

Essentials of Information Technology

PC-CS-305

Controlling Program Flow – Selection Control

Topic & Structure of the lesson



- Controlling Program Flow
 - Selection Control
 - if .. else construct
 - nested if.. else constructs
 - switch...case construct
 - Iteration Control
 - Counter controlled
 - Sentinel controlled

gauravgambhir.cse@piet.co.in

Learning outcomes



- At the end of this lecture you should be able to:
 - Create, edit, compile and run simple Java applications in adherence to syntax and professional programming practices

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

Key terms you must be able to use



- If you have mastered this topic, you should be able to use the following terms correctly in your assignments and exams:
 - control structure
 - switch ... case construct
 - iteration
 - looping

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

A simple Java program



```
//Welcome1.java
//First Java Program
public class Welcome1
{
    public static void main(String[] args)
    {
        System.out.println("Welcome to Java Programming!");
    }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

A simple Java program



- Save the file with the same name as the class name with a .java extension.
- Compile and run the program

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



A simple Java program

- Comments – to document programs and improve program readability
 - single-line comment
 - multiple line comment
 - delimiter
 - documentation comment
- White spaces – for readability
 - blank lines
 - space characters
 - tab characters

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



A simple Java program

- ```
public class Welcome1
{
 • class definition Welcome1
 – every program in Java has at least one class
 definition defined by the programmer - programmer-
 defined classes / user-defined classes
 – class
 • keyword / reserved word
 • lowercase letters
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



## A simple Java program

- class name
  - capital first letter for every word in class name
  - identifier
    - series of characters consisting of letters, digits, underscores and dollar signs that does not begin with a digit and does not contain any spaces
    - case sensitive - uppercase and lowercase letters are different

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



## A simple Java program

public static void main (String[ ] args)

- a part of every Java application
- main with parenthesis – indicates it is a building block called **method**
  - Java class definition contains one or more methods
  - exactly one of those methods must be called **main** and **defined**
- **void** – the method will perform a task but will not return any information when it completes its task

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## A simple Java program



```
System.out.println("Welcome to Java Programming!");
```

- a statement
- instructs computer to perform an action
  - print a string of characters / character string /message / string literal
  - white spaces within "" not ignored

**System.out.**

- Standard output object

**System.out.println**

- displays / prints a line of text in the command window

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample Program



```
class CalculateValues
{
 public static void main(String[] args)
 {
 System.out.println("Program to calculate values!");

 int x=7, y=10;
 int d=0, s=0, q=0;

 d=y-x;
 s=x+y;
 q=y/x;

 System.out.println("The product is " +x*y);
 System.out.println("The difference is " +d);
 System.out.println("The sum is " +s);
 System.out.println("The quotient is " +q);
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – if (null else)



```
if (boolean expression)
 statement1

class FirstIf
{
 public static void main(String[] args)
 {
 System.out.println("Simple IFs");

 int x=5, y=10;

 if(x<y)
 System.out.println("X is less than Y");
 }
}
```

```
if (boolean expression)
{
 statements
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – if ... else



```
if (boolean expression)
 statement1
else
 statement2
```

```
if (boolean expression)
{
 statements
}
else
{
 statements
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program



```
class SecondIf
{
 public static void main(String[] args)
 {
 System.out.println("Simple IFs");

 int x=5, y=10;

 if(x>y)
 System.out.println("X is greater than Y");
 else
 System.out.println("X is less than Y");
 }
} ----- Run -----
Simple IFs
X is less than Y

Output completed (0 sec consumed) - Normal Termination
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – nested if



```
if (boolean expression 1)
 statement1
else
 if (boolean expression 2)
 statement2
 else
 statement3
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Example



```
if(score>=90)
 System.out.println("The grade is A");
else
 if(score>=80)
 System.out.println("The grade is B");
 else
 if(score>=70)
 System.out.println("The grade is C");
 else
 if(score>=60)
 System.out.println("The grade is D");
 else
 System.out.println("The grade is F");
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – nested if



```
if (boolean expression 1)
 if (boolean expression 2)
 statement1
 else
 statement2
else
 statement3
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Example



```
if (temperature >=50)
 if (temperature >=80)
 System.out.println("Good day for swimming");
 else
 System.out.println("Good day for golfing");
else
 System.out.println("Good day to play tennis");
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – compound if



```
if (boolean expression1 and boolean expression2)
 statement1
else
 statement2
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Example



```
if (temperature >=50) && (temperature >=80)
 System.out.println("Good day for swimming and golfing");
else
 System.out.println("Good day to play tennis");
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – compound if



```
if (boolean expression1 or boolean expression2)
 statement1
else
 statement2
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Example



```
if (letter =='y') || (letter == 'Y')
 System.out.println("The letter you have chosen is Y");
else
 System.out.println("The letter you have chosen is not a Y");
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Conditional Constructs – switch



```
switch (expression)
{ case constant-1: statements-1 break;
 case constant-2: statements-2 break;
 . . // (more cases) .
 case constant-N: statements-N break;
 default: // optional default case statements-(N+1)
} // end of switch statement
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Example



```
class Switch
{
 public static void main(String[] args)
 {
 char movie_id='A';
 switch (movie_id)
 {
 case 'A' : System.out.println("Armageddon"); break;
 case 'S' : System.out.println("Star Wars"); break;
 case 'C' : System.out.println("Casino Royale"); break;
 default : System.out.println("Sorry these are the only ones listed");
 break;
 }
 }
}
----- Run -----
Armageddon
Output completed (0 sec consumed) - Normal Termination
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Iteration control : Counter-controlled



- while loop construct
  - entry-controlled loop statement
  - makes a test condition before the loop is executed
- do ... while construct
  - used when need to execute the body of the loop before the test is performed
  - exit controlled loop
- for loop
  - entry-controlled loop statement

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program – while



```
class WhileLoop
{
 public static void main(String[] args)
 {
 System.out.println("While Loop");
 int x=5; // initialize the loop control variable
 while (x<=10) // test the loop control variable
 {
 System.out.println("Value of x is " +x);
 x++; // update the loop control variable
 }
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample Program-While



```
class WhileDemo {
 public static void main(String[] args){
 int count = 1;
 while (count < 11) {
 System.out.println("Count is: " + count);
 count++;
 }
 }
}
```

Count is: 1  
Count is: 2  
Count is: 3  
Count is: 4  
Count is: 5  
Count is: 6  
Count is: 7  
Count is: 8  
Count is: 9  
Count is: 10

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program – do...while



```
class DoWhileDemo {
 public static void main(String[] args){
 int count = 1;
 do {
 System.out.println("Count is: " + count);
 count++;
 } while (count < 11);
 }
}
```

```
Count is: 1
Count is: 2
Count is: 3
Count is: 4
Count is: 5
Count is: 6
Count is: 7
Count is: 8
Count is: 9
Count is: 10
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program – do...while



```
class DoLoop
{ public static void main(String[] args)
{
 System.out.println("Do Loop");

 int i=1, sum=0;

 do // statements in the loop is executed at least once
 {
 System.out.println("\ni=" + i);
 System.out.println("Value of sum is" +sum);
 sum = sum + i;
 i = i + 1;
 }
 while (sum<10); // test the loop control variable
}
```

```
Do Loop
i=1
Value of sum is 0
i=2
Value of sum is 1
i=3
Value of sum is 3
i=4
Value of sum is 6
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program – for



```
class ForLoop
{
 public static void main(String[] args)
 {
 System.out.println("For Loop");
 int x;
 // initialization, testing and updating the loop control variable is within
 // the for loop
 for (x=1; x<=10; x++)
 {
 System.out.println("Value of x is " +x);
 }
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program - Enhanced for loop



```
public class Enhancedforloop {

 public static void main(String args[]){
 int [] numbers = {10, 20, 30, 40, 50};

 for(int x : numbers){
 System.out.print(x);
 System.out.print(",");
 }
 System.out.print("\n");
 String [] names ={"James", "Larry", "Tom", "Lacy"};
 for(String name : names) {
 System.out.print(name);
 System.out.print(",");
 }
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Iteration control : Sentinel-controlled



- while loop construct

– the statements within the while loop need to be executed until a special (sentinel) value is met

**input the first data item into the variable //initialize the loop**

**//control variable**

**while (variable != sentinel) //test the loop control variable**

**{**

**input the first data item into the variable //update the loop**

**//control variable**

**.**

**.**

**}**

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample Program - SentinelWhile



```
import java.util.Scanner;
class SentinelWhile
{
 public static void main(String[] args)
 {
 int sentinel=19;
 System.out.println("Please enter a number between 1-20");
 Scanner s = new Scanner(System.in);
 int number=s.nextInt();
 while (number != sentinel)
 {
 System.out.println("Your guess is wrong retry");
 number=s.nextInt();
 }
 System.out.println("Your guess is right");
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample Program - SentineldoWhile



```
import java.util.Scanner;
class SentineldoWhile
{
 public static void main(String[] args)
 {
 int sentinel=19;
 System.out.println("Please enter a number between 1-20");
 Scanner s = new Scanner(System.in);
 int number=s.nextInt();
 do
 {
 System.out.println("Your guess is wrong retry");
 number=s.nextInt();
 }while (number != sentinel) ;
 System.out.println("Your guess is right");
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Iteration control : Flag-controlled



- uses a boolean variable to control the loop
- takes the following form:

```
found = false; //initialize the loop control variable
while (!found) //test the loop control variable
{
 .
 .
 .
 if (expression)
 found = true; //update the loop control variable
 .
 .
 .
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program



```
// Flag-controlled while loop
import java.io.*;
import java.util.*;
public class FlagControlledLoop1
{
 public static void main (String[] args)
 {
 int num;
 int guess;
 boolean done;

 Scanner s=new Scanner(System.in);
 num = (int)(Math.random() * 100);
 done = false;

 while (!done)
 { System.out.print("Enter an integer greater than or equal to 0 and less than 100 :");
 guess=s.nextInt();
 System.out.println();

 if (guess == num)
 { System.out.println("You guessed the correct number");
 done = true; }
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample program



```
else
{ if (guess < num)
 { System.out.println("Your guess is lower than the number");
 System.out.println("Guess again");
 }
 else
 { System.out.println("Your guess is higher than the number");
 System.out.println("Guess again");
 }
}
} // end while
} //end main method
} //end class
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## break



- enables early exit from a loop
  - when break is encountered inside a loop, loop is immediately exited and program continues with the statement immediately following the loop
- causes loop to be terminated
- can be used within a while, do or for loops

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Sample Program - break



```
class WhileBreak
{
 public static void main(String[] args)
 {
 System.out.println("While Loop");
 int x=5;
 while (x<=10)
 {
 System.out.println("Value of x is " +x);
 if (x==8)
 {
 break;
 }
 x++;
 }
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Type Conversion and Casting



- It is fairly common to assign a value of one type to a variable of another type.
- If the two types are compatible, then Java will perform the conversion automatically.
- For example, it is always possible to assign an **int value to a long variable**.
- However, not all types are compatible, and thus, not all type conversions are implicitly allowed.

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

## Type Conversion and Casting



- For instance, there is no automatic conversion defined from **double to byte**.
- It is still possible to obtain a conversion between incompatible types.
- An *automatic type conversion* will take place if the following two conditions are met:
  - The two types are compatible.
  - The destination type is larger than the source type.

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

PC-CS-305: Essentials of Information Technology

## Type Conversion and Casting



- When these two conditions are met, a *widening conversion takes place*.
- However, there are no automatic conversions from the numeric types to **char or boolean**. **Also, char and boolean are not compatible with each other.**

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Casting



- If you want to assign an **int value to a byte variable?** **This conversion will not** be performed automatically, because a **byte is smaller than an int**.
- **This kind of conversion** is sometimes called a *narrowing conversion, since you are explicitly making the value narrower so that it will fit into the target type.*

Example : Conversion.java

- int a;
- byte b;
- // ...
- b = (byte) a;

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

# Arrays



- An *array* is a group of like-typed variables that are referred to by a common name.
- Arrays of any type can be created and may have one or more dimensions.
- A specific element in an array is accessed by its index.
- Arrays offer a convenient means of grouping related information.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Declaring and instantiating an array object



- General form to declare a one-dimensional array  
**dataType[ ] arrayName;**  
**int[ ] list;**
- Instantiating an array object  
**arrayName = new dataType[intExp];**  
**list = new int[10];**

a positive integer that specifies the number of components in the array

PC-CS-305: Essentials of Information Technology

23

## One-Dimensional Arrays



- Example allocates a 12-element array of integers and links them to **month\_days**.

```
int [] month_days;
month_days = new int[12];
```

- **month\_days** will refer to an array of 12 integers. Further, all elements in the array will be initialized to zero.
- All array indexes start at zero.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## One-Dimensional Arrays



- **month\_days[1] = 28;** this statement assigns the value 28 to the second element of **month\_days**.
- **System.out.println(month\_days[3]);**  
This line displays the value stored at index 3.
- Example Array.java , Average.java

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Alternate ways to declare an array



- operator [ ] appears after the identifier, not after the data type

**int list[ ];**

- only alpha is an array reference variable; beta is an int variable

**int alpha[ ], beta;**

- both gamma and delta are to be array reference variables

**int [ ] gamma, delta;**

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Alternate ways to declare an array



- declaring a named constant integral type and then use the value of the named constant to specify the size of the array

**final int ARRAY\_SIZE = 10;**

**int[ ] list = new int[ARRAY\_SIZE];**

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## One-Dimensional Arrays



- Arrays can be initialized when they are declared. The process is much the same as that used to initialize the simple types.
- An *array initializer* is a list of comma-separated expressions surrounded by curly braces.

```
class AutoArray {
 public static void main(String args[]) {
 int month_days[] = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31,
 30, 31 };
 System.out.println("April has " + month_days[3] + " days.");
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Multidimensional Arrays



- *Multidimensional arrays are actually arrays of arrays.*
- To declare a multidimensional array variable, specify each additional index using another set of square brackets.
- For example, the following declares a twodimensional array variable called **twoD**.
- `int twoD[][] = new int[4][5];`

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



## Multidimensional Arrays

- *Example*

```
// Demonstrate a two-dimensional array.
class TwoDArray {
 public static void main(String args[]) {
 int twoD[][] = new int[4][5];
 int i, j, k = 0;
 for(i=0; i<4; i++) {
 for(j=0; j<5; j++) {
 twoD[i][j] = k;
 k++;
 }
 for(i=0; i<4; i++) {
 for(j=0; j<5; j++)
 System.out.print(twoD[i][j] + " ");
 System.out.println();
 }
 }
}
```

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology



## Strings

- Java's string type, called String, is not a simple type. Nor is it simply an array of characters.
- Rather, String defines an object, and a full description of it requires an understanding of several object-related features.
- The String type is used to declare string variables. You can also declare arrays of strings.

gauravgambhir.cse@piet.co.in

PC-CS-305: Essentials of Information Technology

## Strings



- A variable of type String can be assigned to another variable of type String. You can use an object of type String as an argument to `println()`.

- Example,

```
String str = "this is a test";
System.out.println(str);
```

- Here, str is an object of type String. It is assigned the string “this is a test”. This string is displayed by the `println()` statement.

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

PC-CS-305: Essentials of Information Technology

## Summary of Main Teaching Points



- Controlling Program Flow
  - Selection Control
    - if .. else construct
    - nested if.. else constructs
    - switch...case construct
  - Iteration Control
    - Counter controlled
    - Sentinel controlled

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

PC-CS-305: Essentials of Information Technology

## Summary of Main Teaching Points



- Type conversions and Casting
- Arrays
- Strings

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

PC-CS-305: Essentials of Information Technology

## Question and Answer Session



# Q & A

[gauravgambhir.cse@piet.co.in](mailto:gauravgambhir.cse@piet.co.in)

PC-CS-305: Essentials of Information Technology