PROGRAM STRUCTURE

CSC 111

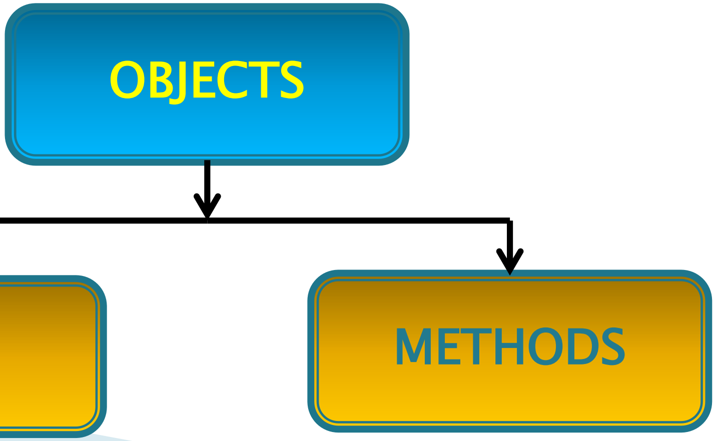
Outline

1. Programming Approaches
2. Object-Oriented Programming
3. Program Structure
4. Output Statements

1. PROGRAMMING APPROACHES

Two programming approaches are known:

- The Structured Programming
  - Also known as modular programming.
  - The problem is divided into smaller sub-problems (modules).
  - Each sub-problem is then analyzed and solved.
  - The solutions of all sub-problems are then combined to solve the overall problem.
  - Examples of such programming model languages include Pascal, Fortran and C.
- The Object-Oriented Programming
  - Identify the components of the problem. These are called objects.
  - For each object, identify the relevant data & operations (methods) to be performed on that data.
  - Define the relationship between each object and the other.
  - Examples of programming languages that follow such model are C++ and Java.


OBJECTS

DATA

METHODS

2. OBJECT-ORIENTED PROGRAMMING

EXAMPLE 1

A program is needed by a local store that rents videos to customers. Identify the objects of such program. Also, specify the data and methods for each identified object.

- Objects:
  - Video
  - Customer

- Video
  - Data
    - Movie name
    - Starring actors
    - Production company
    - Production date
    - Number of copies in the store
  - Methods
    - Reduce number of copies
    - Add number of copies

- Customer
  - Data
    - Customer name
    - Customer ID
    - VIP
  - Methods
    - Add a customer
    - Remove a customer

2. OBJECT-ORIENTED PROGRAMMING

EXAMPLE 2

A program is needed by a university Registrar. Identify the objects of such program. Also, specify a few data and methods for each identified object.

- Objects:
  - Student
  - Course

- Student
  - Data
    - Student name
    - Student ID
    - GPA
  - Methods
    - Add a student
    - Remove a student
    - Print a student's schedule
    - Update a student's GPA

- Course
  - Data
    - Course code
    - Course name
    - Pre-requisites
    - Co-requisites
  - Methods
    - Add a course
    - Remove a course
    - Define schedule

3. PROGRAM STRUCTURE

PROGRAM LAYOUT

```
// import necessary libraries
public class ProgramLayout
{
  public static void main (String[] args)
    {
      // Declaration section: to declare needed variables
      // Input section: to enter values of used variables
      // Processing section: processing statements
      // Output section: display program output
    } // end main
} // end class
```

4. OUTPUT STATEMENTS

PROGRAM 1

- The following is a simple Java program:

```
// This is my first Java program
public class Welcome
{
   public static void main (String[] args)
      {
         System.out.println (“Welcome to Java”);
      }
}
```

- When you compile and execute this program, you will get the following output:

```
Welcome to Java
```

4. OUTPUT STATEMENTS

PROGRAM 1 – EXPLAINED

```
1  // This is my first Java program
2  public class Welcome
3  {
```

Line	Symbol/Word	Explanation
1	//	Precedes a comment. A comment is discarded by the compiler. The comment ends with the end of the line. Comments are written to document (clarify) the program.
2	public	Reserved word to Java. To be explained later.
2	class	Reserved word to Java. Any Java program must start with a class. A class includes one or more methods.
2	Welcome	Class name. When a Java program is saved, it is usually given the class name. The system automatically gives the extension ".java". So, in this example, the name of the file on the disk is "welcome.java".

4. OUTPUT STATEMENTS

PROGRAM 1 – EXPLAINED (cnt'd)

```
1  // This is my first Java program
2  public class Welcome
3  {
4     public static void main (String[] args)
5       {
6          System.out.println("Welcome to Java");
```

Line	Symbol/Word	Explanation
3	{	Opening brace to the class Welcome. Any opening brace must be closed in the program. This brace is closed in line 8.
4	public static void String[] args	Reserved words to Java. To be explained later.
4	main	This is a non-optional name to the "main" method. A Java class should have at most one "main" method. Program execution thread always begins with the "main" method.
5	{	Opening brace to the "main" method. This brace closes at line 7.

4. OUTPUT STATEMENTS

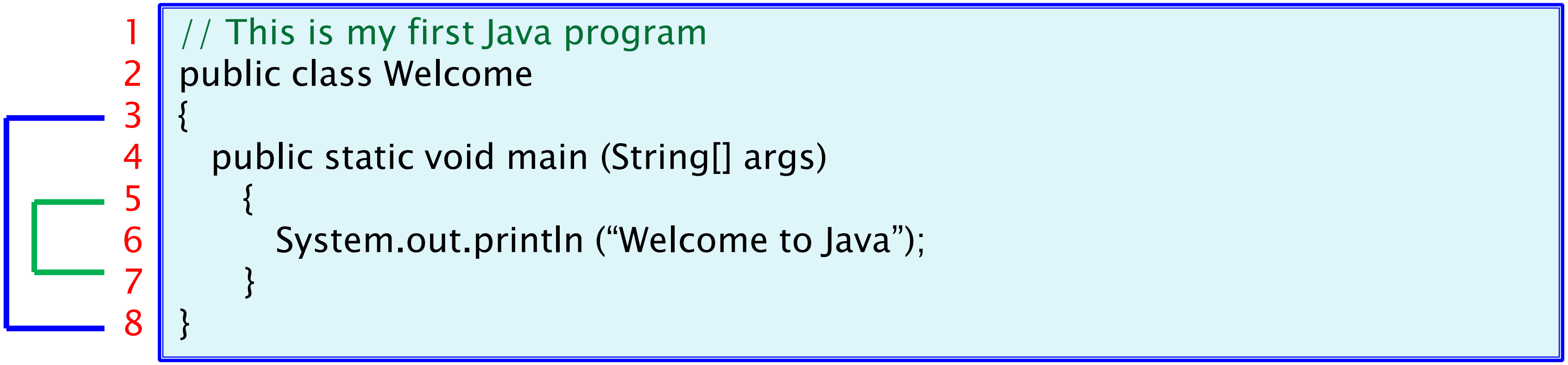
PROGRAM 1 – EXPLAINED (cnt'd)

```
1 // This is my first Java program
2 public class Welcome
3 {
4    public static void main (String[] args)
5       {
6          System.out.println ("Welcome to Java");
7       }
8 }
```

Line	Symbol/Word	Explanation
6	System.out.println (“Welcome to Java”)	This is an output statement. The words enclosed between double quotes are called a string. “Welcome to Java” is a string. The string is displayed on the screen as it appears exactly in the program. The double quotes are not printed. (Refer to slide 6 to see the output).
6	;	The semicolon ends the output statement.
7	}	The closing brace to the “main” method.
8	}	The closing brace to the “Welcome” class.

4. OUTPUT STATEMENTS

PROGRAM 1 – EXPLAINED (cnt'd)



```
// This is my first Java program
public class Welcome
{
   public static void main (String[] args)
      {
         System.out.println (“Welcome to Java”);
      }
}
```

In Java programs, braces are nested. In other words, the brace that opens first, closes last. (Refer to the figure)

4. OUTPUT STATEMENTS

PROGRAM 2

```
/* Author name: XYZ
    This is another simple Java program
*/
public class SimpleProgram
{
   public static void main (String[] args)
      {
         System.out.println (“The sum of 2 and 3 is ” + 5);
         System.out.println (“7 + 8 = “ + (7 + 8) );
      }
}
```

- The program output is as follows:

```
The sum of 2 and 3 is 5
7 + 8 = 15
```

4. OUTPUT STATEMENTS

PROGRAM 2 – EXPLAINED

```
1  /* Author name: XYZ
2     This is another simple Java program
3  */
4  public class SimpleProgram
5  {
6     public static void main (String[] args)
7        {
8           System.out.println ("The sum of 2 and 3 is " + 5);
```

Line	Symbol/Word	Explanation
1	/*	Starts a multi-line comment.
3	*/	Ends a multi-line comment.
4	SimpleProgram	Class name.
8	"The sum of 2 and 3 is"	This string is printed as it is written in the program like the previous example.
8	+	Strings concatenation operator. It joins the string with 5. The output is shown in the previous slide.
8	5	The system automatically converts the number 5 into a string to concatenate it with the first string.

4. OUTPUT STATEMENTS

PROGRAM 2 – EXPLAINED (cnt'd)

```
1  /* Author name: XYZ
2      This is another simple Java program
3  */
4  public class SimpleProgram
5  {
6     public static void main (String[] args)
7       {
8          System.out.println ("The sum of 2 and 3 is " + 5);
9          System.out.println ("7 + 8 = " + (7 + 8) );
```

Line	Symbol/Word	Explanation
9	"7 + 8 = "	This string will print as is.
9	(7 + 8)	The parenthesis instruct the compiler to add the two numbers resulting in 15.
9	+	Concatenation operator. Joins the first string with the number 15. The number 15 is converted to a string before being joined to the first string.

The effect of the + operator is determined according to the surrounding operands. If it is between strings, it "concatenates"; if it is between numbers, it "adds".

Self-Check Exercises (1)

- A program is needed for a local store that rents cars. Identify the objects in this program. Also, specify a few data and methods of each identified object.
- A program is needed for a hospital. Identify objects in this program. Also, specify a few data and methods of each identified object.
- What is the output of the following program:

```
// This is an exercise
public class SimpleProgram
{
   public static void main (String[] args)
      {
         System.out.println (“5 multiplied by 8 is ” + 40);
         System.out.println (“5 * 8 = “ + (5 * 8) );
      }
}
```

Self-Check Exercises (2)

- Consider the following program and answer the questions below:

```
// This is an exercise
public class SimpleProgram
{
   public static void main (String[] args)
      {
         System.out.println (“5 multiplied by 10 is ” + 50);
         System.out.println (“5 * 10 = “ + “(5 * 10)” );
      }
}
```

  - What is the class name in this program?
  - How many methods are there in this class? What are their names?
  - Identify the comments
  - Identify the strings

Self-Check Exercises (3)

- Detect the errors in the program below: (8 errors)

```
/* This is an exercise
public IncorrectProgram
{
   public void Main (String args)
      {
         System.out.println (“I study Java ” + “one”)
         System.out.println (“3 + 7 =  + (7 * 3) );
}
```