CHAPTER 4: INTRODUCTION TO JAVA CODING

5.1 ESSENTIALS

In order to start writing code in java, it is essential to understand the meaning and use of certain keywords that are the building blocks of any java code.

Keywords	Meaning	Example Code

public	An access modifier that allows classes, attributes, methods and constructors to be accessible by other classes.	public class Example1 {
		public static void main(String args[]) {
		System.out.println("Hello World.");
		}
		}

class	This keyword defines a class.	
main	This defines the main method.	
args	It represents arguments passed in the java command-line.	
static	It is used for memory management. It is used to share properties of a class with other classes.	
void	It is used during method declaration to indicate that a method does not return any type.	
String[] args	It represents an array of strings that stores the arguments passed in the java command-line.	

5.2 SCOPES

In java, a block of code is the code inside the curly braces, {}. The scope of a variable represents the space or block of code in which the variable can be accessed and modified. Usually, the scope of a variable is the block of code in which it is created, for example- (1) A variable declared inside the curly braces of a loop is only valid (can be accessed and modified) within the scope of the loop, (2) A variable declared inside the curly braces of a method is only valid within the scope of the method.

Example Code	Output

public class Example1 {
public static void main(String args[]) {
int x=10;
int y=25;
int z=x+y;
while (x>5){
int m=3; //m initialized inside loop.
int n=m+x; //n initialized inside loop.
x--;
}
 	
System.out.println("Sum of x+y = " + z);
System.out.println(m+" "+n); //m & n not accessible outside loop.
}
}
	error: cannot find symbol

public class Example2 {	3 13

public static void main(String args[]) {
int x=10;
int y=25;
int z=x+y;
while (x>5){
int m=3; //m initialized inside loop.
int n=m+x; //n initialized inside loop.
System.out.println(m+" "+n); //m & n accessible only inside loop.
x--;
}
 	
System.out.println("Sum of x+y = " + z);
}
}
	3 12
	3 11
	3 10
	3 9
	Sum of x+y = 35

public class Example3 {
public static void main(String args[]) {
int x=10;
int y=25;
int z=x+y;
while (x>5){
int m=3; //m initialized inside loop.
int n=m+x; //n initialized inside loop.
x--;
}
 	
System.out.println("Sum of x+y = " + z);
m+=5; //m cannot be modified outside loop.
System.out.println(m+" "+n); //m & n not accessible outside loop.
}
}
	error: cannot find symbol

public class Example4 {
public static void main(String args[]) {
int x=10;
int y=25;
int z=x+y;
while (x>5){
int m=3; //m initialized inside loop.
int n=m+x; //n initialized inside loop.
m+=5; //m can only be modified inside loop.
System.out.println(m+" "+n); //m & n accessible only inside loop.
x--;
}
 	
System.out.println("Sum of x+y = " + z);
}
}
	8 13
	8 12
	8 11
	8 10
	8 9
	Sum of x+y = 35

5.3 Printing

In order to show an output of a program, we use the print statement, **System.out.println("Text to be printed.").** The **System.out** command basically invokes the System class to generate an output. The **println()** is a built-in public method used to show an output. It prints the text inside the method parameter, as well as a new line. To avoid adding the new line at the end, **print()** can be used instead as **System.out.print()**.

Example Code	Output
public class Example1 {	10

public static void main(String args[]) {
int x=10;
while (x>5){
System.out.println(x); //New line is added at the end.
x--;
}
}
}
	9
	8
	7
	6

public class Example1 {
public static void main(String args[]) {
int x=10;
while (x>5){
System.out.print(x); //No new line is added at the end.
x--;
}
}
}
	109876

5.4 Comments

In a program, comments are lines which are not executed but are there to help someone who is not familiar with the code to understand how the code works. The compiler ignores these lines while compiling. Writing comments can also help to debug code easily.

In java, comments are 3 types:

1. **Single-line comments:** These are comments written in a single line and are usually short. The comment is initiated using two forward slashes, **//.** Anything written after the second slash in the same line will be considered as a comment.
2. **Multi-line comments:** These are comments written when descriptions need to be written for methods or loops in multiple lines. Writing multi-line comments using // is possible, but it becomes tedious when there are several lines, and // needs to be written before each line. In order to make writing multi-line comments easier, use a forward slash, /, followed by an asterisk, *, then write the comment, and end using an asterisk, and a forward slash.
3. **Documentation comments:** It is often referred to as doc comment. It is usually used when writing code for a software or package to generate a documentation page for reference. It follows the same convention as a multi-line comment, only an exception of an additional asterisk at the beginning of every new line.

Comment type	Example
Single line	//This is a single-line comment.
Multi–line	/*Multi-line comment starts This is a multi-line comment. Multi-line comment ends.*/
Documentation	/** Doc comment starts *continue *add tags *doc comment ends*/

5.5 Worksheet

A. Take two integer numbers from the user and print the summation and average.

B. Take the values of x, y and z and solve the result of the following equation:
result= x+(x%2+y*z^ (4/2))

C. Take the first name and last name from the user and print the outputs accordingly.

Sample Input #1: Samiha Haque	Output: Full Name: Samiha Haque
Sample Input #2: Sabbir Ahmed	Output: Full Name: Sabbir Ahmed

D. Take the first name, middle name and last name from the user and print the outputs accordingly. Keep in mind that a user may or may not have a middle name. If the user does not have one, they must enter an empty string as the middle name.

Sample Input #1: Aiman Jabeen Ramisa	Output: Full Name: Aiman Jabeen Ramisa
Sample Input #2: Dibyo Fabian Dofadar	Output: Full Name: Dibyo Fabian Dofadar
Sample Input #3: Sifat Tanvir	Output: Full Name: Sifat Tanvir