International study centre

AN INTRODUCTION TO OBJECT-ORIENTATION AND THE JAVA PROGRAMMING LANGUAGE:

CONTROL STRUCTURES

❑Name : John Alamina
❑Email : john.alamina@hud.ac.uk

Outline

Types of control Structures

- Sequential
- Selection (branch)
- Iteration (loop)

Sequential Statements

- Commands
- Declaration e.g: int x;
- Assignment e.g. int x=45;
- Expressions e.g. z= x+y;
- Method calls e.g. Math.sin(x);

Any Questions?

Selection (Branch) control

- Control structures that are not sequential in nature by default will contain a condition statement.
- That is, execution will follow two or more different paths.
- Any condition that all paths never lead to previously executed statements is a selection.
- However, if any path returns back to a previously executed statement those set of statements are in a loop control.

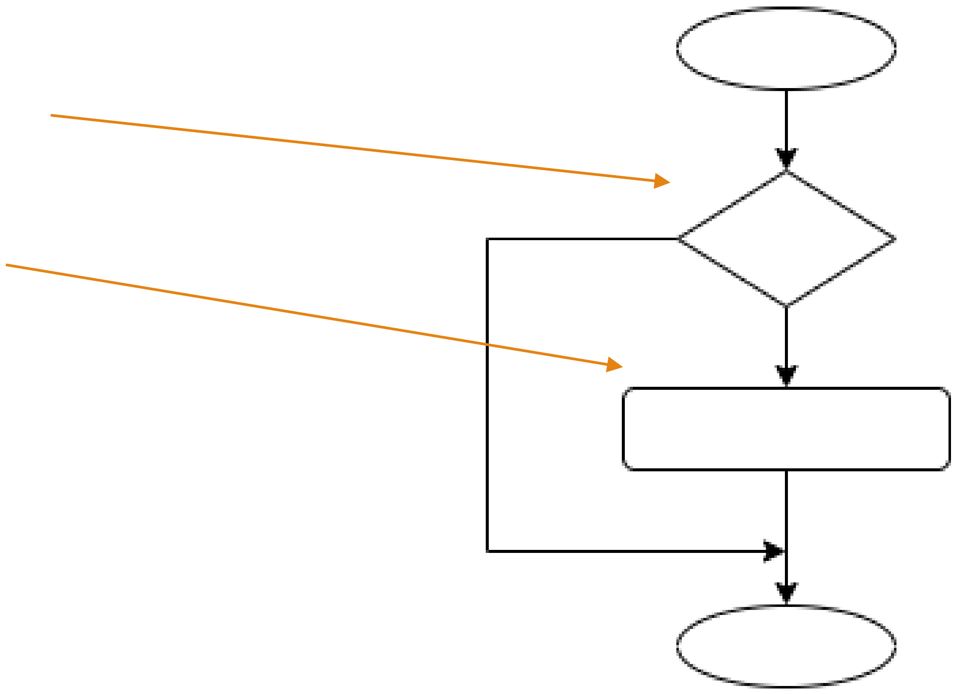
Selection types

1. **Unconditional if**
2. **Conditional-if**
3. **Bi-conditional if**
4. **Cascaded-if**
5. **Nested-if**
6. **Switch statement**

Unconditional-if

Has

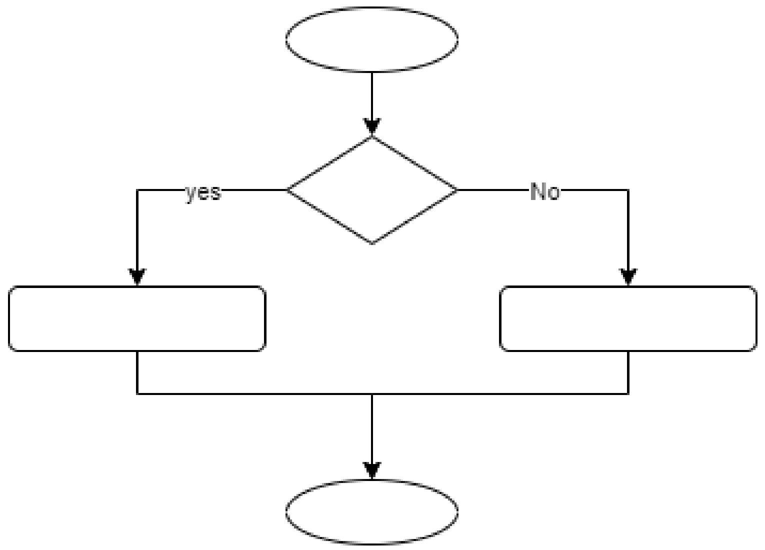
- **1 Condition**
- **1 Branch**
- **0 cascades**



Conditional-if

Has

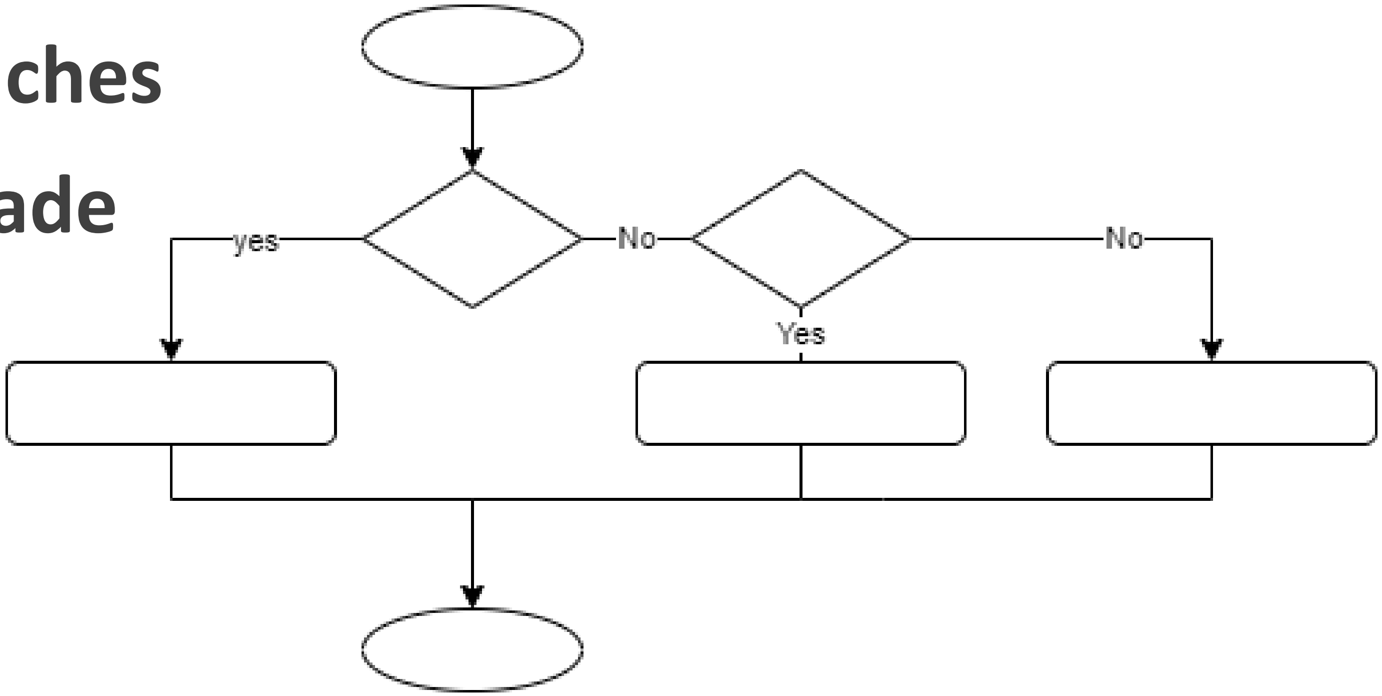
- **1 Condition**
- **2 Branches**
- **0 cascades**



Bi-Conditional-if

Has

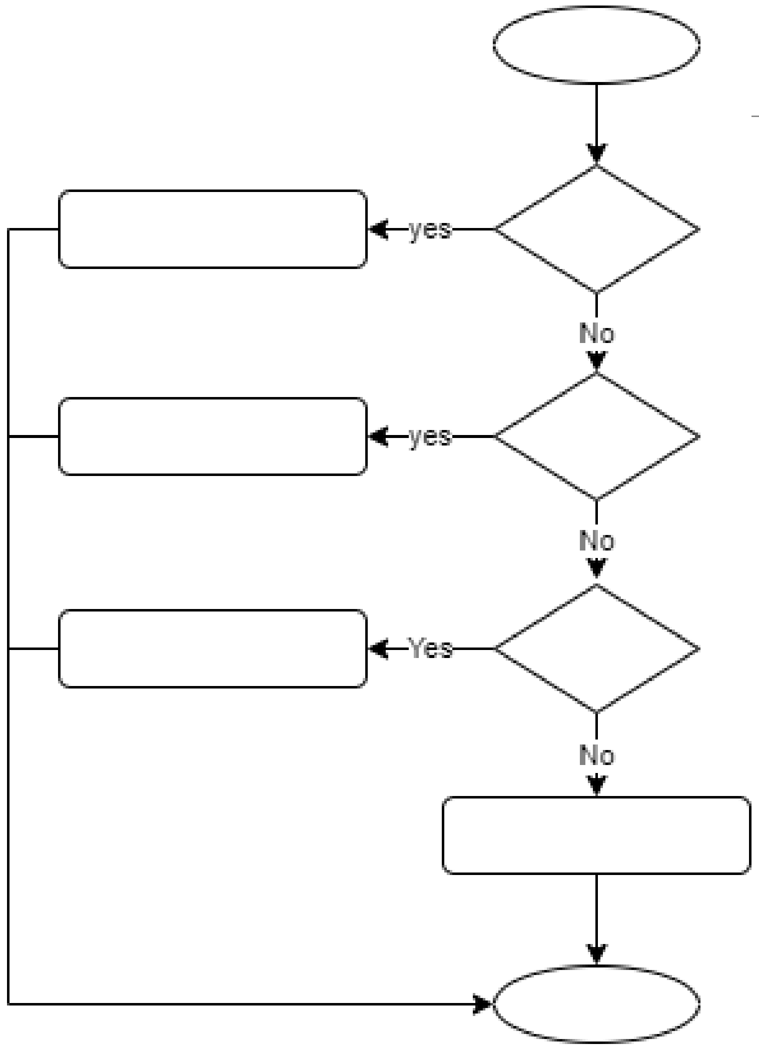
- **2 Conditions**
- **3 Branches**
- **1 cascade**



Cascaded-if

Has

- **3+ Conditions**
- **4+ Branches**
- **2+ cascade**



Branch Example

```
public class Selection{
   public static void main (String[] a){
      int A=5; //  A is going to be temperature in degrees C
      if(A==0){
         System.out.println( " Freezing! " ); //
      }else if(A<0){
         System.out.println( " Sub Zero " ); //
      }else{
         System.out.println( " Above zero " );
      }
   }
}
```

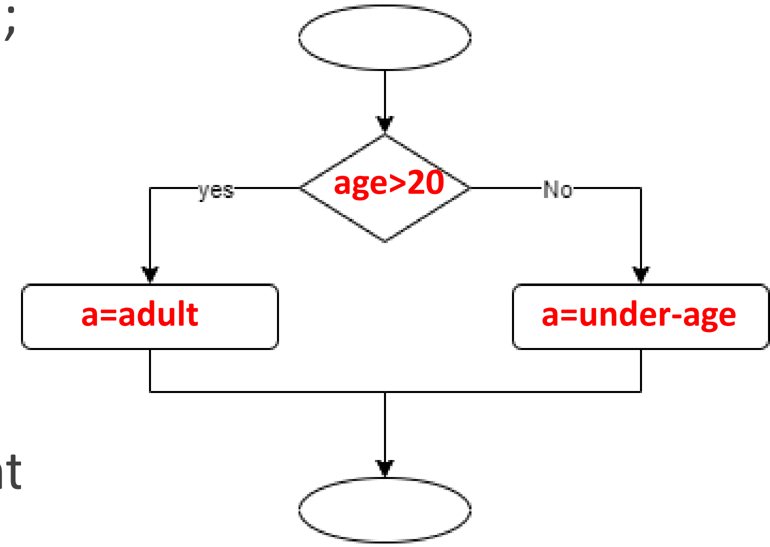
Any Questions?

Switch-statements

```
public class Cascade{
  public static void main(String []ar){
    int a=10, b=20;
    char op='+';
    int c=0;
    switch(op) {
      case '+':
        c = a + b;
        break;
      case '-':
        c = a - b;
        break;
      default:
        break;
    }
    System.out.println(c);
  }
}
```

Conditional Expression

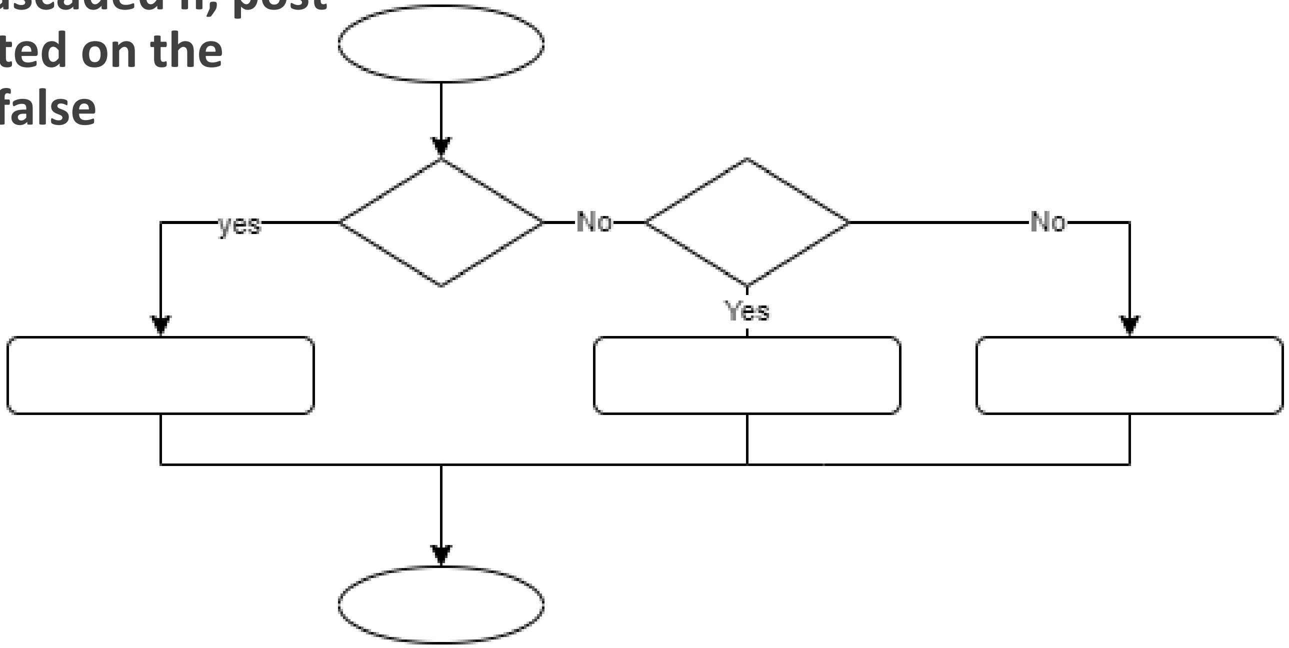
- This is a single expression that is equivalent to a conditional-if statement
- String a = age>20?"adult":"under-age";
- ternary operator (expression) takes:
  - condition
  - true value
  - false value
  - The equivalent algorithm looks like this
- The equivalent conditional-if statement
  - String a;
  - if (age>20){a="adult";}else{a="under-age";}



Any Questions?

Nested-if

The difference between nested and cascaded-if is that for a nested--if Post-condition is predicated on precondition being true while for the cascaded if, post condition is predicated on the precondition being false



Nested-if example

```
public class NestedIf{
    public static void main(String []ar){
        String pin="1000"; int bal=20;
        if(pin.equals("1000") {
            if (bal>20)
                System.out.println("can withdraw");
            else
                System.out.println("insufficient
balance");
         }else
            System.out.println("invalid pin");
    }
}
```

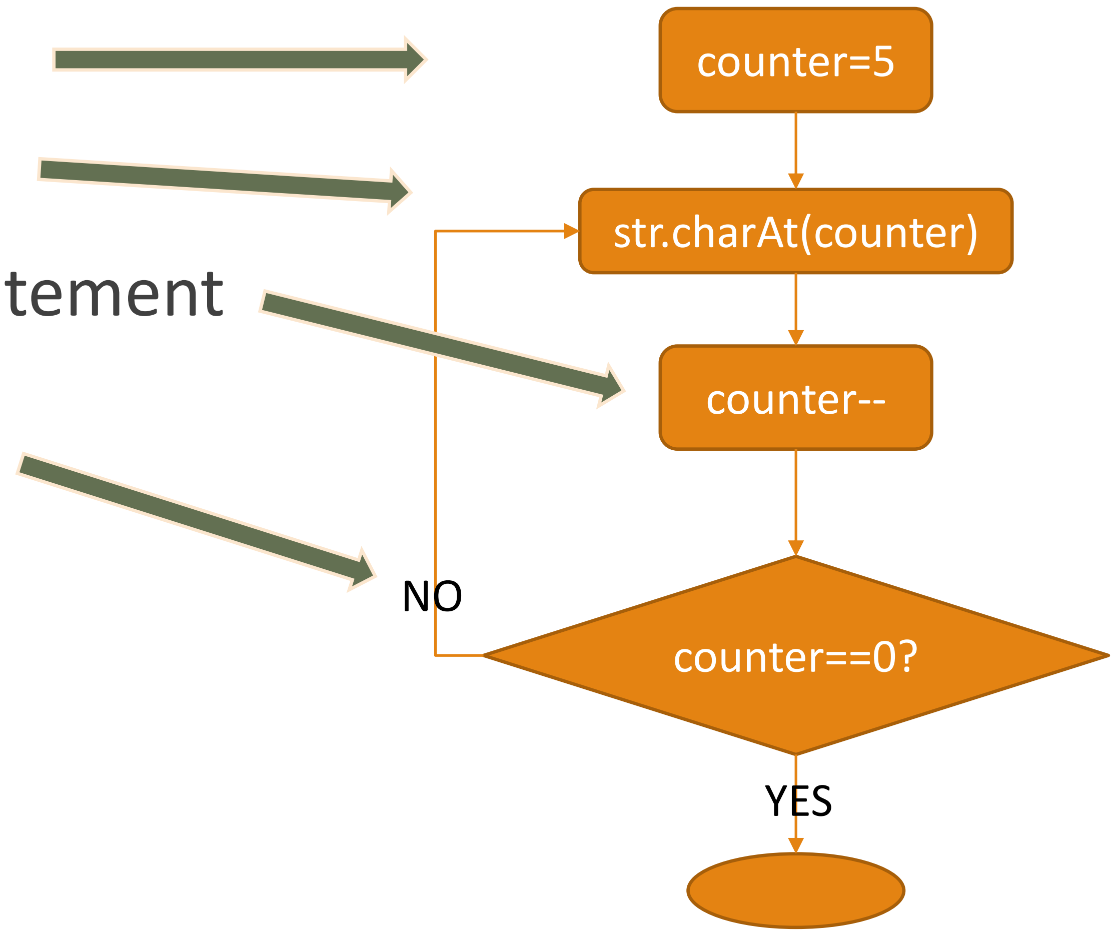
Any Questions?

Iteration (Loop) control

- Control structures that are not sequential in nature by default will contain a condition statement.
- That is, execution will follow two or more different paths.
- Any condition that all paths never lead to previously executed statements is a selection.
- However, if any path does return back to a previously executed statement those set of statements are in a loop control.

Loop Structure can have <u>**any**</u> of the following

- Count Initializer
- Loop body
- count update statement
- Stop condition



Loop Classes

- Deterministic
- Non-deterministic
- Pre-condition (for, and while loops)
- Post-condition (do-while loop)
- Infinite
- finite
- Nested

Loop class characteristics

Loop Class	Initialiser	Update	Can also be	Stop condition
Deterministic	☑	☑	Pre-post-finite	☑
Non-deterministic	☒	☒	Pre-post-finite	☑
Pre-condition	☒☑	☒☑	Deterministic-non-deterministic-finite	☒☑
Post-condition	☒☑	☒☑	Deterministic-non-deterministic-finite	☒☑
Infinite	☒	☒	Non-deterministic	☒
finite	☒☑	☒☑	Pre-post-deterministic-non-deterministic	☑

Loop Example

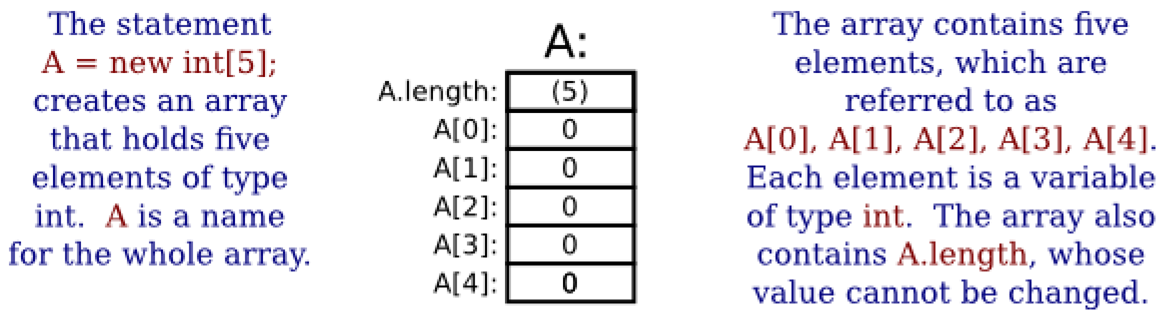
```
public class Control{
    public static void main(String []a){
        int length;
        int width;
        for(int i=0;i<length;i++){
            for(int j=0;j<width;j++){
                System.out.print("*");
            }
            system.out.println("\n");
        }
    }
}
```

Arrays

- Storage for a collection of similar types (primitive or advanced) defined with a single variable known as an array.
- Arrays are indexed collections where each element within the array is accessed using an ordered sequence of positive whole numbers starting from zero.
- Can be iterated (accessed sequentially in order) using a deterministic loop.
- Arrays in themselves are objects. They need to be instantiated using the “new” keyword.
- n-dimensional arrays can be iterated using nested loops.
- Values of arrays for primitive types are defaulted to 0

Using arrays

```
//Store 3 Numbers using an integer variable;
int i=5;
i=6;
i=7;
System.out.println(“”+i); //only stores the most recent value
int i1=5,i2=6,i3=7;
System.out.println(“”+i1+” “+i2+” “+i3); //works but will be difficult for
storing large number of values
int [] arr=new int[3]; //declare array
arr[0]=5; arr[1]=6; arr[2]=7;//array assignment starting from zero.
for(int i=0;i<arr.length;i++)System.out.print(arr[i]);//iterating an array
```



The statement A = new int[5]; creates an array that holds five elements of type int. A is a name for the whole array.

The array contains five elements, which are referred to as A[0], A[1], A[2], A[3], A[4]. Each element is a variable of type int. The array also contains A.length, whose value cannot be changed.

math.hws.edu/javanotes/c3/s8.html

Any Questions?

Exercises

1. A) Write a program to determine whether the value of an integer variable initialised with the name temp is sub-zero, above zero degrees, exactly zero degrees. If above zero it should display "Above Zero". If exactly zero it should display "Freezing" and if under zero should display "Sub zero".

   B) What type of branch control structure is this program?

2. Write a Java program that will output the grade of an integer score entered into the program. 70-100 is a A, 60-69 is a B, 50-59 is a C, 45-49 is a D, 40-44 is an E, and 0-39 is a R. The program should only exit when a value outside these ranges is entered.

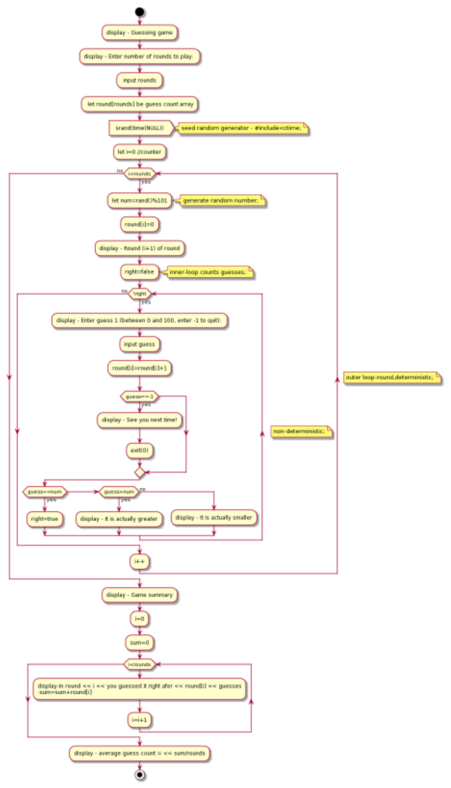
   B) What type of loop control structure is this program?

3. For each of the above two programs, draw the equivalent flowchart and paste the draw.io image and url to the padlet board

Take-home exercise

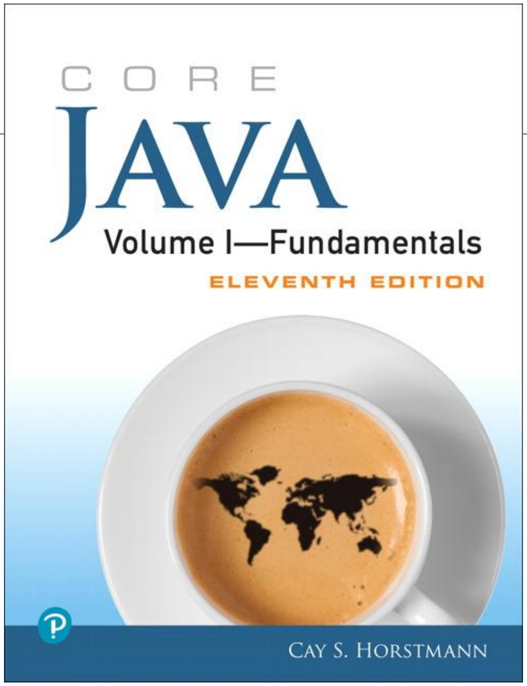
3. Write a Java class to implement the guessing game given by the following algorithm (ctrl-click image to enlarge)

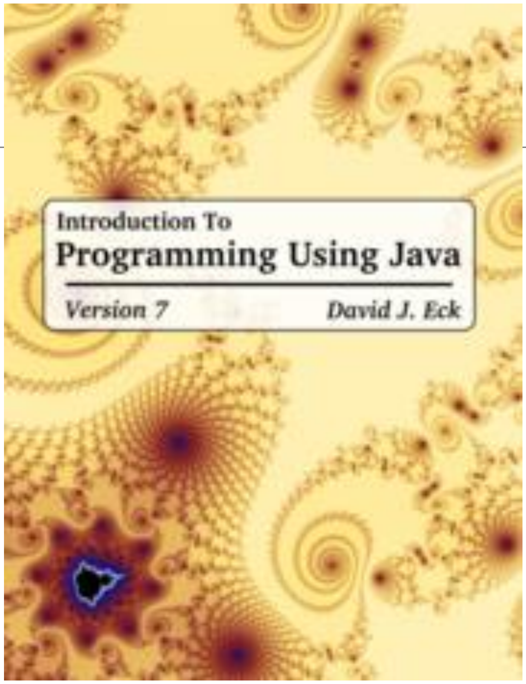
Guessing game algorithm



display - Guessing game
display - Enter number of rounds to play:
input rounds
let round[rounds] be guess count array
srand(time(NULL))
seed random generator - #include<ctime;
let i=0 //counter
i<rounds
no
yes
let num=rand()%101
generate random number;
round[i]=0
display - Round (i+1) of round
right=false
inner-loop counts guesses;
!right
no
yes
display - Enter guess 1 (between 0 and 100, enter -1 to quit):
input guess
round[i]=round[i]+1
outer loop-round,deterministic;
guess==-1
yes
display - See you next time!
exit(0)
non-deterministic;
guess==num
yes
guess<num
no
yes
right=true
display - It is actually greater
display - It is actually smaller
i++
display - Game summary
i=0
sum=0
i<rounds
display-In round << i << you guessed it right afer << round[i] << guesses
:sum=sum+round[i]
i=i+1
display - average guess count = << sum/rounds

Supplementary material





- The Java Tutorial
- Java API documentation
- Link to today's Session screencast
- Link to John's Group Padlet
- Link to Kelly's Group Padlet