித்து அந்த வரியின் இறுதி வரை, இது ஒரு சரியான comment.

## **JavaDoc comments**

மேலே விளக்கி உள்ள Multi Line comments மற்றும் Single Line comments-கள் அல்லாமல், Java-வில் மூன்றாவது வகையான comment உள்ளது. அது JavaDoc comments. அதைப் பற்றி பின் வரும் அத்தியாயங்களில் தெரிந்து கொள்வோம்.

# **Class**

அடுத்துள்ள வரிகளை காண்போம்:

*public class HelloWorld {*

*public static void main(String[] args) {*

*System.out.println(“Hello World”); // This statement will print ‘Hello World’*

*}*

*}*

இதில் class என்பதற்கான விளக்கத்தையும், main என்பதற்கான விளக்கத்தையும் பின்னால் பார்ப்போம். தற்பொழுது உங்களுக்கு தெரிய வேண்டியது:

1. ஒரு program file-ல் ஒரு public class மட்டும்தான் இருக்க வேண்டும். அதனுடைய பெயரும், அந்த program file-இன் பெயரும் public class-இன் பெயரும் ஒன்றாக இருக்க வேண்டும். உதாரணத்திற்கு, மேலே உள்ள எடுத்துக்காட்டில், file-இன் பெயர் HelloWorld.java. அதில் உள்ள public class-இன் பெயர் HelloWorld.
2. உங்கள் program-ஐ நீங்க execute செய்யும் பொழுது, JVM உங்கள் file-ல் உள்ள public class-ஐ தேடி, அதில் உள்ள main method-ஐ தேடி, அதில் உள்ள statements-ஐ execute செய்யும்.

அதாவது, பொதுவாக உங்கள் Java program-இன் அமைப்பு, இப்படித்தான் இருக்கும்:

|  |
| --- |
| public class <உங்கள் class பெயர்> {  public static void main(String[] args) {  <execute செய்ய வேண்டிய உங்கள் statement-கள்>  }  } |

அதாவது, மேலே உள்ள எடுத்துக்காட்டில், உங்கள் program-இன் execution, “*System.out.println(“Hello World”); // This statement will print ‘Hello World’”* என்ற வரியில் இருந்ததுதான் ஆரம்பிக்கும். இந்த வரி என்ன செய்கிறது என்று பார்ப்போம்.

முதல் அத்தியாயத்தில் பார்த்தது போல, JDK உங்களுக்கு உதவி செய்வதற்காக பல்வேறு classes மற்றும் methods-ஐ உருவாக்கி கொடுத்துள்ளது. அதில் ஒன்றுதான் System.out.println method. இந்த method, அதற்கு நீங்கள் கொடுக்கும் value-வை திரையில் அச்சிடும். மேலே உள்ள எடுத்துக்காட்டில், System.out.println method-க்கு Hello World என்ற value-வை அனுப்புகிறோம். அதனால், JVM, ‘Hello World’ என்ற value-வை திரையில் அச்சிடும். இதுவே, ‘Hello World’-க்கு பதிலாக, வேறு ஏதாவது ஒரு value-வை அனுப்பினால், அந்த JVM அந்த value-வை திரையில் அச்சிடும்.

அடுத்துள்ள “*// This statement will print ‘Hello World’”* என்ற பகுதி வெறும் comment. JVM இந்த பகுதியை புறக்கணித்து விடும்.

இந்த statement-க்கு பிறகு, execute செய்ய வேறு எந்த statements-ம் இல்லாததால், உங்கள் program-இன் execution முடிந்து விடும்.

# **சுய மதிப்பீட்டு பயிற்சிகள்**

1. திரையில் ‘This is my first program.’ என்ற value-வை அச்சிட program எழுதவும்.
2. திரையில் ‘Java is a programming language’ என்ற value-வை அச்சிட program எழுதவும்.
3. திரையில் ‘1 + 1 = 2’ என்ற value-வை அச்சிட program எழுதவும்.
4. திரையில் ‘Welcome to the World of Java!!!’ என்ற value-வை அச்சிட program எழுதவும்.

மேலே உள்ள அனைத்து பயிற்சிகளையும் தனித் தனி program-களாக தனித் தனி file-களில் எழுதி பழகுங்கள்.