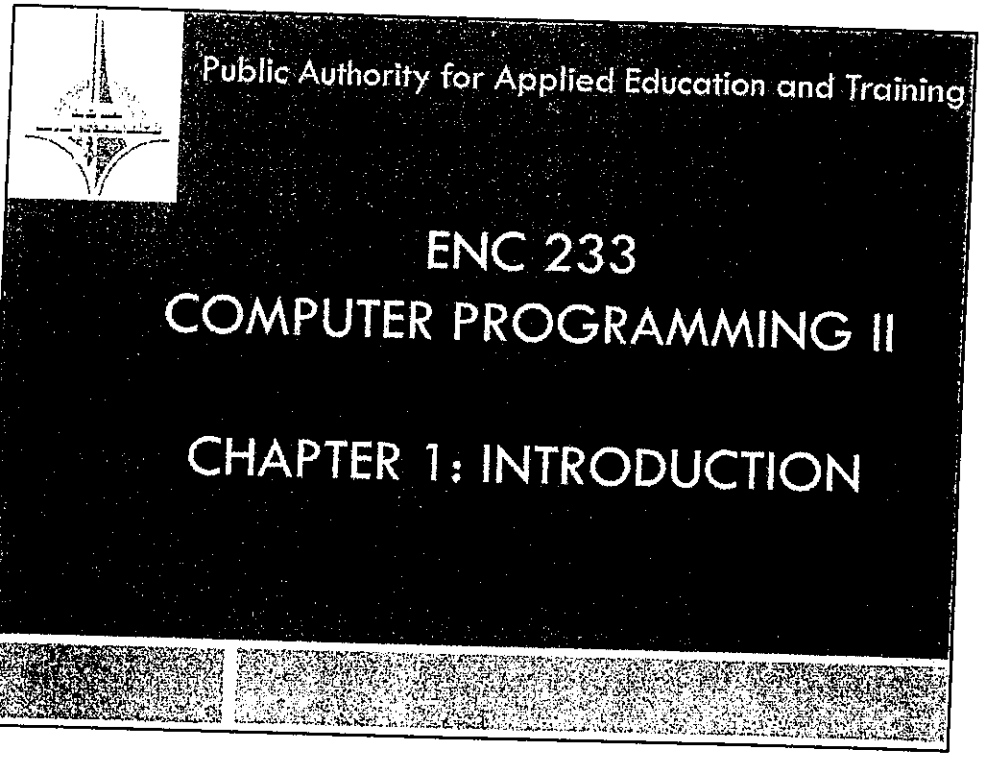
لغه برمجه 2

م. مريم الجيماز



Chapter1 Objectives

- Learning about programming
- Finding out about Java
- Comparing Java with other programming languages
- Terminology in Java's technology
- Flowchart In Programming

Introduction

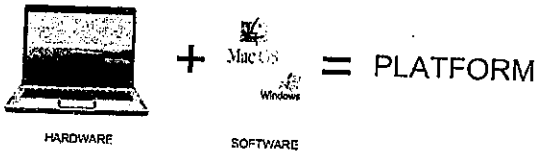
- What is Programming?
  - Tell the computer what to do! In the language that the computer understands.
- What is Java?
  - Programming language named after the famous island that grows coffee which programmers are famously known for drinking!

Cont. Introduction

- Java is a programming language.
  - You are already familiar with other programming languages, C and VB, you only need to learn the various features of Java and syntax!

Platform Independence

- Java is platform independent
  - Java programs can run on different types of computers
- A Java program runs on any computer with a Java Runtime Environment (JRE)
- JRE is available for almost every type of computer
  - PCs running Windows, Mac, Unix or Linux
  - Huge mainframes computers
  - Cellphones



http://j4school.files.wordpress.com/2013/01/platform.gif

Cont. Platform Independence

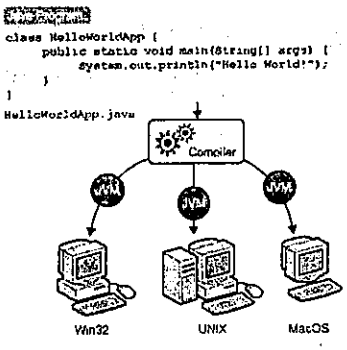
- Other programming languages provide compatible compilers for different platforms.
- What is a complier?
  - A program that translate programs written in a programming language into a form that can actually run on a computer, machine language.
- Compile different versions of the programs for each platform.
- Different platform compilers are not identical!

Cont. Platform Independence

- Java's platform independence is based on the concept of Java Virtual Machine (JVM).
- JVM is a hypothetical computer platform
  - Does not really exist as actual hardware
- JRE is an emulator. A program that sets aside part of your hard drive to act like a computer (JVM) that can execute Java programs.

Cont. Platform Independence

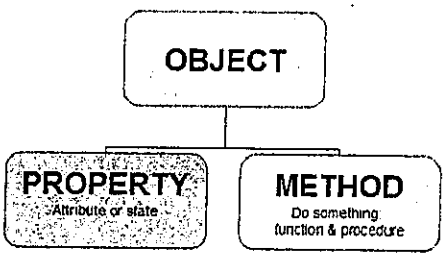
- Java compiler translates the Java program into the machine language of the JVM.
  - Known as the bytecode
- The JRE runs the bytecode in the JVM.
- You can execute a Java program on any computer that has JRE installed, without recompiling the program.



http://insaneberne.yolasite.com/resources/helloWorld.gif

Object Orientation

- Java is an Object Oriented programming language.
- Java programs are made up from programming elements called **objects**.
- Objects have two characteristics:
  - State (data)
  - Behavior (methods)

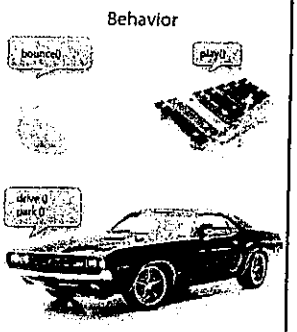


http://www.teachitza.com/delphi/object.gif

http://docs.oracle.com/javase/tutorial/java/concepts/object.html

Objects

- Dogs have state (name, colour, breed, hungry) and behaviour (barking, fetching, wagging tail).
- Bicycles also have state (current gear, current pedal cadence, current speed) and behaviour (changing gear, changing pedal cadence, applying brakes).
- Books have state (title, author, publisher) and behavior (get book title).



http://365cloud.internaldrive.com/2011/12/behavior.jpg

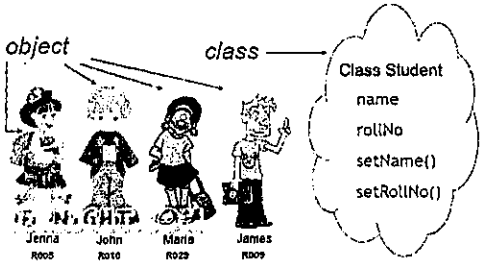
Java Class

- Classes are closely related to objects.
- A class is a program code you write to create objects.
- Class describes the data and methods that define the object's state and behavior.
- When the program executes, classes are used to create objects.

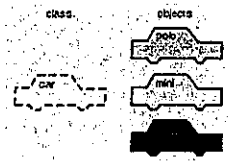
http://docs.oracle.com/javase/tutorial/java/concepts/class.html

Cont. Java Class

- A class is the blueprint from which individual objects are created.



http://www.deknight.com/wp-content/uploads/2011/10/java_object_class_difference.png

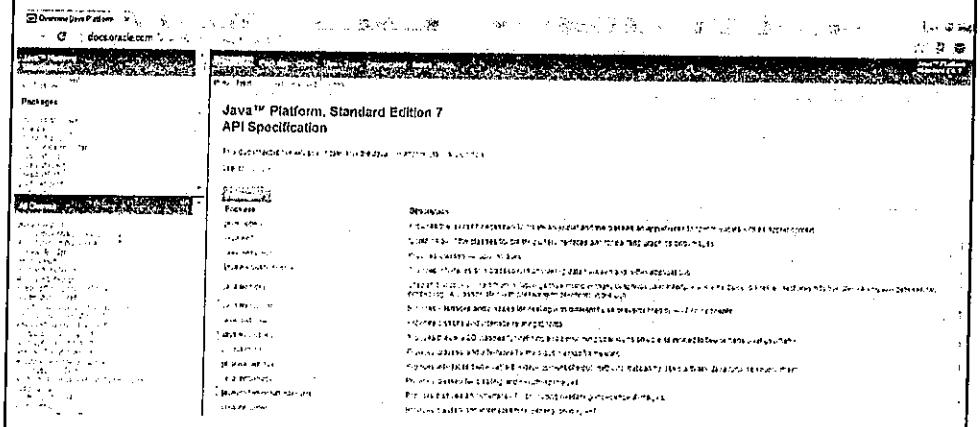


http://upload.wikimedia.org/wikipedia/commons/thumb/6/62/CPT-OOP-objects_and_classes.svg/226px-CPT-OOP-objects_and_classes.svg.png

Java API

- Java comes with a library of classes that provide commonly used functions.
- This library is called the Java API.
- The java API has classes that let you write data to files, create windows on-screen, retrieve information from a database.
- Java API documentation:
  http://docs.oracle.com/javase/7/docs/api/index.html

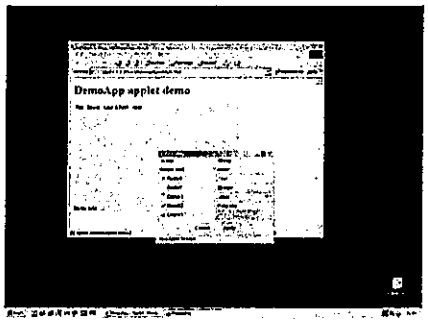
Cont. Java API



Java Program Types

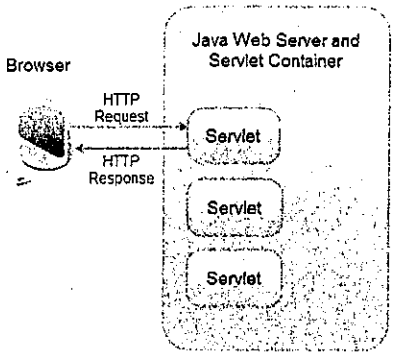
- Two types of Java programs on the Internet:
  - Applets
    - Run directly within a web browser.
    - Browser starts a JVM which runs the applet's bytecode.
  - Servlets
    - Web-based Java programs that run on servers rather than the user's web browser.
    - Generates a page of HTML and sends to user.

Cont. Java Program Types



http://new.linuxfocus.org/common/images2/article350/java_applet.jpg

http://www.codespread.com/wp-content/uploads/2013/06/java.jpg



http://file.mrbool.com/mrbool/articles/abhimanyu/MYSQLjsp/ServletsOverview01.png

Comparing Programming Languages

- Each executable statement ends with a semicolon.
- Braces {} are used to mark blocks of code.
- Programs starts from main routine.

```
#include <stdio.h>

main()

{
   printf("hello, world\n");
}
```

http://2.bp.blogspot.com/_JoC7eqPOTQ/TT82CdxxTZI/AAAAAAAAAYo/AVOndpA_xEE/s400/hello.jpg

```
class HelloWorld
{
    public static void main(String[] a)
    {
        System.out.println("Hello world");
    }
}
```

http://taylorpetrick.files.wordpress.com/2011/07/hello_world_java.png

Java Features

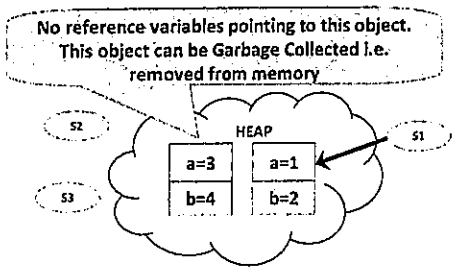
- Type Checking
  - Assign value of the same type as the variable's

```
int a = 2;
String b = "Hello!";
String c = a * b
```

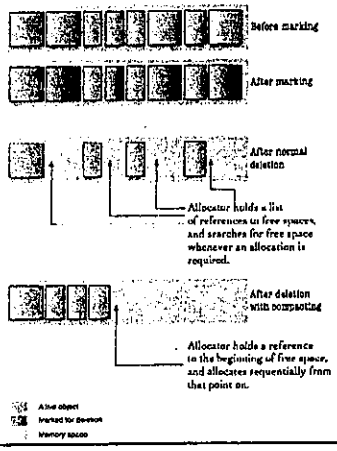
  - Error compiling the above lines of code, can not multiply an integer and a string!

Cont. Java Features

- Automatic memory management
  - Memory is freed up automatically when no longer needed (garbage collector process)



http://www.javatutorialhub.com/wp-content/uploads/2012/07/GarbageCollection4.jpg



http://cheoticjava.com/wp-content/uploads/2008/01/Memory%20collection.png

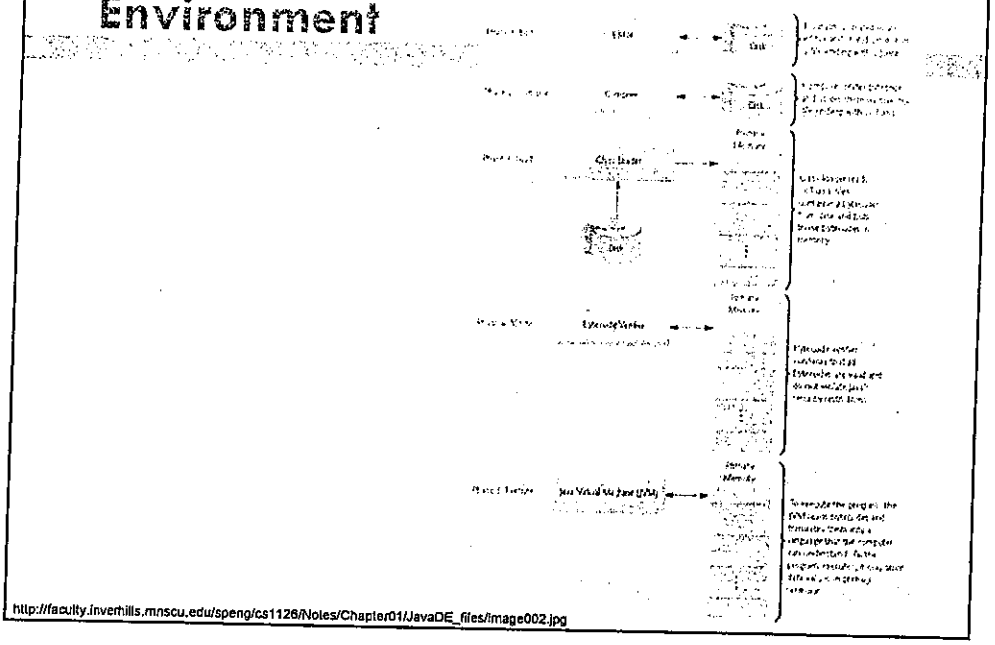
Cont. Java Features

- Exception Handling
  - Statements that can potentially cause an exception are bracketed by a code that can catch and handle that exception

```
public void writeList() {
    PrintWriter out = null;

    try {
        System.out.println("Entering" + " try statement");

        out = new PrintWriter(new FileWriter("OutFile.txt"));
        for (int i = 0; i < SIZE; i++)
            out.println("Value at: " + i + " = " + vector.elementAt(i));
    } catch (ArrayIndexOutOfBoundsException e) {
        System.err.println("Caught ArrayIndexOutOfBoundsException: "
                           + e.getMessage());

    } catch (IOException e) {
        System.err.println("Caught IOException: " +  e.getMessage());

    } finally {
        if (out != null) {
            System.out.println("Closing PrintWriter");
            out.close();
        }
        else {
            System.out.println("PrintWriter not open");
        }
    }
```

http://docs.oracle.com/javase/tutorial/essential/exceptions/putItTogether.html

Typical Java Development Environment



http://faculty.inverhills.mnscu.edu/speng/cs1126/Notes/Chapter01/JavaDE_files/image002.jpg

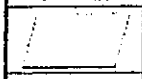
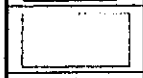
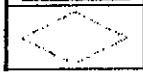
Cont. Typical Java Development Environment

- Java programs normally undergo five phases
  - Edit
    - Programmer writes program in an editor, and stores it on disk
    - *Source code* (name.java)
  - Compile
    - Compiler creates *bytecodes* from program
    - Bytecodes are portable (machine-independent)
    - name.class
  - Load
    - Class loader loads bytecodes from disk into memory
  - Verify
    - Verifier ensures bytecodes do not violate security requirements
  - Execute
    - Interpreter (JVM) translates bytecodes into machine language

Dietel & Associates – Chapter 1 – Slide 20

Flowchart In Programming

- Flowchart are very helpful in writing program and explaining program to others.

Symbol	Purpose	Description
	Flow line	Used to indicate the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Used to represent start and end of flowchart.
	Input/Output	Used for input and output operation.
	Processing	Used for airthmetic operations and data-manipulations.
	Desicion	Used to represent the operation in which there are two alternatives, true and false.

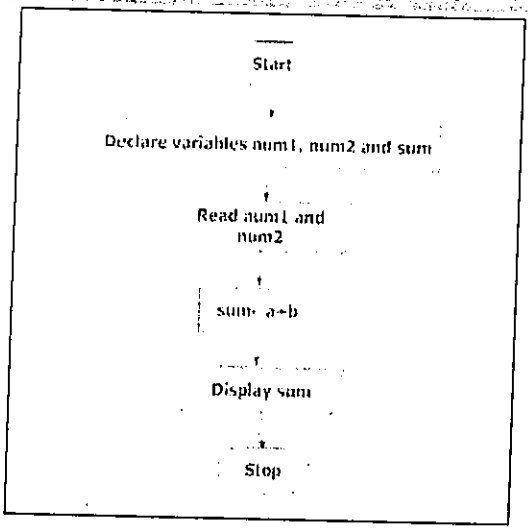
http://www.programiz.com/article/flowchart-programming

Example 1

- Draw a flowchart to add two numbers entered by user.

http://www.programiz.com/article/flowchart-programming

Cont. Example 1



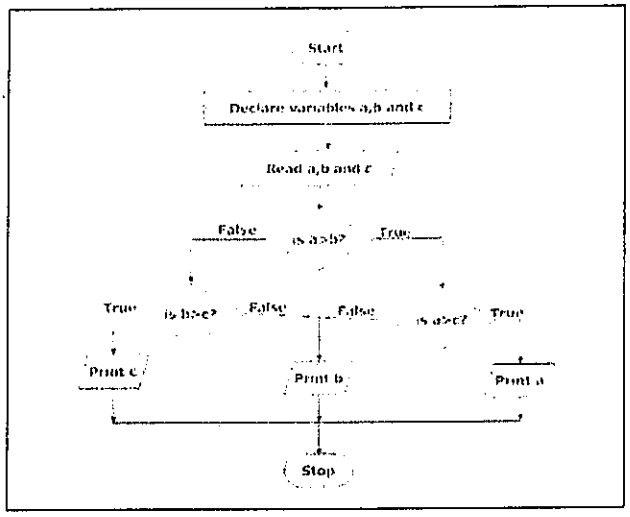
http://www.programiz.com/article/flowchart-programming

Example 2

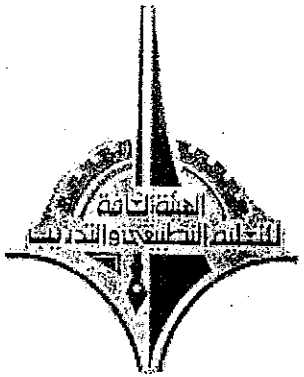
- Draw flowchart to find the largest among three different numbers entered by user.

http://www.programiz.com/article/flowchart-programming

Cont. Example 2



http://www.programiz.com/article/flowchart-programming



Public Authority for Applied Education and Training

ENC 233
COMPUTER PROGRAMMING II

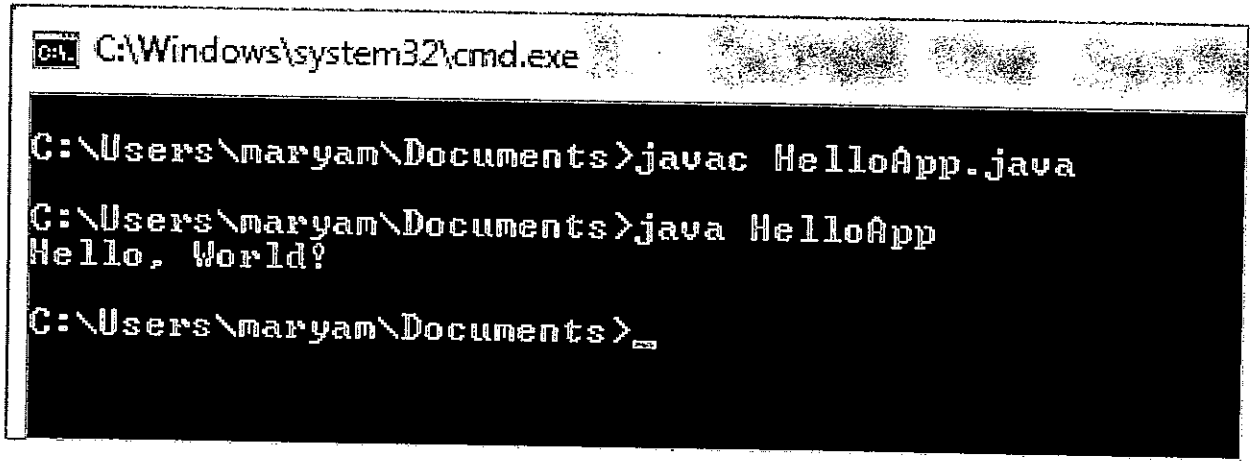
CHAPTER 2: PROGRAMMING BASICS

Chapter 2 Objectives

- Learning about Hello, World! Program
- Introduce Java keywords, statements and blocks
- Creating proper variable declarations
- Learning about data types
- Working with Strings
- Understand input and output streams

The HelloApp Program

```
1  public class HelloApp
2  {
3      public static void main(String[] args)
4          {
5              System.out.println("Hello, World!");
6          }
7  }
```



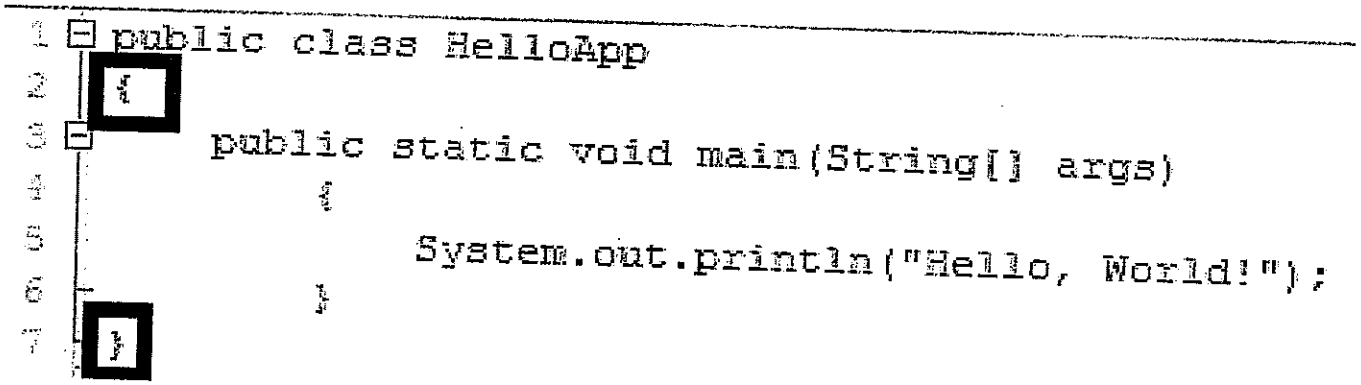
Cont. The HelloApp Program

- public
  - A keyword
  - Indicates that the Java element (HelloApp class) is a public class, therefore other classes can use it.
  - Alternative to public is private
- class
  - A keyword
  - The element defined here is a class
- HelloApp
  - An identifier

```
1  public class HelloApp
2  {
3      public static void main(String[] args)
4      {
5          System.out.println("Hello, World!");
6      }
7  }
```

Cont. The HelloApp Program

- The opening brace { marks the beginning of the body of the class and the closing brace } marks the end of the body of the class



Cont. The HelloApp Program

- main is a method of the HelloApp class
  - static is required when you declare the main method
  - void is a keyword that indicates that no value is returned by the method
  - main is an identifier that provides the name of the method
  - (String[] args) is the parameters list that is passed to the method

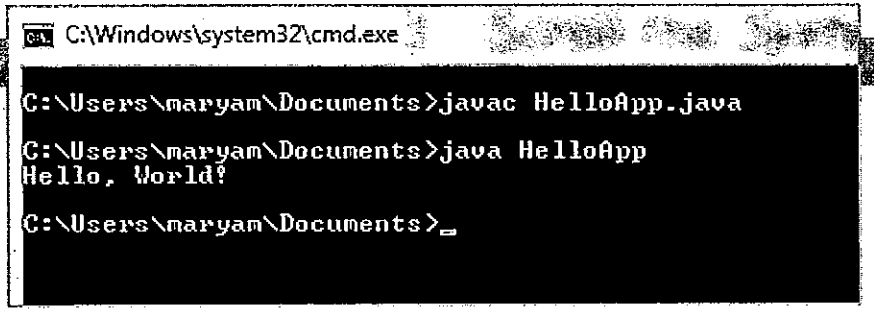
```
public class HelloApp
{
    public static void main(String[] args)
    {
        System.out.println("Hello, World!");
    }
}
```

Cont. The HelloApp Program

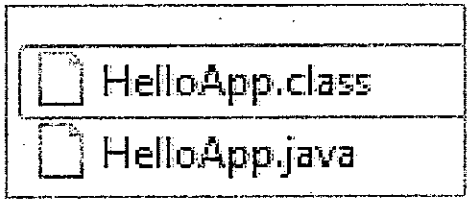
- Only statement in the entire program
- Calls a method named println that belongs to the System.out object
- println method displays a line of text on the console
- Statement ends with a semi colon ;

```
1  public class HelloApp
2  {
3      public static void main(String[] args)
4          {
5              System.out.println("Hello, World!");
6          }
7  }
```

Cont. The HelloApp Program



- javac HelloApp.java
  - Compile the java file
  - Creates a class file (HelloApp.class) that contains the Java bytecodes
- java HelloApp
  - Runs the program
  - Loads the JVM into memory
  - Locates HelloApp class contained in HelloApp.class file
  - Runs the main method of the HelloApp class



Java's Keywords

- Keyword has a special meaning and can not be used as identifiers in the program

abstract	assert	boolean	break
byte	case	catch	char
class	const	continue	default
do	double	else	enum
extends	final	finally	float
for	goto	if	implements
import	instanceof	int	interface
long	native	new	package
private	protected	public	return
short	static	strictfp	super
switch	synchronized	this	throw
throws	transient	try	void
volatile	while		

http://4.bp.blogspot.com/-M0TXqJV560o/T3VTpewVuuI/AAAAAAAAAEY/U_viNDGRrRI/s1600/keywords.png

Cont. Java's Keywords

- Keywords are case-sensitive.
- If you type **If** instead of **if** or **For** instead of **for**, the complier complains about the error.

caseSensitive.java

```
1 public class caseSensitive {
2     public static void main(String[] args) {
3         For(int i = 0; i<3; i++)
4             System.out.println("Loop" + i);
5     }
6 }
```

```
C:\Users\maryam\Documents\Java>javac caseSensitive.java
caseSensitive.java:3: error: '.class' expected
        For(int i = 0; i<3; i++)
                 ^
caseSensitive.java:3: error: illegal start of type
        For(int i = 0; i<3; i++)
                       ^
caseSensitive.java:3: error: not a statement
        For(int i = 0; i<3; i++)
                       ^
caseSensitive.java:3: error: ';' expected
        For(int i = 0; i<3; i++)
                            ^
4 errors
```

caseSensitive.java

```
1 public class caseSensitive {
2     public static void main(String[] args) {
3         for(int i = 0; i<3; i++)
4             System.out.println("Loop" + i);
5     }
6 }
```

```
C:\Users\maryam\Documents\Java>javac caseSensitive.java

C:\Users\maryam\Documents\Java>java caseSensitive
Loop0
Loop1
Loop2
```

Types of Statements

- Declaration statements
  - Create variables you can use to store data

```
int i;
String s = "This is a string";
```

- Expression statements
  - Performs calculations

```
i = a + b;
salesTax = invoiceTotal * taxRate;
```

- Other statements
  - if-then
  - for, while, or do

End-of-line Comments

- Comments are ignored by the compiler

```
/**
 * This is the main method
 */
public static void main(String[] args) {

    //Calculate the sales tax
    salesTax = invoiceTotal * taxRate;

}
```

Declaring Variables

- A variable is a storage location that has a name and contains information.
- Basic form of a variable declaration:
  - type *name*;

```
int x;
String firstName;
double circle_radius;
char a, b, c;
```

Cont. Declaring Variables

- The variable name must not be a keyword or reserved word.
- Variable names are case-sensitive.
- All variable names must begin with a letter of the alphabet, an underscore (_), or a dollar sign ($).
- The name can be of any length.
- After the first initial letter, variable names may also contain letters and the digits 0 to 9. No spaces or special characters are allowed.

Cont. Declaring Variables

Samples of acceptable variable names: **YES**	Samples of unacceptable variable names: **NO**
Grade	Grade(Test)
GradeOnTest	GradeTest#1
Grade_On_Test	3rd_Test_Grade
GradeTest	Grade Test (has a space)

http://mathbits.com/MathBits/Java/DataBasics/Namingrules.htm

Initializing Variables

- Giving variables value using an assignment statement.
  - type *name* = expression;

```
int x = 1;
String firstName = "Fatma";
double circle_radius = 3.54;
char a = 'k', b = 'v', c='r';
```

Primitive Data Types

Data Type	Size	Default Value	Range
boolean	1 bit	false	-
byte	8 bit	0	-128 to +127
char	16 bit	\u0000	\u0000 to \uFFFF
short	16 bit	0	-32768 to +32767
int	32 bit	0	-2147483648 to +2147483647
float	32 bit	0.0	1.4E-45 to 3.4028235E38
long	64 bit	0	-9223372036854775808 to +9223372036854775807
double	64 bit	0.0	4.9E-324 to 1.7976931348623157E308

http://i0.wp.com/programmingtpoint.com/wp-content/uploads/2013/08/java-data-types.jpg

Cont. Primitive Data Types

```
public class variables {

    public static void main(String[] args) {

        int x = 1;

        System.out.println("The integer x has a value = " + x);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac variables.java

C:\Users\user\Documents\Java Programs>java variables
The integer x has a value = 1
```

Cont. Primitive Data Types

```
public class variables {

    public static void main(String[] args) {

        double y = 3.14;

        System.out.println("The double y has a value = " + y);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac variables.java

C:\Users\user\Documents\Java Programs>java variables
The double y has a value = 3.14
```

Cont. Primitive Data Types

```
1  public class variables {
2
3      public static void main(String[] args) {
4
5          int z = 3.14;
6
7          System.out.println("The double z has a value = " + z);
8
9      }
10 }
```

```
C:\Users\user\Documents\Java Programs>javac variables.java
variables.java:5: error: possible loss of precision
        int z = 3.14;
                ^
  required: int
  found:    double
1 error
```

Special Characters

- Escape sequence to represent special characters.

Escape Sequence	Meaning
\b	backspace
\t	tab
\n	newline
\r	carriage return
\"	double quote
\'	single quote
\\	backslash

http://faculty.simpson.edu/lydia.sinapova/www/cmsc150/LN150_Lewis/L03-Fig03.jpg

Cont. Special Characters

```
public class specialCharacters {

    public static void main(String[] args) {

        System.out.println("Name\tAge");
        System.out.println("-----------");
        System.out.println("Sara\t20");
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac specialCharacters.java

C:\Users\user\Documents\Java Programs>java specialCharacters
Name    Age
-----------
Sara    20
```

Cont. Special Characters

```
1  public class specialCharacters {
2
3      public static void main(String[] args) {
4
5          System.out.println("Programming\n");
6          System.out.println("is FUN!");
7          System.out.println("\n*****************\n");
8          System.out.println("Programming");
9          System.out.println("is FUN!");
10     }
11 }
```

```
C:\Users\user\Documents\Java Programs>javac specialCharacters.java

C:\Users\user\Documents\Java Programs>java specialCharacters
Programming

is FUN!

*****************

Programming
is FUN!
```

Cont. Special Characters

```
1  public class specialCharacters {
2
3      public static void main(String[] args) {
4
5          System.out.println("The sky is \"blue\".");
6
7      }
8  }
```

```
C:\Users\user\Documents\Java Programs>javac specialCharacters.java

C:\Users\user\Documents\Java Programs>java specialCharacters
The sky is "blue".
```

Concatenating Strings

- Combing two strings or more by using the plus sign (+)

```
public class concatination {

    public static void main(String[] args) {
        String h = "Hello,";
        String w = "World!";

        String greeting = h + w;

        System.out.print(greeting);
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac concatination.java

C:\Users\user\Documents\Java Programs>java concatination
Hello,World!
```

Cont. Concatenating Strings

```
public class concatination {

    public static void main(String[] args) {
        String h = "Hello,";
        String w = "World!";

        String greeting = h + "\t" + w;

        System.out.print(greeting);
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac concatination.java

C:\Users\user\Documents\Java Programs>java concatination
Hello,  World!
```

Cont. Concatenating Strings

```
public class concatination {

    public static void main(String[] args) {
        String h = "Hello,";
        String w = "World!";

        System.out.print(h + "\t" + w);
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac concatination.java

C:\Users\user\Documents\Java Programs>java concatination
Hello,  World!
```

Standard Output Stream

- Display text output on-screen
  - System.out.print()
    - It prints on the same line
    - It does not start a new line
  - System.out.println()
    - It does start a new line

Cont. Standard Output Stream

```
1  public class outputStream {
2      public static void main(String[] args) {
3          int i = 2;
4          int j = 5;
5          System.out.print(i);
6          System.out.print(" and ");
7          System.out.print(j);
8          System.out.print("\n");
9      }
10 }
```

```
C:\Users\user\Documents\Java Programs>javac outputStream.java

C:\Users\user\Documents\Java Programs>java outputStream
2 and 5
```

Getting Input with the JOptionPane Class

- Display a dialog box to get text input from the user
  - JOptionPane.showInputDialog(*message*)
- The JOptionPane class is part of the javax.swing package
  - Need to add import javax.swing.JOptionPane statement to the beginning of the program.

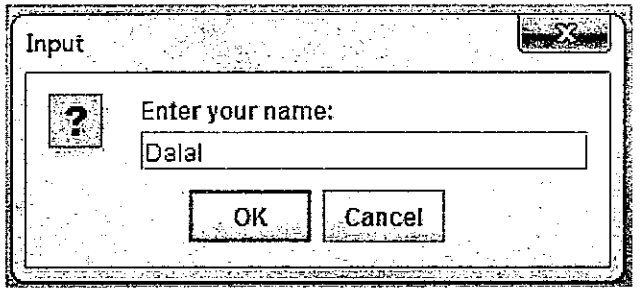
Cont. Getting Input with the JOptionPane Class

```
public class inputStream {

    public static void main(String[] args) {

        String firstName = "Lulwa";

        System.out.println("Hello! My name is " + firstName);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac inputStream.java

C:\Users\user\Documents\Java Programs>java inputStream
Hello! My name is Lulwa
```

Cont. Getting Input with the JOptionPane Class

```
import javax.swing.JOptionPane;

public class inputStream {

    public static void main(String[] args) {

        String firstName = JOptionPane.showInputDialog("Enter your name:");

        System.out.println("Hello! My name is "+ firstName);

    }
}
```

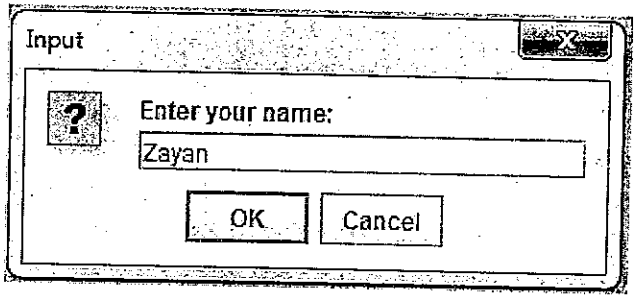


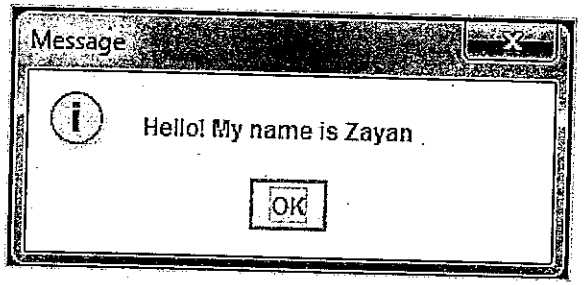

```
C:\Users\user\Documents\Java Programs>javac inputStream.java

C:\Users\user\Documents\Java Programs>java inputStream
Hello? My name is Dalal
```

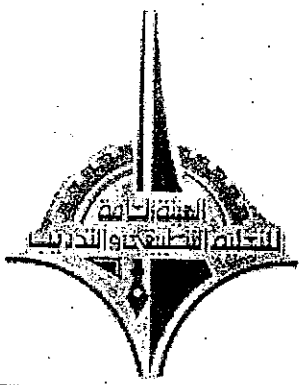
Getting Output with the JOptionPane Class

- Brings up a dialog displaying a message
  - JOptionPane.showMessageDialog(null, *message*);

```
import javax.swing.JOptionPane;

public class inputStream {

    public static void main(String[] args) {

        String firstName = JOptionPane.showInputDialog("Enter your name:");

        JOptionPane.showMessageDialog(null,"Hello! My name is "+ firstName);

    }
}
```







Public Authority for Applied Education and Training

ENC 233
COMPUTER PROGRAMMING II

CHAPTER 3: WORKING WITH NUMBERS AND EXPRESSIONS

Chapter 3 Objectives

- Learning about operators
- Learn more about expressions
- Incrementing and decrementing
- Introduce some functions in the Math class

Arithmetic Operators

- Special symbol used to perform a mathematical operation.

Operator	Name	Example expression	Meaning
*	Multiplication	a * b	a times b
/	Division	a / b	a divided by b
%	Remainder (modulus)	a % b	the remainder after dividing a by b
+	Addition	a + b	a plus b
-	Subtraction	a - b	a minus b

http://www.expertsmind.com/CMSImages/1411_operater.png

Cont. Arithmetic Operators

```
public class arithmetic {

    public static void main(String[] args) {
        int a = 21, b = 5;
        int c = a + b;
        int d = a - b;
        int e = a * b;
        int f = a / b;
        int g = a % b;

        System.out.println("c = " + c);
        System.out.println("d = " + d);
        System.out.println("e = " + e);
        System.out.println("f = " + f);
        System.out.println("g = " + g);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac arithmetic.java

C:\Users\user\Documents\Java Programs>java arithmetic
c = 26
d = 16
e = 105
f = 4
g = 1
```

Unary Operators

- Operators that work on just one operand.

++expr	Prefix increment	Adds 1 to the value of the expression that follows
--expr	Prefix decrement	Subtracts 1 from the value of the expression that follows
+expr	Unary plus	Indicates a positive number (usually redundant)
-expr	Unary minus	Negates an expression (including literals)

http://www.java-forums.org/attachments/ocaj/3743d1335949797t-tutorial-review-java-operators-java-certification-exam-d3-unaryoperators.jpg

Operator	Use	Description
++	op++	Increments op by 1; evaluates to the value of op before it was incremented
++	++op	Increments op by 1; evaluates to the value of op after it was incremented
--	op--	Decrements op by 1; evaluates to the value of op before it was decremented
--	--op	Decrements op by 1; evaluates to the value of op after it was decremented

http://1.bp.blogspot.com/_wHvjCgYEZZA/SaJ0xL-4fRI/AAAAAAAAABU/lqv1JlBYoaE/s400/jj.JPG

Cont. Unary Operators

```
public class arithmetic {

    public static void main(String[] args) {

        int i = 3;
        int j = -i;
        int k = -j;

        System.out.println("i = " +i);
        System.out.println("j = " + j);
        System.out.println("k = " + k);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac arithmetic.java

C:\Users\user\Documents\Java Programs>java arithmetic
i = 3
j = -3
k = 3
```

Cont. Unary Operators

```
public class arithmetic {

    public static void main(String[] args) {

        int i = 3, j = 6, k = 2;

        System.out.println("i = " +i);
        i++;
        System.out.println("i = " + i);
        ++i;
        System.out.println("i = " + i);

        System.out.print("\n");

        System.out.println("j = " + j);
        System.out.println("++j = " + ++j);
        System.out.println("j = " + j);

        System.out.print("\n");

        System.out.println("k = " + k);
        System.out.println("k++ = " + k++);
        System.out.println("k = " + k);

    }
}
```

Cont. Unary Operators

```
C:\Users\user\Documents\Java Programs>javac arithmetic.java

C:\Users\user\Documents\Java Programs>java arithmetic
i   = 3
++i = 4
i   = 5

j   = 6
++j = 7
j   = 7

k   = 2
k++ = 2
k   = 3
```

Math Class

Method	Explanation
max(arg1, arg2)	Returns the larger of the two arguments. Both arguments must be of the same type and can be int, long, float or double. The return argument is of the same type.
min(arg1, arg2)	Returns the smaller of the two arguments. Both arguments must be of the same type and can be int, long, float or double. The return argument is of the same type.
pow(arg1, arg2)	Returns the value of the first argument raised to the power of the second argument. All arguments are doubles.
random()	Returns a random number that is greater or equal to 0.0 but less than 1.0. Return value is a double.
round(argument)	Returns the integer that is closest to the argument. If the argument is a double, returns a long. If the argument is a float, returns an int.
sqrt(argument)	Returns the square root of the argument. All arguments are doubles.

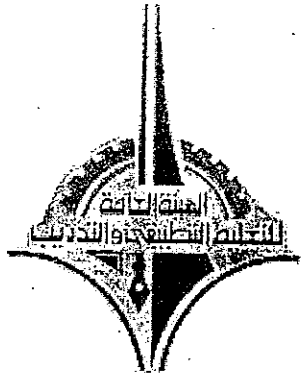
Cont. Math Class

```
public class mathFunctions {

    public static void main(String[] args) {

        double a = -6.4, b = 5;

        System.out.print("a = "+a);
        System.out.println(" and abs(a) = " + Math.abs(a));


        System.out.println("\nThe maximum value = " + Math.max(a, b));
        System.out.println("The minimum value = " + Math.min(a, b));

        System.out.println("\na to the power b = " + Math.pow(a, b));

        double r = Math.random();

        System.out.println("\nA random number between 0 and 1 :" + r);
        System.out.println("A random rounded number between 0 and 1 :" + Math.round(r));


        double f = Math.random() * 5;

        System.out.println("\nA random between 0 and 5 :" + f);
        System.out.println("A random rounded number between 0 and 5 :" + Math.round(f));

        System.out.println("\nThe square root of b = " + Math.sqrt(b));
    }
}
```

Cont. Math Class

```
C:\Users\user\Documents\Java Programs>java mathFunctions
a = -6.4 and abs(a) = 6.4

The maximum value = 5.0
The minimum value = -6.4

a to the power b = -10737.418240000003

A random number between 0 and 1 :0.94795322389644449
A random rounded number between 0 and 1 :1

A random between 0 and 5 :1.7145107206501549
A random rounded number between 0 and 5 :2

The square root of b = 2.23606797749979
```

```
C:\Users\user\Documents\Java Programs>java mathFunctions
a = -6.4 and abs(a) = 6.4

The maximum value = 5.0
The minimum value = -6.4

a to the power b = -10737.418240000003

A random number between 0 and 1 :0.26599568216357796
A random rounded number between 0 and 1 :0

A random between 0 and 5 :2.6998158498596747
A random rounded number between 0 and 5 :3

The square root of b = 2.23606797749979
```



Public Authority for Applied Education and Training

ENC 233
COMPUTER PROGRAMMING II

CHAPTER 4: MAKING CHOICES

Chapter 4 Objectives

- Learning about relational operators
- Understanding if, else and if-else statements
- Understanding Boolean expressions
- Using logical operators
- Comparing strings

Making Choices

- Statements that make a decision
- if statements executes a statement or a block of statements only if the test condition turns out to be true
- if statements rely on the use of boolean expressions that result in a simple true or false result

Boolean Expressions

- A Boolean expression when evaluated returns a Boolean value – true or false.
- Basic Boolean expression has this form:
  - *expression* relational-operator *expression*

Operator	Use	Description
>	op1 > op2	op1 is greater than op2
>=	op1 >= op2	op1 is greater than or equal to op2
<	op1 < op2	op1 is less than op2
<=	op1 <= op2	op1 is less than or equal to op2
==	op1 == op2	op1 and op2 are equal
!=	op1 != op2	op1 and op2 are not equal

http://1.bp.blogspot.com/-qQSRZ54elGo/UMVvP5MOicI/AAAAAAAAAD4/9Enzzh8Qik4/s400/ja.JPG

Using if Statements

- Syntax

```
if (condition)
  {
  code to be executed if condition is true
  }
```

http://www.w3schools.com/js/js_if_else.asp

Using if ... else Statements

- Syntax

```
if (condition)
  {
  code to be executed if condition is true
  }
else
  {
  code to be executed if condition is not true
  }
```

http://www.w3schools.com/js/js_if_else.asp

Using if ... else ... if Statements

- Syntax

```
if (condition1)
  {
  code to be executed if condition1 is true
  }
else if (condition2)
  {
  code to be executed if condition2 is true
  }
else
  {
  code to be executed if neither condition1 nor condition2 is true
  }
```

http://www.w3schools.com/js/js_if_else.asp

Example 1

```
public class conditions {

    public static void main(String[] args) {
        int number = 3;

        int remainder = number % 2;

        if (remainder == 0){
            System.out.println(number + " is an even number");
        }
        else
            System.out.println(number + " is an odd number");
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac conditions.java

C:\Users\user\Documents\Java Programs>java conditions
3 is an odd number
```

Example 2

- Ask the user to enter her test score
- If her test score is greater or equal to 90, her grade is A
- If her test score is greater or equal to 80 and less than 90, her grade is B
- If her test score is greater or equal to 70 and less than 80, her grade is C
- If her test score is greater or equal to 60 and less than 70, her grade is D
- If her grade is less than 60 her grade is F
- Display the letter grade in a message box

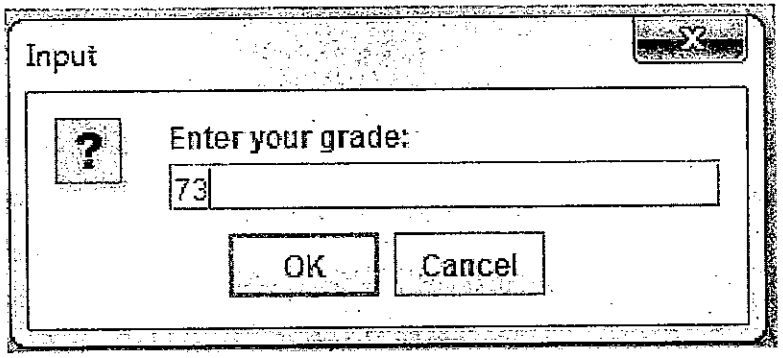
Cont. Example 2

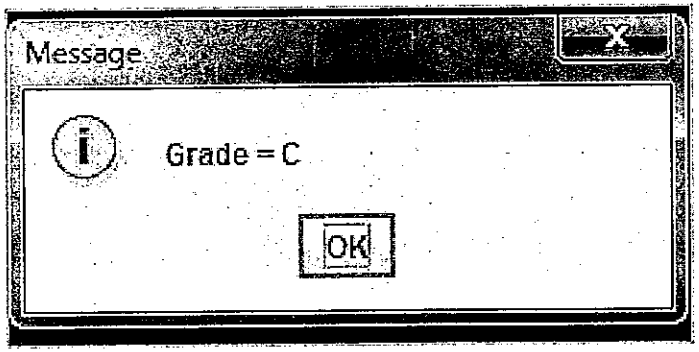
- Note that you need to convert a String to an integer, to do that use the following:
  - Integer.parseInt(*StringToBeConvertedToNumber*);

Cont. Example 2

```
import javax.swing.JOptionPane;

public class conditions {

    public static void main(String[] args) {

        String testscoreString = JOptionPane.showInputDialog("Enter your grade:");
        char grade;

        int testscore = Integer.parseInt (testscoreString);

        if (testscore >= 90) {
            grade = 'A';
        } else if (testscore >= 80) {
            grade = 'B';
        } else if (testscore >= 70) {
            grade = 'C';
        } else if (testscore >= 60) {
            grade = 'D';
        } else {
            grade = 'F';
        }
        JOptionPane.showMessageDialog(null, "Grade = " + grade);
    }
}
```

Cont. Example 2





Logical Operators

- An operator that returns a Boolean result that is based on the Boolean result of one or more other expressions.

Operator	Use	Description
`&&`	`op1 && op2`	Returns `true` if `op1` and `op2` are both `true`; conditionally evaluates `op2`
`||`	`op1 || op2`	Returns `true` if either `op1` or `op2` is `true`; conditionally evaluates `op2`

http://www.vanbellenet.be/programming/java/s_operators/images/conditional_operators.png

Comparing Strings

- The == operator does not work when comparing strings
- Suppose you want to know if a string variable named answer contains the value "Yes".

```
if(answer == "Yes")
    System.out.println("The answer is Yes");
```

- This is incorrect! Below is the proper string comparison.

```
if(answer.equals("Yes"))
    System.out.println("The answer is Yes");
```

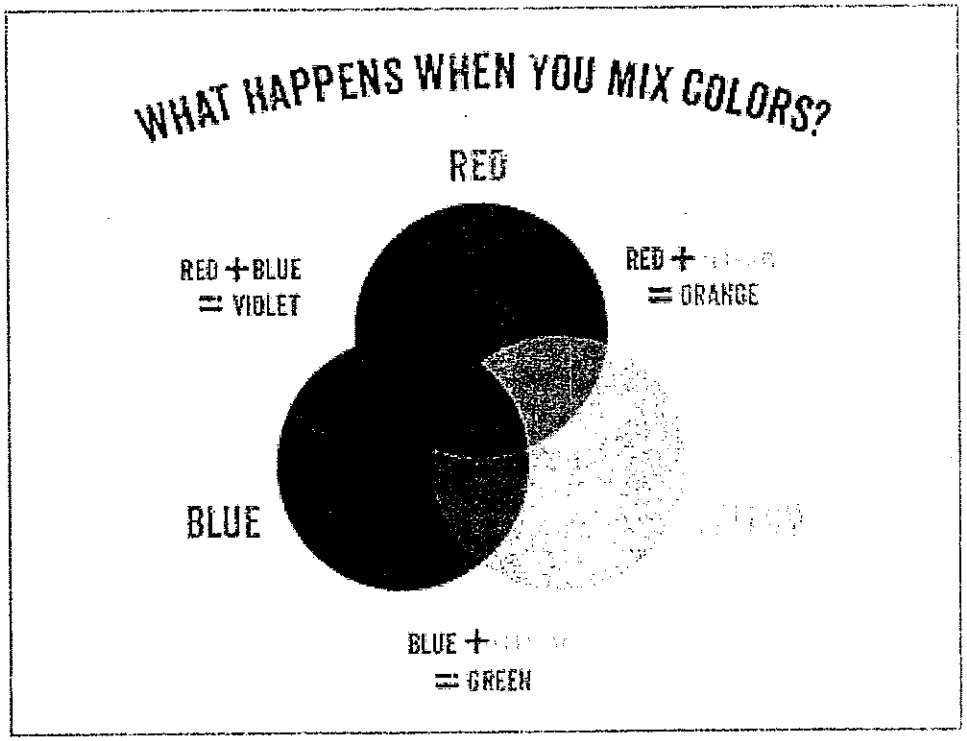
Cont. Comparing Strings

- Suppose that the user entered "yes" which will result in false when comparing with "Yes".
- Sometimes you want to ignore the letter case

```
if(answer.equalsIgnoreCase("yes")
    System.out.println("The answer is Yes");
```

Example 3

- Write a program that describes what happens when you mix each two of the three primary colours

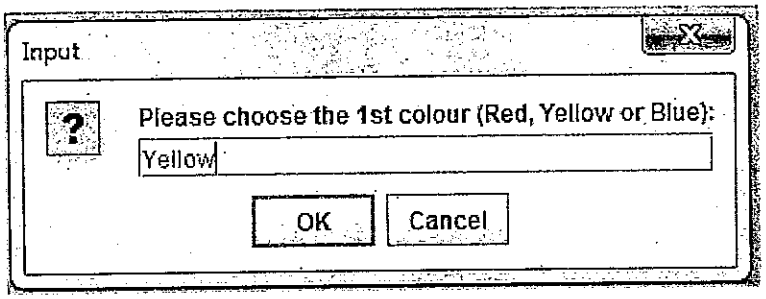


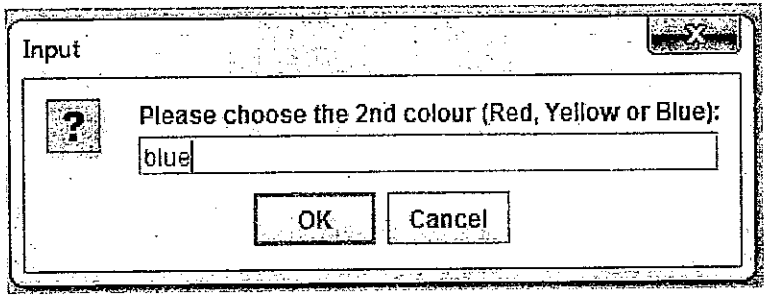
http://www.mrprintables.com/images/mix-primary-colors-chart.jpg

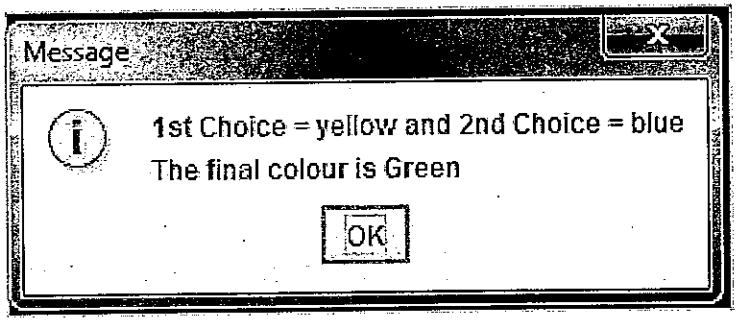
Cont. Example 3

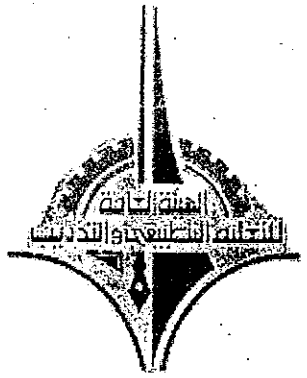
```
import javax.swing.JOptionPane;

public class conditions {

    public static void main(String[] args) {

        String choice1 = JOptionPane.showInputDialog("Please choose the 1st colour (Red, Yellow or Blue):");
        String choice2 = JOptionPane.showInputDialog("Please choose the 2nd colour (Red, Yellow or Blue):");

        String finalMix;

        if ( (choice1.equalsIgnoreCase("Red") && choice2.equalsIgnoreCase("Yellow")) || (choice1.equalsIgnoreCase("Yellow") && choice2.equalsIgnoreCase("Red")) ) {
            finalMix = "Orange";
        } else if ( (choice1.equalsIgnoreCase("Red") && choice2.equalsIgnoreCase("Blue")) || (choice1.equalsIgnoreCase("Blue") && choice2.equalsIgnoreCase("Red")) ){
            finalMix = "Violet";
        } else if ( (choice1.equalsIgnoreCase("Yellow") && choice2.equalsIgnoreCase("Blue")) || (choice1.equalsIgnoreCase("Blue") && choice2.equalsIgnoreCase("Yellow")) ){
            finalMix = "Green";
        } else {
            finalMix = "unknown!";
        }

        JOptionPane.showMessageDialog(null, "1st Choice = "+choice1+ " and 2nd Choice = "+choice2+"\nThe final colour is " + finalMix);

    }
}
```

Cont. Example 3









Public Authority for Applied Education and Training

ENC 233
COMPUTER PROGRAMMING II

CHAPTER 5: LOOPS

Chapter 5 Objectives

- Learning about looping a program
- Learning about different types of loops

Using Loops

- Executing same statements more than once!
- Loops rely on conditional expressions to tell them when to stop looping
- Without proper conditional expressions, loops will go on forever

While Loops

- Begins with evaluating the condition.
- If condition true, statements are executed.
- If condition false, statements are not executed and while loop ends.

```
while (condition)
  {
  code block to be executed
  }
```

http://www.w3schools.com/js/js_loop_while.asp

Cont. While Loops

- Use while loop to print the even numbers from 2 through 20 on the console.

Cont. While Loops

```
public class evenCounter {

    public static void main(String[] args) {

        int number = 2;
        while(number <= 20){
            System.out.print(number + " ");
                number = number + 2;
        }
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac evenCounter.java

C:\Users\user\Documents\Java Programs>java evenCounter
2 4 6 8 10 12 14 16 18 20
```

Breaking Out of a Loop

- Sometime you need to exit a loop.
- When a *break* statement is executed, the loop ends immediately.
- Modify the previous even counter to exit when the number reaches 12

Cont . Breaking Out of a Loop

```
public class evenCounter {

    public static void main(String[] args) {

        int number = 2;
        while(number <= 20){

            if (number == 12)
                break;

            System.out.print(number + " ");
                number = number + 2;
        }
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac evenCounter.java

C:\Users\user\Documents\Java Programs>java evenCounter
2 4 6 8 10
```

Do-While Loops

- The condition that stops the loop is tested after the statement is executed at least once.

```
do
   {
   code block to be executed
   }
while (condition);
```

http://www.w3schools.com/js/js_loop_while.asp

Cont. Do-While Loops

```
public class evenCounter {

    public static void main(String[] args) {

        int number = 2;
        do{
            System.out.print(number + " ");
            number = number + 2;
        }while(number <= 20);
    }
}
```

```
C:\Users\user\Documents\Java Programs>javac evenCounter.java

C:\Users\user\Documents\Java Programs>java evenCounter
2 4 6 8 10 12 14 16 18 20
```

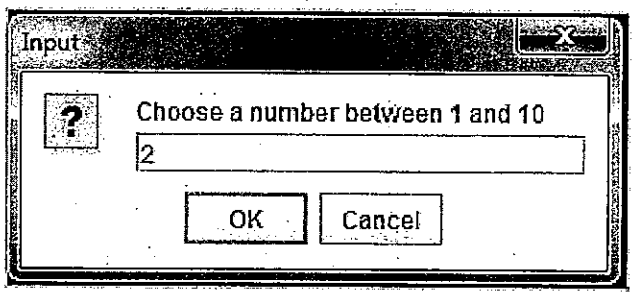
Example 1

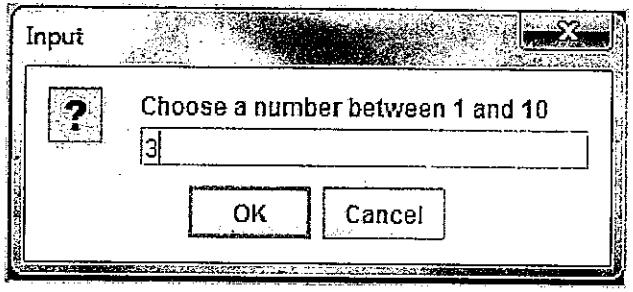
- Ask the user to enter a number between 1 and 10.
- If the number is between 1 and 10, print the number on the console.
- Else ask the user again to enter a number between 1 and 10.
- If the user enters -1 exit the loop and print "Thank you!" on the console.

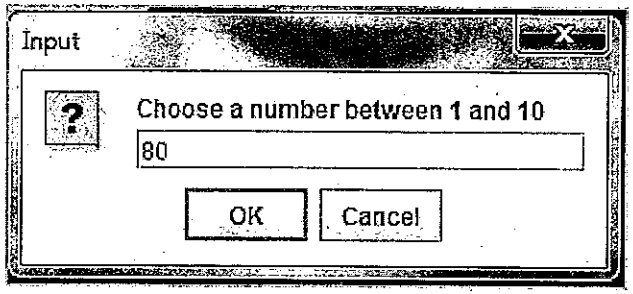
Cont. Example 1

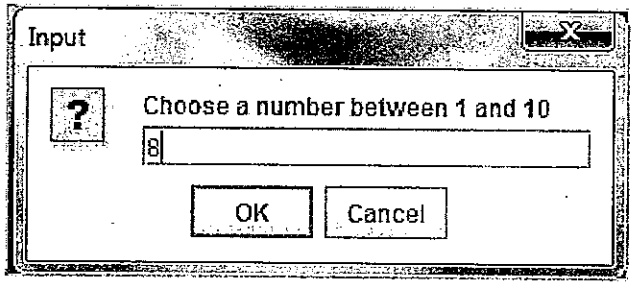
```
import javax.swing.JOptionPane;

public static void main(String[] args) {
    int number;
    do{
        number = Integer.parseInt(JOptionPane.showInputDialog("Choose a number between 1 and 10"));

        if (number <= 10 && number >= 1)
            System.out.println("The number you have chosen = " + number);
    }while(number != -1);

    System.out.println("Thank you!");
}
```

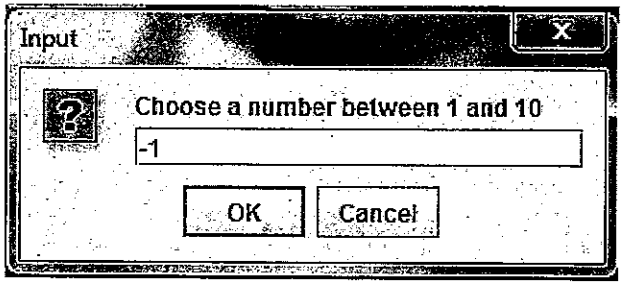
Cont. Example 1












```
C:\Users\user\Documents\Java Programs>javac doLoop.java

C:\Users\user\Documents\Java Programs>java doLoop
The number you have chosen = 2
The number you have chosen = 3
The number you have chosen = 8
Thank you!
```

For Loops

- The *initialization* expression initializes the loop; it is executed once before loop begins.
- When the *termination* expression evaluates to false, the loop terminates.
- The *increment* expression is invoked after each iteration through the loop; it is perfectly acceptable for this expression to increment or decrement a value

```
for (initialization; termination; increment) {
        statement(s)
}
```

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html

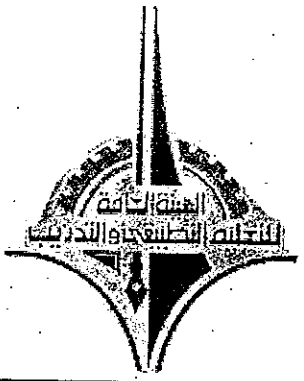
Cont. For Loop

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

The output of this program is:

```
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
```

http://docs.oracle.com/javase/tutorial/java/nutsandbolts/for.html



Public Authority for Applied Education and Training

ENC 233
COMPUTER PROGRAMMING II

CHAPTER 6: ARRAYS

Chapter 6 Objectives

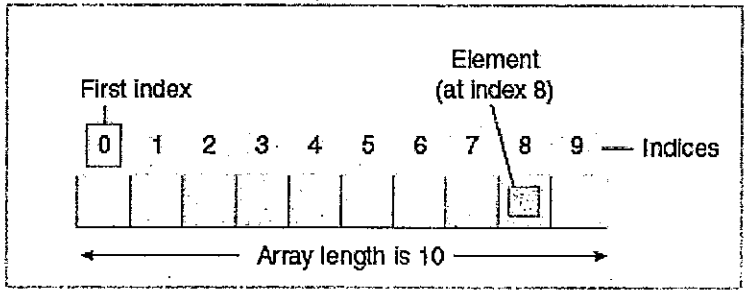
- Learning about one dimensional arrays

Arrays

- An array is a set of variables
- Each item in an array is called an element
- Elements can be accessed using a single variable with an index numbers
- An array has a type
  - int array contains int values
  - String array contains String values

Cont. Arrays

- In an array index numbers start with zero for the first element
- Array has a fixed length when it is created
  - An array with 10 elements, uses index values from 0 to 9
- You can not change the length of an array after you create the array
- x.length returns the length of the array x



http://docs.oracle.com/javase/tutorial/figures/java/objects-tenElementArray.gif

Creating Arrays

- Create an array named days of length 7
- Save the length of the array in a variable called size

```
String[] days = new String[7];
int size = days.length;
```

Cont. Creating Arrays

- Initialize the days array with the name of the week days

```
String[] days = new String[7];

days[0] = "Sunday";
days[1] = "Monday";
days[2] = "Tuesday";
days[3] = "Wednesday";
days[4] = "Thursday";
days[5] = "Friday";
days[6] = "Saturday";
```

```
String[] days = { "Sunday", "Monday", "Tuesday",
"Wednesday", "Thursday", "Friday", "Saturday"};
```

Example 1

- Initialize an array called days with the name of the week days.
- Print out the number of days using the length field.
- Using a loop print out all the week days on the console.

Cont. Example 1

```
public class arrays {

    public static void main(String[] args) {

        String[] days = { "Sunday", "Monday", "Tuesday",
        "Wednesday", "Thursday", "Friday", "Saturday"};

        int i, size;

        size = days.length;

        System.out.print("There are " + size + " days in the week\n\n");

        for (i = 0; i < size; i++)
            System.out.println(days[i]);

    }
}
```

```
C:\Users\user\Documents\Java Programs>javac arrays.java

C:\Users\user\Documents\Java Programs>java arrays
There are 7 days in the week

Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
```

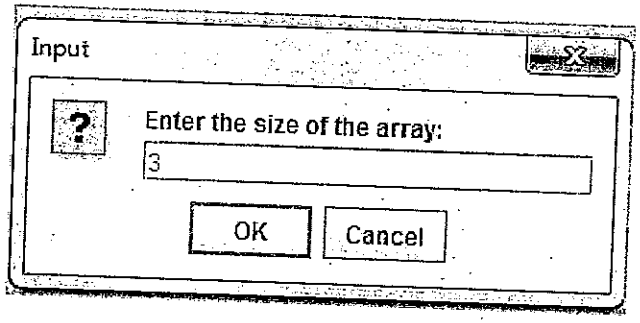
Example 2

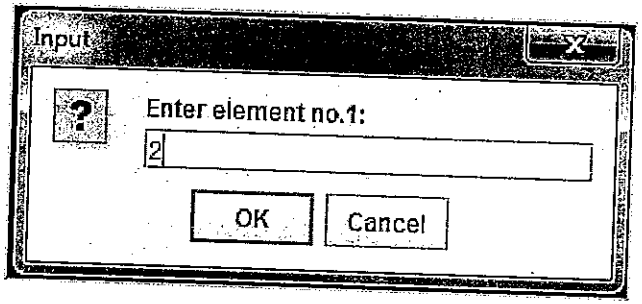
- Ask the user for the number of elements she wants to enter
- Create an array of type int with the size that the user entered
- Ask the user to enter that many elements each of type of int
- Once all int values are entered, print the values in the array on the console

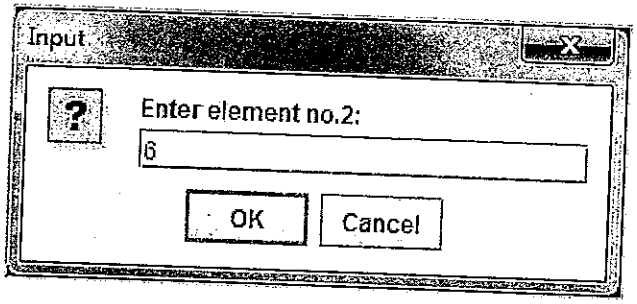
Cont. Example 2

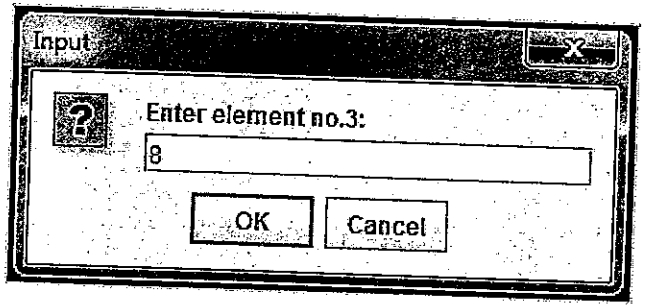
```
import javax.swing.JOptionPane;

public class arrays {

    public static void main(String[] args) {

        int size = Integer.parseInt(JOptionPane.showInputDialog("Enter the size of the array: "));

        int[] numbers = new int[size];

        int i, element;

        for (i = 0; i < size; i++){
            element = Integer.parseInt(JOptionPane.showInputDialog("Enter element no."+ (i+1) +": "));
            numbers[i] = element;
        }

         for (i = 0; i < size; i++)
            System.out.println(numbers[i] + "\n");
    }
}
```

Cont. Example 2










```
C:\Users\user\Documents\Java Programs>java arrays
2

6

8
```

The String class has many methods that are designed to you let you find out information about a string. Below are the most useful methods.

Method	Description
int length()	Returns the length of this string.
char charAt(int)	Returns the char value at the specified index.
boolean contains(String str)	Returns true if this string contains the parameter value.
int indexOf(Char ch)	Returns the index within this string of the first occurrence of the specified character.
int indexOf(String str)	Returns the index within this string of the first occurrence of the specified substring.
String replace(char oldChar, char newChar)	Returns a new string resulting from replacing all occurrences of oldChar in this string with newChar.
String replaceAll(String old, String new)	Returns a new string that's based on the original string, but with every occurrence of the first string replaced by the second parameter.
String replaceFirst(String old, String new)	Returns a new string that's based on the original string, but with the first occurrence of the first string replaced by the second parameter.
String[] split(String)	Splits the string into an array of strings, using the string parameter as a pattern to determine where to split the strings.
String toLowerCase()	Converts all of the characters in this String to lower case
String toUpperCase()	Converts all of the characters in this String to upper case
String trim()	Returns a copy of the string, with leading and trailing whitespace omitted.

Programming II – String Class Methods

```
String s = "A wonderful day for a picnic!";
int len = s.length();
System.out.println("The length of the string is "+len);
```

1

2

```
String s = "A wonderful day for a picnic!";
char c = s.charAt(4);
System.out.println("The character at position 4 is "+c);
```

1

2

```
String s1 = "A wonderful day for a picnic!";
String s2  = "pic";

System.out.println("s1 contains s2: "+ s1.contains(s2));
System.out.println("s1 contains x letter: "+ s1.contains("x"));
System.out.println("s1 contains \"y f\" string: "+ s1.contains("y f"));
```

1

2

3

```
String s1 = "A wonderful day for a picnic!";
String s2  = "pic";

System.out.println("s1 has "+s2+" at index: "+ s1.indexOf(s2));
System.out.println("s1 contains x letter at index: "+ s1.indexOf("x"));
System.out.println("s1 contains o letter at index: "+ s1.indexOf('o'));
```

1

2

3

```
String s1 = "A wonderful day for a picnic!";
String s2 = s1.replace('o', '*');

System.out.println(s2);
```

1

2

```
String s1 = "A wonderful day for a wonderful picnic!";
String s2 = s1.replaceAll("wonderful", "nice");
String s3 = s1.replaceFirst("wonderful", "nice");

System.out.println(s2);
System.out.println(s3);
```

1

2

3

```
String s1 = "Hello, my name is Farah, I am a college student.";

String[] words = s1.split(",");

System.out.println("The string before splitting:" + s1);

System.out.println("The string after splitting:");

for(int i = 0; i < words.length; i++)
        System.out.println(words[i]);
```

1

2

3

4

5

6

Programming II – String Class Methods

```
String s1 = "Hello!";

System.out.println(s1.toLowerCase());
System.out.println(s1.toUpperCase());
```

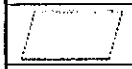
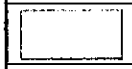
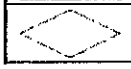
```
1
2
3
```

```
String s1 = "    Hello   There...     ";
String s2 = "My name is Zayan.";

System.out.print("Before trimming s1:");
System.out.print(s1);
System.out.println(s2);

System.out.print("After trimming s1:");
System.out.print(s1.trim());
System.out.println(s2);
```

```
1
2
3
4
5
6
```

Assignment 1

Note: This assignment is for topics covered in Chapter 1. You do NOT have to submit it.

Symbol	Purpose	Description
	Flow line	Used to indicate the flow of logic by connecting symbols.
	Terminal(Stop/Start)	Used to represent start and end of flowchart.
	Input/Output	Used for input and output operation.
	Processing	Used for arithmetic operations and data-manipulations.
	Desicion	Used to represent the operation in which there are two alternatives, true and false.

1. Draw a flowchart to ask a user to enter their first name and display it on the screen.
2. Draw a flowchart to add numbers 0 to 5 and store the result in variable Sum.
3. Draw a flowchart to add all the odd numbers between 1 and 100 and then displays the sum.
4. Draw a flowchart to ask the user to enter one of the following choices (1, 2, 3 or -1)
   i) If the choice is 1, display "You chose 1" and display the choices again.
   ii) If the choice is 2, display "You chose 2" and display the choices again.
   iii) If the choice is 3, display "You chose 3" and display the choices again.
   iv) If the choice is -1, display "Thank You" and exit the program.

5. Explain the difference between Java Applets and Servlets.
6. List three features of the Java programming language and briefly explain each.
7. What are the five phases the Java program goes through? Briefly explain each.
8. What are the two characteristics of an Object.

Assignment 2

Note: This assignment is for topics covered in Chapter 2. You do NOT have to submit it.

1. What is the output of the following statements:

```
System.out.println("The sky is \"blue\".");
```

```
int i = 3;
int j = -i;
int k = -j;

System.out.println("k = " + k);
```

```
String h = "Hello,";
String w = "World!";

System.out.print(h + "\n" + w);
```

```
System.out.print("My name is:");
System.out.println("Fatma");
System.out.println("I am \t 20 years old!");
System.out.print("I am an Architect");
```

2. Write a program that asks a user to enter their first name and displays it on the screen.

3. Write a program that calculates the average of the following numbers (3, 7, 98, 45, 10) and then displays the average on the screen.

4. Write a program that asks a user to enter their first name and then asks where they were born and display the following in a message box "Hello <first name>. You were born in <place of birth>".

Assignment 3

Note: This assignment is for topics covered in Chapter 3. You do NOT have to submit it.

Q1: What is the output of the following statements:

```
int a = 3;
int b = 4;

System.out.println(a + b);
System.out.println(" " + a + b + " ");
```

Q2: Write four different Java statements to add 1 to integer variable x

Q3:What is the value of the following variables. Assume at the start, all variables have value 5.

a. product = x++;
b. sum +=1;
c. sub = --x;

Assignment 4

Note: This assignment is for topics covered in Chapter 4. You do NOT have to submit it.

1. What is the output of the following statements:

```
int testscore = 76;
char grade;

if (testscore >= 90) {
        grade = 'A';
} else if (testscore >= 80) {
        grade = 'B';
} else if (testscore >= 70) {
        grade = 'C';
} else if (testscore >= 60) {
        grade = 'D';
} else {
        grade = 'F';
}
System.out.println("Grade = " + grade);
```

```
int testscore = 83;
char grade;

if (testscore >= 90) {
        grade = 'A';
}
 if (testscore >= 80) {
        grade = 'B';
}
if (testscore >= 70) {
        grade = 'C';
}
if (testscore >= 60) {
        grade = 'D';
} else {
        grade = 'F';
}
System.out.println("Grade = " + grade);
```

2. What is the output of the following statements if:
   a. x = 1, y = 0, z = -2;
   b. x = -1, y = 0, z = -2;
   c. x = -1, y = 2, z = -2;
   d. x = 2, y = 2, z = 3;

```
if (x > y && x > z){
        System.out.println("First");
        if (x != (y+1))
                System.out.println("Nested 1");
        else
                System.out.println("Nested 2");
}else if (x < y || x > z){
        System.out.println("Second");
        if (x == 1)
                System.out.println("Nested 3");
        else if(y == 0)
                System.out.println("Nested 4");
}else if (y < z){
        System.out.println("Third");
}
```

3. Write a program that asks the user to enter float numbers and calculates the average. The out should be in a message box, "The average of <num1> <num2> <num3> <num4> <num5> is <average>".
4. Write a program that asks the user to enter their birth year
   a. If they are above 18, ask them for their first name and then display "Hello! <first name>"
   b. If they are under 18, display a message box "You are under 18 years"
5. Write a program that calculates the tax due on a saving account. As the user for how much they have in their saving account. The tax is 10%.
6. Write a program that asks the user for which year of college they are in and their gender.
   a. If they are in year 1 and are females, they go to building 1
   b. If they are in year 1 and are males, they go to building 2
   c. If they are in year 2 or year 3 and are females, they go to building 3
   d. If they are in year 2 and are males, they go to building 4
   e. If they are in year 3 and are males, they go to building 3
   f. If they are in year 4 and are either females or males, they go to building 5

Assignment 5

Note: This assignment is for topics covered in Chapter 3 and Chapter 4. You do NOT have to submit it.

1. What is the output of the following statements:

```
int testscore = 95;
char grade;

if (testscore >= 90)
        grade = 'A';
if (testscore >= 80)
        grade = 'B';
if (testscore >= 70)
        grade = 'C';
if (testscore >= 60)
        grade = 'D';
else
        grade = 'F';

System.out.println("Grade = " + grade);
```

```
int j = -7;

System.out.println(j++);
System.out.println(++j);
```

```
int i = 2, k = 1;

System.out.println(k++);
System.out.println(--i);
System.out.println(i--);

System.out.print("The minimum of "+i+" and "+k+" is\t:"+Math.min(i, k));
```

```
int i = 3, j = -1, k = 1;

if(i > 5 && j < 0 || k == 2)
        System.out.printf("Statement is true!");
else
        System.out.printf("Statement is false!");
```

2. Write a program that generates three random integers (0 to 100) and then prints the smallest and largest of these three numbers. The output should be as follows:

   Number 1 =
   Number 2 =
   Number 3 =
   The larger number is
   The smallest number is

3. Write a program that asks the user if they would like to use which measurements, KGs and meters or Pounds and feet. If the user chooses the former, the program asks the user to input the weight in kilograms and the height in meters and prints out the weight in pounds and height in feet. If the user chooses the latter, the program asks the user to input the weight in pounds and the height in feet and prints out the weight in kilograms and height in meters.

   Pound = 2.5 * kilogram
   Feet = 3.2 * meter

4. Write a program that inputs the weight in kilograms and the height in meters. The program then calculates the Body Mass Index (BMI)

$$BMI = \frac{weight\ (kg)}{height\ (m) * height(m)}$$

   - Depending on the BMI value, the program should print the following messages:
     a. If the BMI is less than or equal to 18.5 print "You are underweight"
     b. If the BMI is greater than 18.5 and less than or equal to 25 print "You are normal"
     c. If the BMI is greater than 25 print "You are overweight"

5. Write a program that asks the user to enter the mass and volume of an object and determine whether the object will float or sink.

   Density = Mass/Volume

   - If density is less than 1 then the object floats otherwise it sinks

Assignment 6

Note: This assignment is for topics covered in Chapter 5. You do NOT have to submit it.

1. What is the output of the following statements:

```
for (int x =1;x<5;x++){
        for  (int y =1;y<5;y++)
                System.out.print(x + "-" + y + "  ");
        System.out.println();
}
```

```
boolean continue_looping = true;
int i=0;
while(continue_looping){
        System.out.print("Loop" + i++ +":");

        if (i == 4)
                continue_looping = false;

        System.out.println("continue_looping value = "+continue_looping);
}
```

2. Write a program that keeps counting even numbers as long as the user answers Yes to "Do you want to keep counting?" question. If the answer is No, stop counting and display "Thank you!" and exit the program.
   The output should be as following:

   2
   Do you want to keep counting? (Y or N): Y
   4
   Do you want to keep counting? (Y or N): Y
   6
   Do you want to keep counting? (Y or N): n
   Thank you!

3. Write a program that counts backwards from a number that the user has entered to 1. Print out all the numbers.

4. Write a program that plays a guessing game. Generate a random integer between 1 and 10. Then ask the user to guess a number between 1 and 10. If the guess is correct, print "You guessed the number correctly!". If the guess is incorrect, "You guessed the number incorrectly!". Then ask the user if they want to play again or not. If the answer is yes start all over, if no display "Thank you!".

Assignment 7

Note: This assignment is for topics covered in Chapter 6. You do NOT have to submit it.

1. What is the output of the following statements:

```
int a = 3;
int b = 4;

System.out.println(a + b);
System.out.println(" " + a + b + " ");
```

```
double[] myList = {1.9, 2.9, 3.4, 3.5};

for (int i = 0; i < myList.length; i++)
        System.out.println(myList[i] + " ");

double total = 0;
for (int i = 0; i < myList.length; i++)
        total += myList[i];

System.out.println("Total is " + total);

double max = myList[0];

for (int i = 1; i < myList.length; i++)
        if (myList[i] > max) max = myList[i];

System.out.println("Max is " + max);
```

2. Write a program that asks the user to enter a word. The program will then print out the word for as many times as it has characters.
3. Write a program that creates an array and fills it with 10 random numbers between 1 and 20. Then it prints all the elements and prints out the average.
4. Write a programs that has two arrays, one for storing students names and the other for their final grades. Ask the user to enter the number of students. And then ask for the student's name and their final grade and store them in their appropriate arrays. One all data is entered print all the students names and grades.

Assignment 8

1. As the user to enter a sentence and the count the number of vowels (a, e, i, o, u) in the sentence and print them out.
2. As the user to enter a sentence and print out the number of spaces in the sentence.
3. Given a secret message try to decipher it, given the following code:

Secret Character	Original Character
&	h
!	a
*	space
-	m
$	o
^	s
+	i
(	e
)	l
=	n
%	g

Q1: Choose the correct answer:

1. Which will legally declare and initialize an array
   a. int [] myList = {"1", "2", "3"};
   b. int [] myList = (5, 8, 2);
   c. int myList [] [] = {4,9,7,0}
   d. int[] myList = {4, 3, 7};

2. Which one of the following will declare an array and initialize it with five numbers?
   a. int [5] array;
   b. int a [] = new int[5];
   c. int [] a = {23,22,21,20,19};
   d. Array a = new Array(5);
3. Java Applications begin execution at method:
   a. init
   b. paint
   c. print
   d. main

Q2: What is the output of the following statements:

```
int a = 3;
int b = 4;

System.out.println(a + b);
System.out.println(" " + a + b + " ");
```

Q3: Write four different Java statements to add 1 to integer variable x

Q4:What is the value of the following variables. Assume at the start, all variables have value 5.

a. product = x++;
b. sum +=1;
c. sub = --x;