Wright College + Chapter 13

Object-Oriented Programming (OOP)
Using Classes and Objects and (Revisit to Arrays and ArrayList)

CIS 144 Java Programming Language + Introduction to Computer Programming

"Hands-On" Mastering Computer Logic, Design and Programming Using Java Language

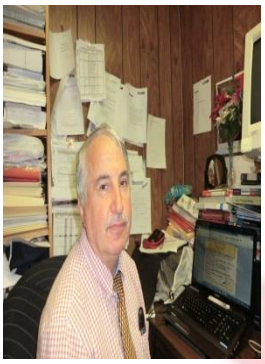
Written By:

Ogar Haji

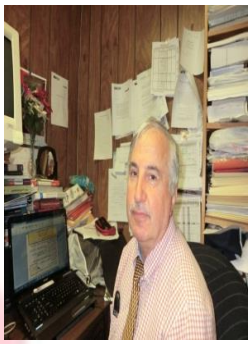
MS Computer Science

DePaul University ✚ Chicago, Illinois

Date Published: May 7, 2021

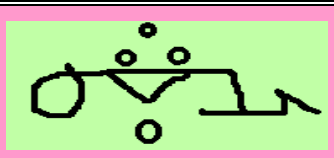


computer Text Books Published by the Author: Ogar Haji



The Following is a List of Computer Text Books Published by the Author: Ogar Haji. He has an MS Degree in Computer Science from DePaul University, Chicago, Illinois - USA. Mr. Ogar Haji has over 30 Years of teaching experience at: The College of Office Technology, Oakton College, Washington College, Truman College, Wright College, Triton College, ITT Technical Institute, Phoenix University and East+West University in Chicago Illinois.

1) **“Hands-On” Mastering Microsoft Windows 8.1 and 7**
2) **“Hands-On” Mastering Microsoft Excel 2016 and 2013**
3) **“Hands-On” Mastering Microsoft Word 2016 and 2013**
4) **“Hands-On” Mastering Microsoft Access 2013 and 2010**
5) **“Hands-On” Mastering Microsoft PowerPoint 2016 & 2013**
6) **“Hands-On” Mastering Microsoft Publisher 2013**
7) **“Hands-On” Mastering MS Visual Basic .Net Language**
8) **“Hands-On” Mastering Java Programming Language**
9) **“Hands-On” Mastering HTML5 & CSS3 Web Page Design**
10) **“Hands-On” Mastering JavaScript Programming Language**
11) **“Hands-On” Mastering Ruby Programming Language**
12) **“Hands-On” Mastering Python Programming Language**
13) **“Hands-On” Mastering QBasic Programming Language**
14) **“Hands-On” Mastering DOS (Disk Operating System)**
15) **“Hands-On” Mastering C# Programming Language**
16) **“Hands-On” Mastering Java Programming Language**



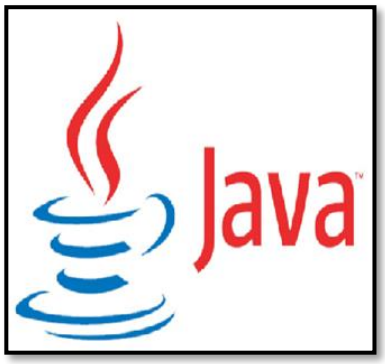
Ogar Haji (Computer Instructor)

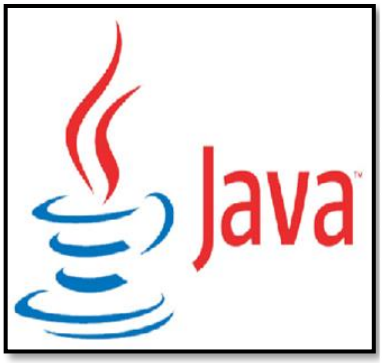
Harry S. Truman College

1145 West Wilson Avenue

Chicago, Illinois 60640

USA





“

Computer Labs Rules

1- **No Drinks, Food, Headphones** allowed in Computer labs. And Please **Turn Off the Cell Phones**.

2- When **Lecturing** is in progress, you are Not allowed to work on the computer. Please **Pay Attention** and **Take Notes**.

3- **Attendance and Punctuality** are very **important**. If you are absent, it is your responsibility to make up for the missing work and assignments. **Attendance will be taken daily.**

4- Students should have a **USB Flash Drive** and **Save Projects** to it.

5- **Practice makes perfect**. Please keep practicing the new features or steps repeatedly until the instructor tells you to stop.

6- You must **Concentrate** on what you are doing. **Talking is Not Allowed** in the computer Lab.

7- Please **Study** the Lessons in your **Java Handout and Text Book Daily** and **review your notes before class**. **There will be a Quiz Once a Week**.

8- Please **Check Mark ☑ the Lessons** in the Handout that you **have completed.**

9- You **must do All Java works**, **Assignments and Tests** located at the End of each Chapter on Time.

CIS144 Java Programming
Instructor:
Ogar Haji

Chapter 13

Object-Oriented Programming (OOP)
Using Classes and Objects
Create a Class and Instantiate an Object

You will learn the following in Chapter 13

- **What is OOP**
- **What is a Class in Visual C# Programming Language**
- **What is an Object in Visual C# Programming Language**
- **How to Create an Object from a Class using new keyword**
- **What are get and set Accessors**
- **Create an Employee Class Lab Exercise**
- **Create a TVProject OOP Class Lab Exercise**
- **Create Vending Machine Simulator Lab Exercise**
- **Using Two Forms and display the second form**
- **Create Car Information Form Lab Exercise**
- **Create BankAccount Class and Windows Form Project to Simulate a Bank Deposit and Withdrawal Project in Java language**
- **Do Lab Assignment 13 NewHadra Pizza Restaurant Project**
- **Do Chapter 13 Homework # 13**

Input/Output

Flowchart Symbols

Processing

+++ Review +++

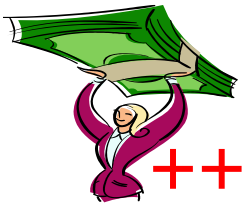
Lesson 260 Review : What are the Flowchart Symbols used in Java Language?

You should always **draw a Flowchart** when you Design, Code and Solve a problem in Java language.

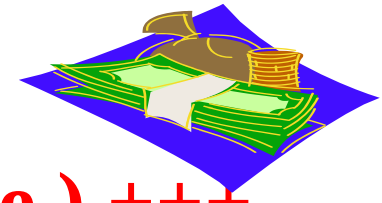
Before you Code a program in Java Language, you have to **Draw a Flowchart** to solve the problem of the program you want to code.

The following symbols are used with Java Programming Language:

Symbol	Symbol Name	Usage
Start or End	**Oval (Beginning and Terminal)** symbol	Use **Oval (Beginning and Terminal) Symbol** at the Beginning of the Flowchart and at the End of the Flowchart. Use with Start and End statements.
Print and Read	**Parallelogram (Input/Output)** Symbol	Use **Parallelogram (Input /Output)** or **I/O** symbol to Input Data, Read Input or Print Output
grossPay = hours * rate	**Rectangle** Symbol	Use the **Rectangle** Symbol for Calculating, Assigning Values
	**FlowLine** Symbol	Use **FlowLine** Symbols to show the Flow or Sequence of the flowchart.
if(hours>40)	**Diamond(Decision)** Symbol	Use **Diamond (Decision)** Symbol with the If or Select statements when deciding if Hours is > 40. The Result will be either True or False.
A	**Connector** Symbol	Use **Connector** Symbol to Connect the Flowchart rather than draw a long Arrow. Use A
calculateGrossPay()	**Function** or **Method (Predefined Process)** Symbol.	Use **Function** or **Method (Predefined Process)** Symbol to call another Function or Method that contains coding statements.



Calculate Gross Pay of Employees Project



+++ (Review Lab Exercise) +++

Lesson 261 Ex : How to Calculate Gross Pay of Employees Project?

Do the following 12 Must Steps to Design, Code and Solve a project using Java Language.

Do Steps 1 thru 7 in your Note Book or on Paper.

Step 1) Purpose of the Program: State what Program will do: (5 Points)

a) This Program will calculate Gross Pay of Employees.

b) It will ask the User to Enter Employee's Full Name:

c) It will ask the User to Enter Number of Hours Worked and

d) It will ask the User to Enter Hourly Rate.

e) The program will then calculate Gross Pay.

Gross Pay = Hours * Rate

f) Display the Gross Pay

Step 2) Input: You should know how the Input looks like: (5 Points)

Enter Employee's Full Name: Ogar Haji

Enter Hours Worked: 40

Enter Hourly Rate: 10

Step 3) Processing and Calculation: The program will process each record and Calculate Gross Pay: (5 Points)

Gross Pay = Hours * Rate

Step 4) Output: You should know how output should look like: (5 Points)

********** Calculate Gross Pay Project **********

The Employee Ogar Haji Gross Pay is = $ 400

********** End of Program **********

Step 5) Flowchart: Draw a Flowchart for Gross Pay program. (5 Points)

Start

1) Declare the Variables to be used in the project

2) Print out to Console "Title/Heading"

3) Prompt user "Enter Employee's Full Name: "

4) Read the fullName and store it in a variable

5) Prompt the user "Enter Hours Worked: "

6) Read hours worked and store it in a string

8) Prompt the user "Enter Hourly Rate: "

9) Read Hourly Rate and store it in a string

11) Calculate Gross Pay = Hours * Rate

12) Print out to Console "Your Gross Pay is = $",

13) Print out to Console "******* End of Program

End

Step 6) PseudoCode: Write a PseudoCode for the Program.(5 Points)

1) Declare the variables to be used in the program
2) Write to Console "The Title or Heading of the program "
3) Prompt the user "Enter Employee's Full Name: "
4) Read from Console fullName and store the string in a variable
5) Prompt the user "Enter Hours Worked: "
6) Read from console hours worked and store in a hours variable
7) Prompt the user "Enter Hourly Rate: "
8) Read from console the rate and store in a rate variable
9) Calculate Gross Pay = Hours * Rate
10) Write to Console "The Employee Gross Pay is = $", gross_pay
11) Write to Console "******* End of Program *******"

Step 7) Code the Program in Java by referencing the Flowchart or Pseudocode you designed above and Save it as CalculateGrossPay to USB.

a)Type the following Java code in NetBeans IDE.

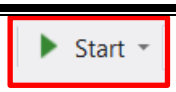
Code for the first part of "CalculateGrossPay" project follows:

```
/*
 **********************************************************************
   Purpose of the Project:
   a) This Interactive Project will prompt the user to enter his/her (First Name,
Last Name, and Hours Worked and  Rate) then it will read the text entered
and store it  in string variables. It Converts the string Hours to integer
numbers into integer and converts the Rate to Double and then Calculate the
Gross Pay.
   b) Project Name: CalculateGrossPay
   c) Date: Saturday, December 28, 2013
   d) Programmer: Instructor – Ogar Haji
 **********************************************************************
*/
```

```
package calculategrosspay;
// import the Classes needed in this project
import java.util.Scanner;
public class CalculateGrossPay {
   public  static void main(String[] args) {
        // 1) Declare the variables to be used in the project
        String firstName, lastName;
        int hours = 0;
        double rate = 0.0;
        double grossPay = 0.0;
```

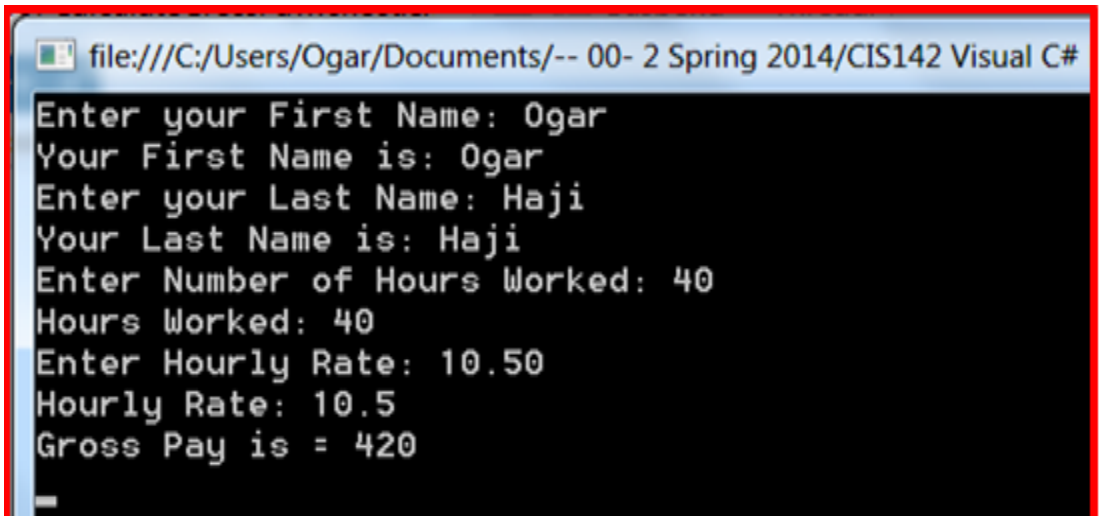
```
        // 2) Prompt the User to Enter his/her First Name
        System.out.print("Enter your First Name:  ");
        // 3) Read First Name from console and store it in a variable
        firstName = input.nextLine();
        // 4) Print First Name to the Console
        System.out.println ("Your First Name is: " + firstName);
        // 5) Prompt the User to Enter his/her Last Name
        System.out.print ("Enter your Last Name: ");
        // 6) Read his/her Last Name from console and store it in a variable
        lastName = input.nextLine();
        // 7) Print out Last Name to the Console
        System.out.println ("Your Last Name is: " + lastName);
        // 8) Prompt the User to Enter Hours Worked
        System.out.print ("Enter Number of Hours Worked: ");
        // 9) Read Hours Worked from Console and Store it in a variable
        hours = input.nextInt();
        // 10) Print out Hours Worked to the Console
        System.out.println ("Hours Worked: " + hours);
        // 11) Prompt the User to Enter Hourly Rate
        System.out.print ("Enter Hourly Rate: ");
```

```
        // 12) Read Hourly Rate from Console and Store it in a variable
        rate = input.nextDouble();
        // 13) Print out Hourly Rate to the Console
        System.out.println ("Hourly Rate: " + rate);
        // 14) Calculate Gross Pay
        grossPay = hours * rate;
        // 15) Print out grossPay to the console
        System.out.println ("Gross Pay is = " + grossPay);
    }
  }
}
```

Step 8) Click **Start**  button to Start Running the program

The following output appears on the Left side of the screen with the Input you entered and the correct calculated GrossPay $400.

If any Syntax Errors Found Do Next Step 9:



Step 9) Debug the Program: Debug or Correct any Syntax Errors until you have a clean Compiled program. (5 Points) (Clean compiled program means No Errors in the program).

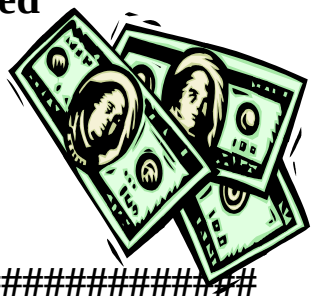
Step 10) Test the Program: Test the Program with Test Data. (5 Points)

Repeat Step 10) Test the program many Times and Test the Program again and again until All conditions are tested:

```
file:///C:/Users/Ogar/Documents/-- 00- 2 Spring 2014/CIS142 Visual C#
Enter your First Name: Mary
Your First Name is: Mary
Enter your Last Name: Smith
Your Last Name is: Smith
Enter Number of Hours Worked: 30
Hours Worked: 30
Enter Hourly Rate: 11.23
Hourly Rate: 11.23
Gross Pay is = 336.9
Mary Smith 30 11.23 336.9
Your Gross Pay is = $336.90
```

Step 11) Documentation (5 Points): You have to add more comments to the Program (like Comments about the Purpose of the Program, Your Name and the Date the Program was written.)

```
##################################################################
# Purpose of the Program:
#   a) This Program will calculate Gross Pay.
#   b) It will ask the User to Enter Employee's Full Name:
#   c) It will ask the User to Enter Number of Hours Worked
#   d) It will ask the User to Enter Hourly Rate.
#   e) The program will calculate Gross Pay.
#       Gross Pay = Hours * Rate
#   f) Display Gross Pay
##################################################################
```



Step 12) Print a Copy of Java Code along with screen printout of the Running program. Submit to your Instructor the Print Copy and the screen Printout (Snaps) along with the following: (Which you did on Paper)

Copy the Java Code and the Output of the program and Paste it in Microsoft Word program:

1) Purpose of the Program.
2) Input: how the Input looks like
3) Processing and Calculations

4) Output: how the Output will look like

5) Flowchart 6) Pseudocode

7) Java Code and

8) Print out copy of Java code and Output after running the program.

Submit the Programs on Time.

Remember Points will be deducted (20%) for Programs submitted Late.

Important Note:

1) Do Steps 1 thru 7 on Paper.

2) Then Get into NetBeans IDE

3) Type the Java code.

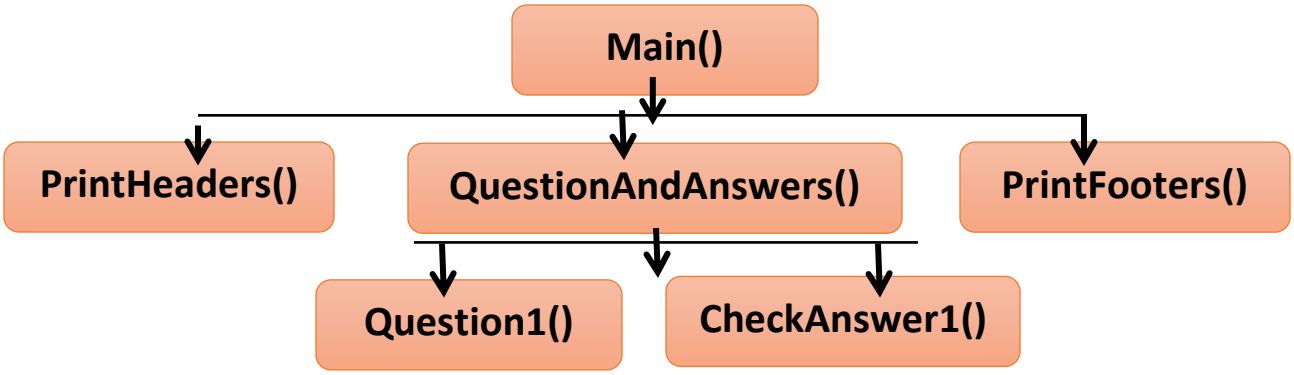
4) Save All the Files

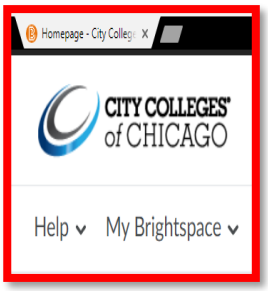
5) Run the Program and Test it with Test Data for All Conditions.

Instructor: Ogar Haji

Do Only 1 Modification at a Time

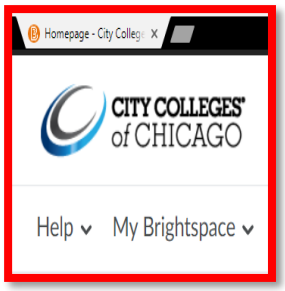
Use Structure Programming:





Upload Your Weekly Assignments: To Brightspace Correct Weekly Assignments Folder

+++ Review +++



Lesson 262 : How to Upload Your Weekly Assignments to Brightspace Correct Weekly Assignments Folder?

You have to Upload your Weekly Assignments and Homework to Brightspace correct Weekly Assignments Folder as following:

1) Copy the Java Code from NetBeans to Word document:

1. Copy the **Java Code from** the **NetBeans IDE** and **Paste it into** the **Microsoft Word Document.**

2. In NetBeans IDE, press **Ctrl+A (select All)** to **select All** the **Java code**.

3. Press **Ctrl+C (Copy)** to **Copy** the **selected Java code** into **computer memory RAM.**

2) Paste the Java Code into Microsoft Word:

1. Get into **Microsoft Word document then** press **Ctrl+V (paste) to Paste** the **copied Java code** from **memory** into **Word document.**

2. Press **Ctrl+Home (go to the Top of Document)** and **type** your **Full Name** at the **top of document followed by** the **Java File Name** in **size 20 and bold**.

3) Print the Screen of the Output of Java NetBeans:

1. Run the **Java project** and **make sure the program** is **running** with **correct output.**

2. Press **PrintScreen button** 

to **capture** the **output screen shot.**

4) Paste the Print Screen of Java output into Paint program:

1. Get into **Paint program** and press **Ctrl+V (paste) to Paste** the **screen shot** in **Paint** program**.**

2. Inside the **Paint program,** Click **Select** 

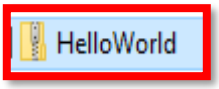
icon and then **Select only** the **Output** of the **Java project.**

3. Press **Ctrl+C (Copy)** to **Copy** the **selected output image.**

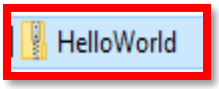
5) Get back into Microsoft Word program:

1. Go back to **Word Document,** press **Ctrl+End (End of Document)** to **go to End of document.**
2. In the **Word document,** press **Ctrl+V (Paste)** to **paste Java output** there.
3. Save the **Word Document** as the **Name of the Java project** and in this **example (Save File as HelloWorld project)**

6) To Compress or Zip the Java project:

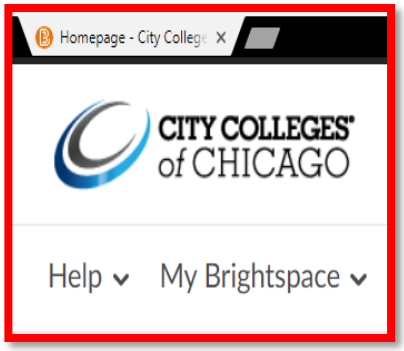


1. Right-Click on **Java project (HelloWorld)** that is **saved on your PC**.
2. Point to **Send to,** then **click on Compressed (Zipped) Folder** and you will have **another File** which is Compressed or Zipped.



7) To Upload the 2 Files to Brightspace:

1. Log on Brightspace with your **User Name** and **Password**
2. Click on your course **CIS 144 Java** to **Select it**.
3. Click on **Assignments**
4. Click on **Assignments**
5. Click on **Week 12 Assignments Folder**
6. Click on **"Add a File"** button
7. Click on **"My Computer"**
8. Click on **Upload** button



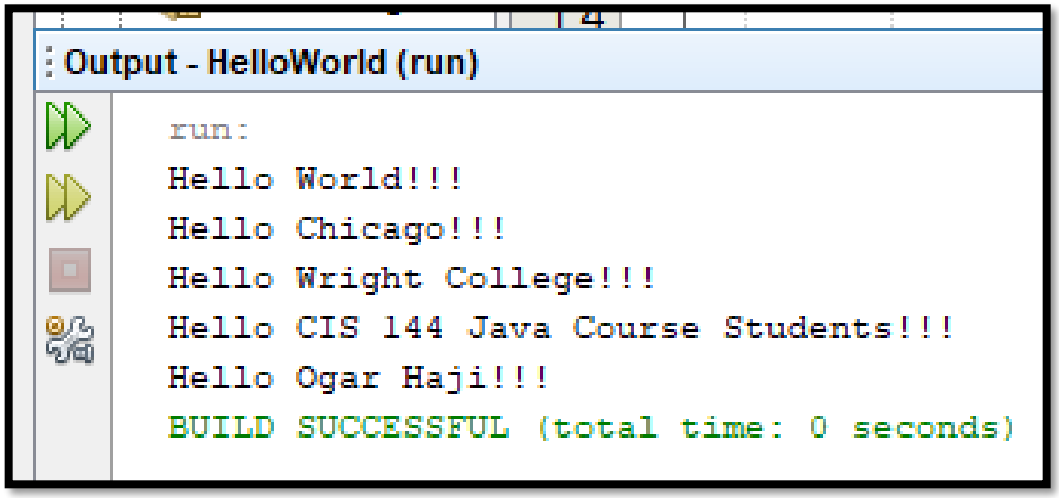
9. Go to the **location** where you **saved the Java project "HelloWorld".**
10. Click on the **File** or **Folder (HelloWorld)**
11. Click on **Add** button and the **File** or **Folder** will be **added** to the **Week 12 Assignments Folder.**

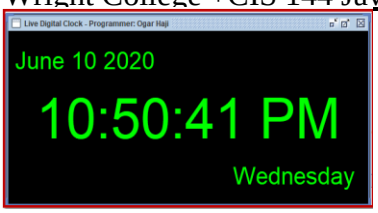
Note: Always **Upload** to **Brightspace 2 Files** of **same Java Project:**
1) The **Microsoft Word Document** of the **Java Project Code along with the Java Output Screen shots.**
2) The **Compressed** or **Zipped File** or **Folder** of **Java Project**.

The Java project "HelloWorld" code in Word Document along with the Output Screen Shots appear as following:

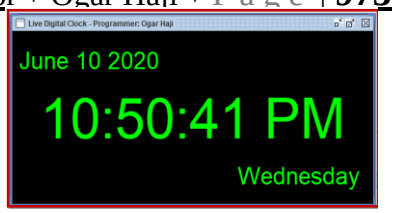
Ogar Haji
CIS 144 Java
Project Name: HelloWorld

```
/*
Project Name: HelloWorld
This Java project will print the message "Hello World" to screen
Programmer: Instructor + Ogar Haji (Type your Full Name)
Date:   June 01, 2017
*/

package helloworld;

public class HelloWorld {
   public static void main(String[] args) {
      // This project will print the Literal String "Hello World" to output screen

      System.out.println ("Hello World!!!");
      System.out.println ("Hello Chicago!!!");
      System.out.println ("Hello Wright College!!!");
      System.out.println ("Hello CIS 144 Java Course Students!!!");
      System.out.println ("Hello Ogar Haji!!!");
   }
}
```





Didplaying a Live Digital Clock



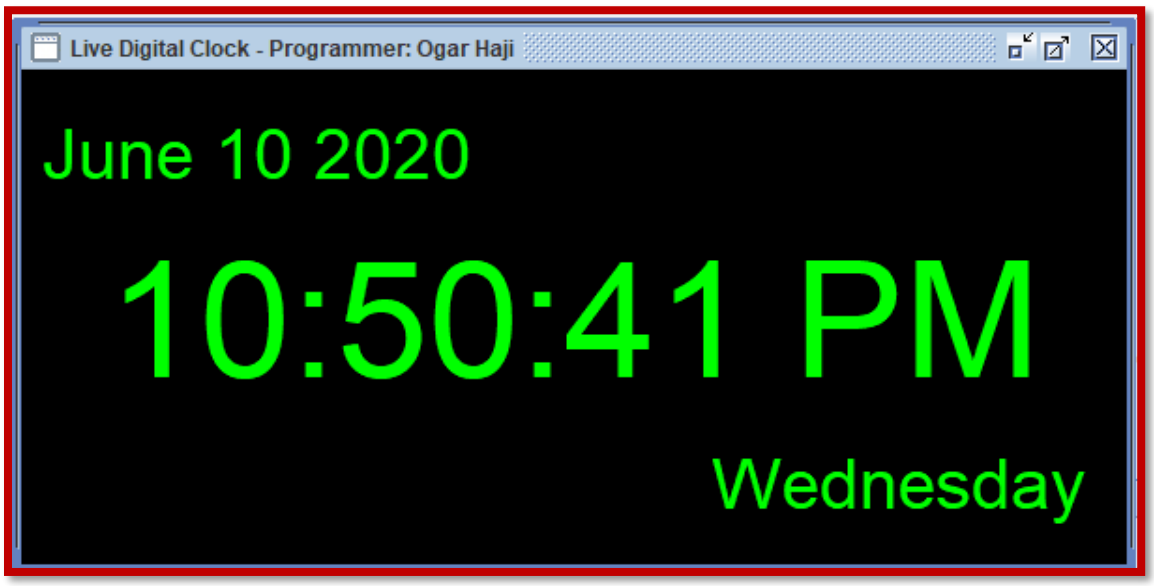
+++(Do Lab Exercise 1) 100 Points+++

Do Lab Exercise 1

Lesson 263 Ex : How to Display a Live Digital Clock in Java Language ?

Do the 12 Must Steps to Design, Code and Solve a project using Java Language.

This Java project will Display a Live Digital Clock.

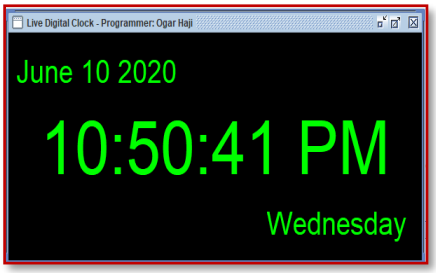


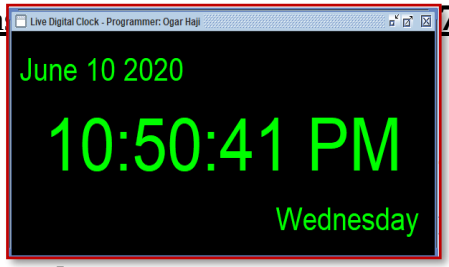
1) Launch **NetBeans IDE** program **Do Lab Exercise**

2) Type following **Java Lab exercise, Save** it as **DigitalClock.**

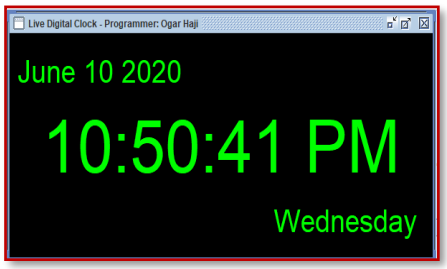
```
/*
   This project DigitalClock is a simulation of a live digital clock
   Project name: DigitalClock
   Programmer: Ogar Haji
*/
```

```
package digitalclock;

// Import the classes needed in the project
import java.awt.*;
import java.awt.event.*;
import javax.swing.*;
import java.util.*;
import java.text.*;
import javax.swing.Timer;

public class DigitalClock {

  public static void main(String[] arguments) {
    // 1) ClockLabel is an extension of the JLabel class that listens to events
    // from a Timer object to update itself with the current date & time.
    ClockLabel dateLabel = new ClockLabel("date");
    ClockLabel timeLabel = new ClockLabel("time");
    ClockLabel dayLabel = new ClockLabel("day");

    JFrame.setDefaultLookAndFeelDecorated(true);

    // 2) Create a new object f of the class JFrame
    JFrame frame = new JFrame();


   // 3) Set the Title of the Frame
    frame.setTitle ("Live Digital Clock - Programmer: Ogar Haji");

    // 4) Set bounds from Left then top
```

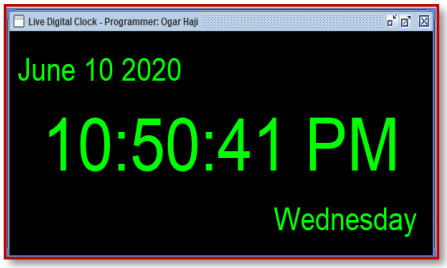




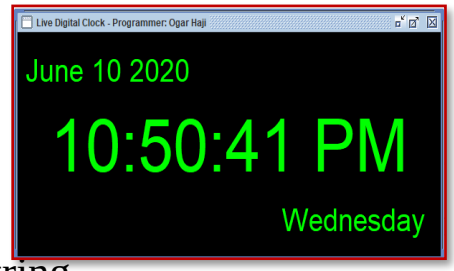

```
    frame.setBounds(400,200,200,200);

    // 5) Set the Size of the frame
    frame.setSize(600,300);

    // 6) Set the default Close of the Frame and the Exit on close
    frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);

    // 7) Set the frame Layout to GridLayout with 3 Rows and 1 column
    frame.setLayout(new GridLayout(3, 1));

    // 8) Add the dateLabel, timeLabel and dayLabel to the frame
    frame.add(dateLabel);
```



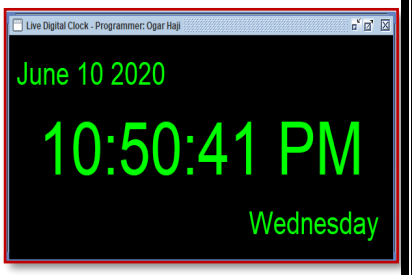

```
    frame.add(timeLabel);

    frame.add(dayLabel);

    // 9) Set the frame Background to Color.Black

    frame.getContentPane().setBackground(Color.BLACK);

    // 10) Display or show the JFrame
    frame.setVisible(true);
  }
}
```



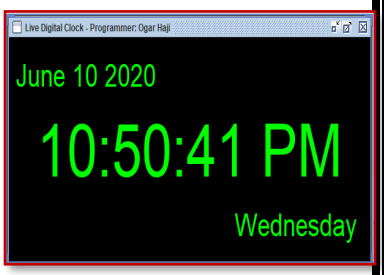

```
// Define the ClockLabel class and code it
class ClockLabel extends JLabel implements ActionListener {

  // 1) Declare variable of type String to store the (date, time or day)
  String dateTimeOrDay;

  SimpleDateFormat sdf;

  public ClockLabel(String dateTimeOrDay) {
```




```
this.dateTimeOrDay = dateTimeOrDay;

// Set Foreground Color to Color Green
setForeground(Color.green);

// use switch statement to check the dateTimeOrDay string
switch (dateTimeOrDay) {
  case "date" :   // if the string is “date”, then set the date format
     sdf = new SimpleDateFormat(" MMMM dd yyyy");
     setFont(new Font("sans-serif", Font.PLAIN, 38));
     setHorizontalAlignment(SwingConstants.LEFT);
     break;
```

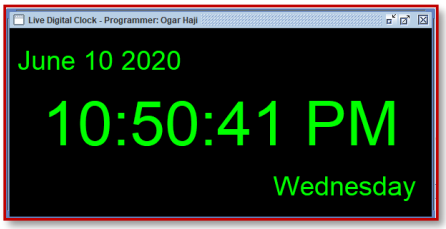


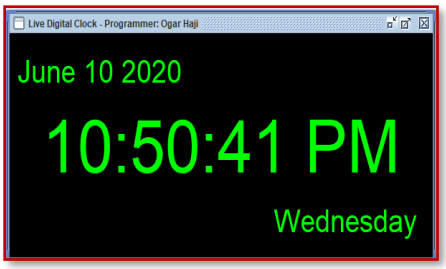

```
  case "time" :  // if the string is “time” then set for time format
    // a is to display AM or PM
     sdf = new SimpleDateFormat("hh:mm:ss a");
     setFont(new Font("sans-serif", Font.PLAIN, 88));
     setHorizontalAlignment(SwingConstants.CENTER);
     break;
```

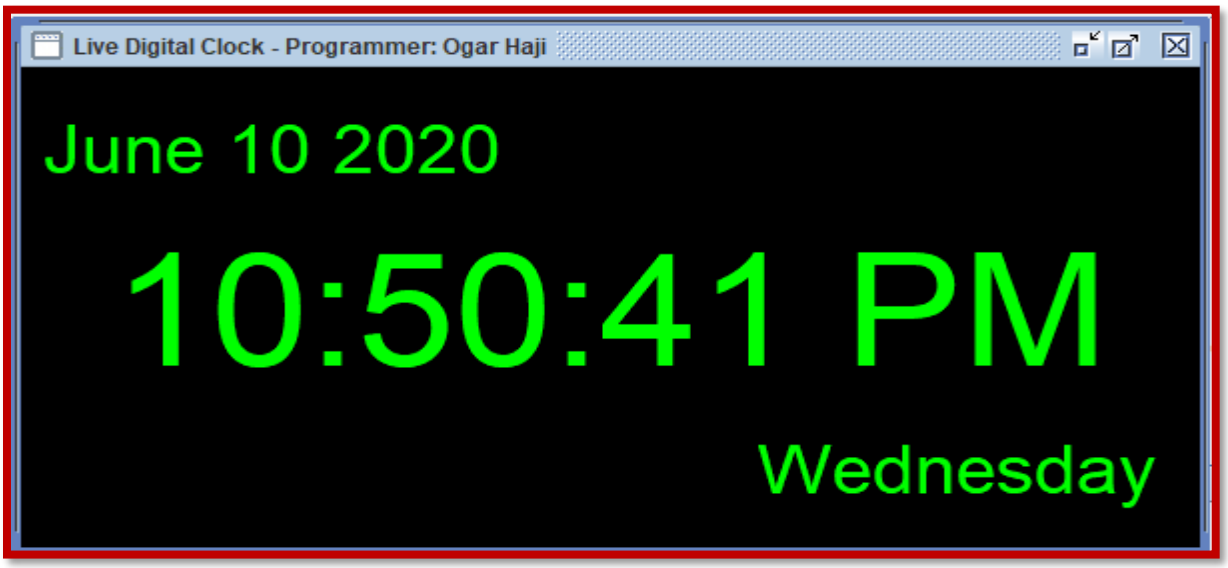


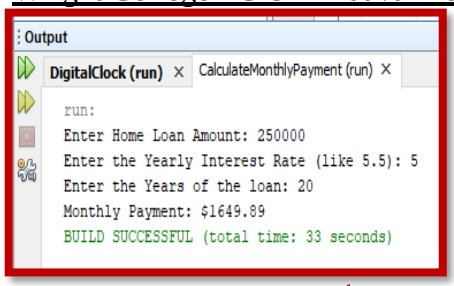

```
  case "day" :  // if the string is “day”,
    // EEEE to display the complete day then set for day format
     sdf = new SimpleDateFormat("EEEE  ");
     setFont(new Font("sans-serif", Font.PLAIN, 38));
     setHorizontalAlignment(SwingConstants.RIGHT);
     break;

  default :
```

```
        sdf = new SimpleDateFormat();
        break;
    }
    // Make the live Digital Clock click every 1000 Milli seconds = 1 Second
    Timer timer = new Timer(1000, this);
        // Start the timer so the clock will start ticking
    timer.start();
  }
  // Define and actionedPerfoemed() method
  public void actionPerformed(ActionEvent ae) {
    // 1) Create an instance object date from the Date class
    Date date = new Date();
    // 2) Set the text of the Date object date to simple date format
    setText(sdf.format(date));
  }
}
```

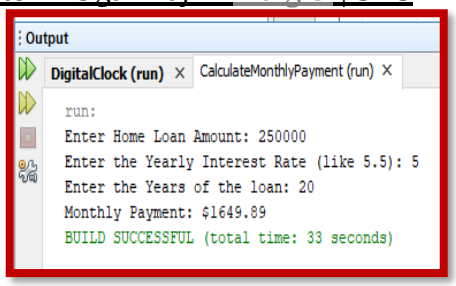








Calculate Monthly Payment of a Loan Java Project



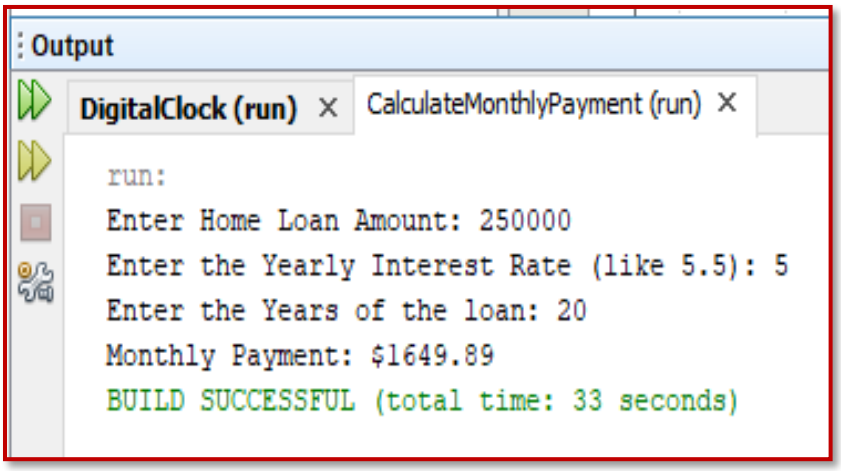
+++(Do Lab Exercise 2) 100 Points+++

Do Lab Exercise 2

Lesson 264 Ex : How to Calculate Monthly Payment of a Loan Java Project ?

Do the 12 Must Steps to Design, Code and Solve a project using Java Language.

This Java project will Display the following .



Do Lab Exercise

3) Launch **NetBeans IDE** program

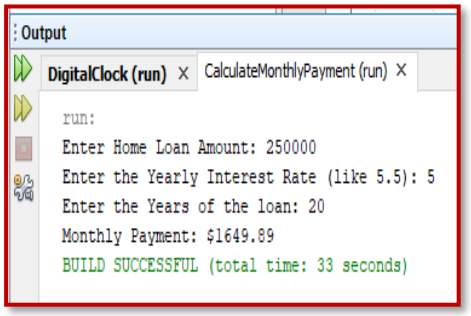
4) Type following **Java Lab exercise, Save** as **CalculateMonthlyPayment.**

```
/*
Calculate Home Loan Payment
*/
package calculatemonthlypayment;
```

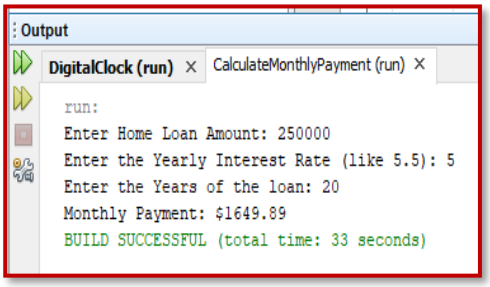
// 1) import the java Class Scanner

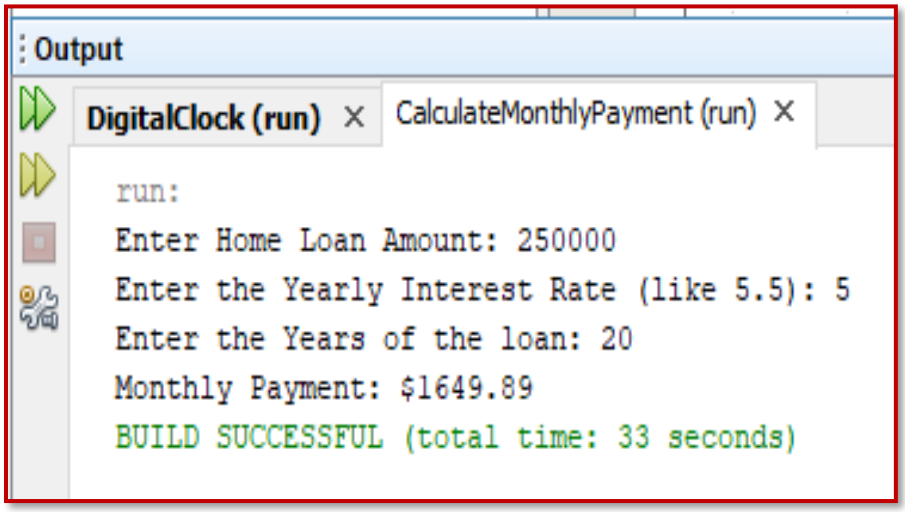
Do Lab Exercise

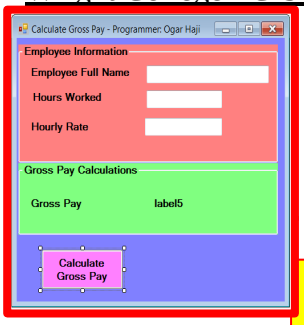
```
import java.util.Scanner;

public class CalculateMonthlyPayment {

  public static void main(String args[]){
    // 1) Declare the variables to be used in the project
    double loanAmount;
    double interestRate;
    int years;
    double monthlyPayment;

    // 2) Create an object input from the class Scanner
    Scanner input = new Scanner(System.in);

    // 3) Prompt the user to enter loanAmount
    System.out.print("Enter Home Loan Amount: ");

    // 4) Get loan Amount and store it in loanAmount variable
    loanAmount = input.nextDouble();

    // 5) Prompt the user to enter the Yearly Interest Rate
    System.out.print("Enter the Yearly Interest Rate (like 5.5): ");

    // 6) Get the Interest Rate and store it in interestRate variable
    interestRate = input.nextDouble();

    // 7) Calculate the Monthly intertest rate
    interestRate = interestRate / 100 / 12;
```



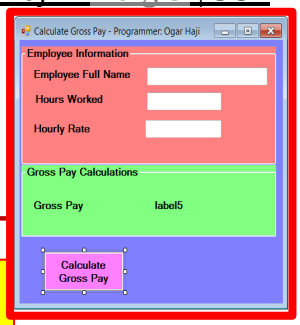
```
    // 8) Prompt the user to enter the Years
    System.out.print("Enter the Years of the loan: ");

    // 9) Get the years and store it in years variable
    years = input.nextInt();

    // 10) Calculate the years in months
    int months = years * 12;

    // Calculate Compound Monthly Payment
     monthlyPayment = (loanAmount * interestRate) /
                 (1 - Math.pow(1 + interestRate, -months));

    // Calculate Simple
     // monthlyPayment = loanAmount * months * interestRate;

    //System.out.println("Compound Calculations:" +result );
    // round to two decimals
    monthlyPayment = (double)Math.round(monthlyPayment * 100) / 100;

    System.out.println("Monthly Payment: $" + monthlyPayment);

  }  // End of main method

} // End of Class
```




Output
DigitalClock (run)
CalculateMonthlyPayment (run)
run:
Enter Home Loan Amount: 250000
Enter the Yearly Interest Rate (like 5.5): 5
Enter the Years of the loan: 20
Monthly Payment: $1649.89
BUILD SUCCESSFUL (total time: 33 seconds)



Using JFrame Form To Calculate Gross Pay Project +(Do Lab Exercise 3)100 Points+-



Do Lab Exercise 3 Repeated

Lesson 265 Ex: Steps to follow to Design, Code and Run a program to Calculate Gross Pay.

Do the following 12 Must Steps to Design, Code and Solve a project using Java Language.

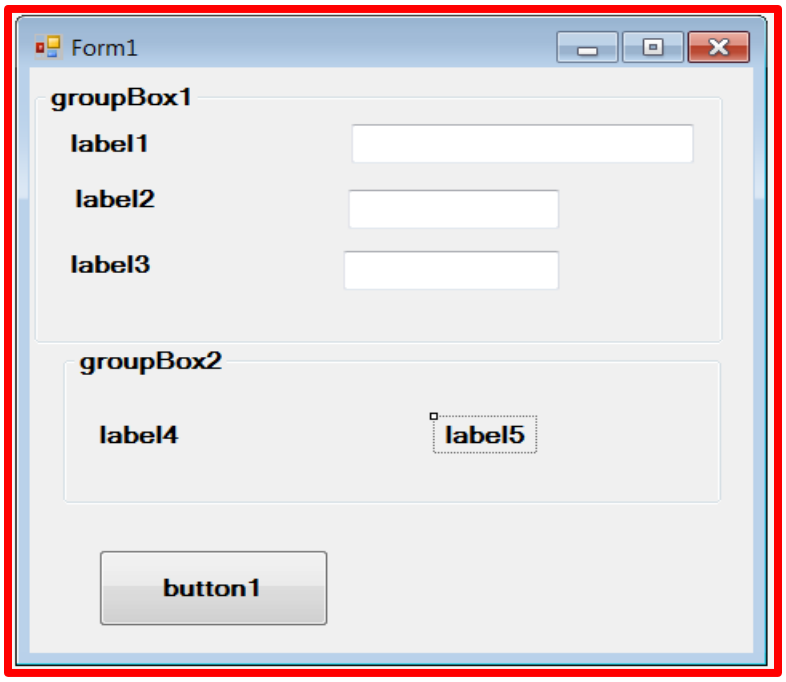
Do Steps 1 thru 7 in your Note Book or on Paper.

Step 1) Purpose of the Program: State what Program will do: (5 Points)

A)This Program will Calculate Gross Pay of Employees.

Step 2) Input: You should know how the Input looks like: (5 Points)

A) Design the **User Interface Form** by Adding **2 Panels (GroupBoxes)**, **5 Labels**, **3 Text Boxes** and **1 Button Controls** as shown below:

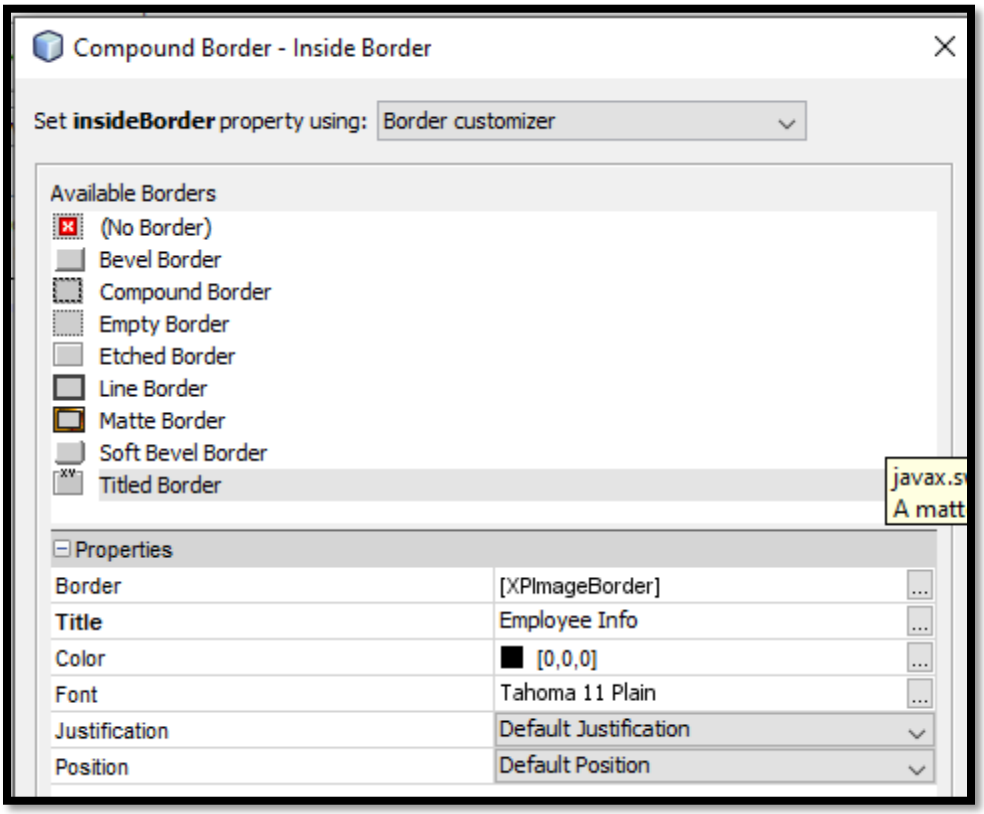


B) Set or **Change** the **Name** and **Text properties** of the **Controls** as shown below:

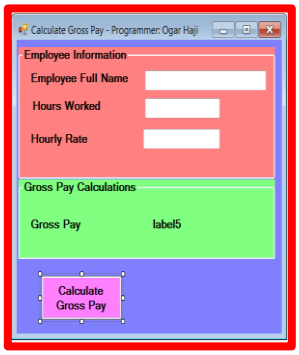
1. Setting the **Text property** of **Panels or GroupBoxes**

Set **Border Title** property of **Panel** to **"Employee Information" .**

Do the following: C Panel control, c Border build button [...],
C Compound Border, C Title and type "Employee Information"



- Set **Border Title** property of **Panel 2** to **"Gross Pay Calculations"**



1. Setting the **Text** and **Name properties** of **jLabels**

- Set **Text** property of **jLabel1** to **Employee Full Name**
- Set **Text** property of **jLabel2** to **Hours Worked**
- Set **Text** property of **jLabel3** to **Hourly Rate**
- Set **Text** property of **jLabel4** to **Gross Pay**
- Set **Name** property of **jLabel5** to **lblGrossPay**

3. Setting the **Text** and **Name properties** of **Text Boxes Controls**

- Set **Variable Name** property of **jTextField1** to **txtEmployeeFullName**
- Set **Variable Name** property of **jTextField 2** to **txtHours**
- Set **Variable Name** property of **jTextField 3** to **txtRate**

4. Setting the **Text** and **Name properties** of **Buttons Controls**

- Set **Text** property of **jButton1** to **Calculate Gross Pay** , set **Variable Name** property to **btnCalculateGrossPay**

5. Setting the **Text Property** of **Form Control**.

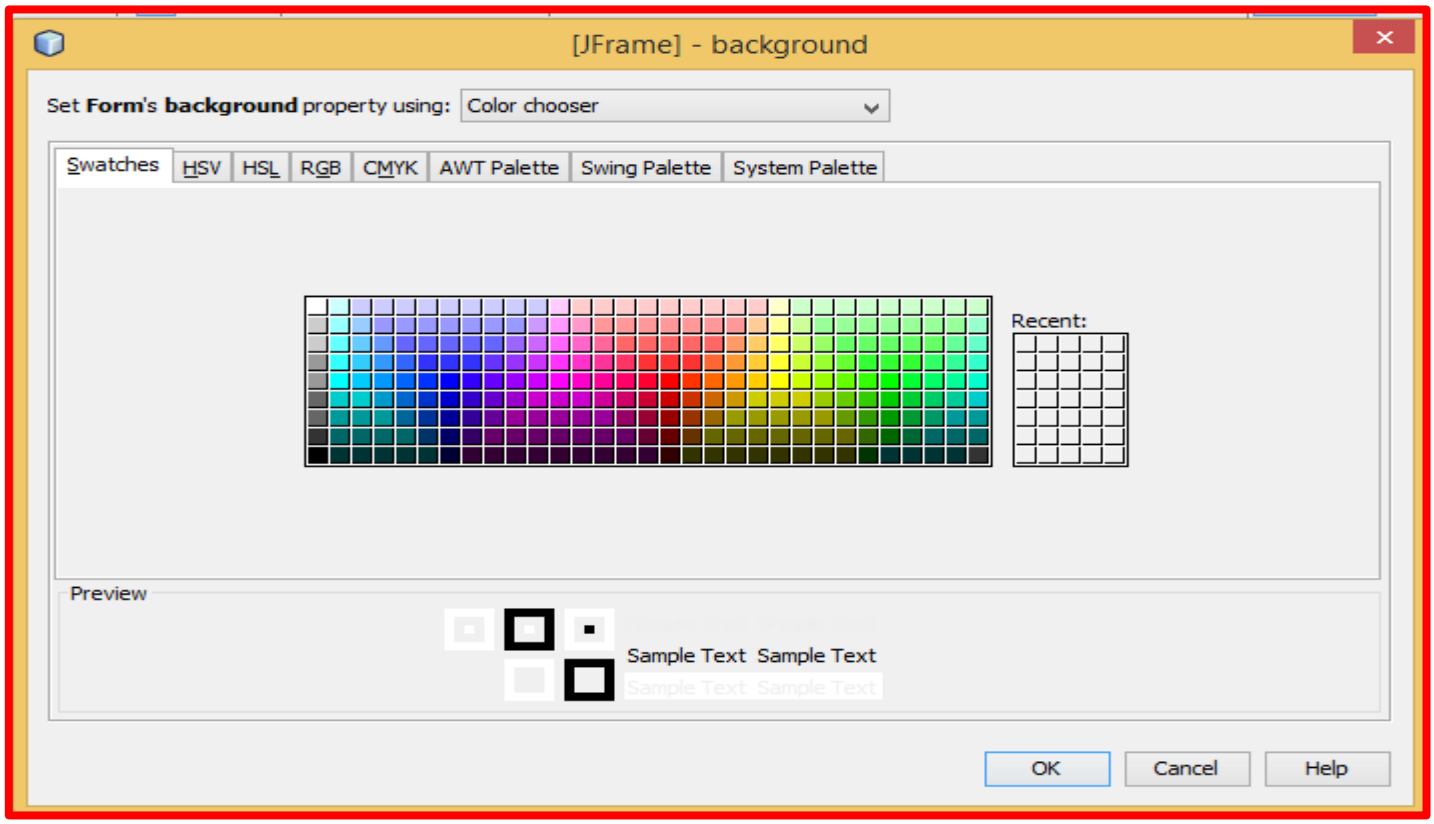
- **Set Text Property** of **Form1** to **"Calculate Gross Pay – Programmer: Your Name"**

6. Setting the **BackColor Property** of **Form Controls**.

a) Click BackGround property Ellipses 

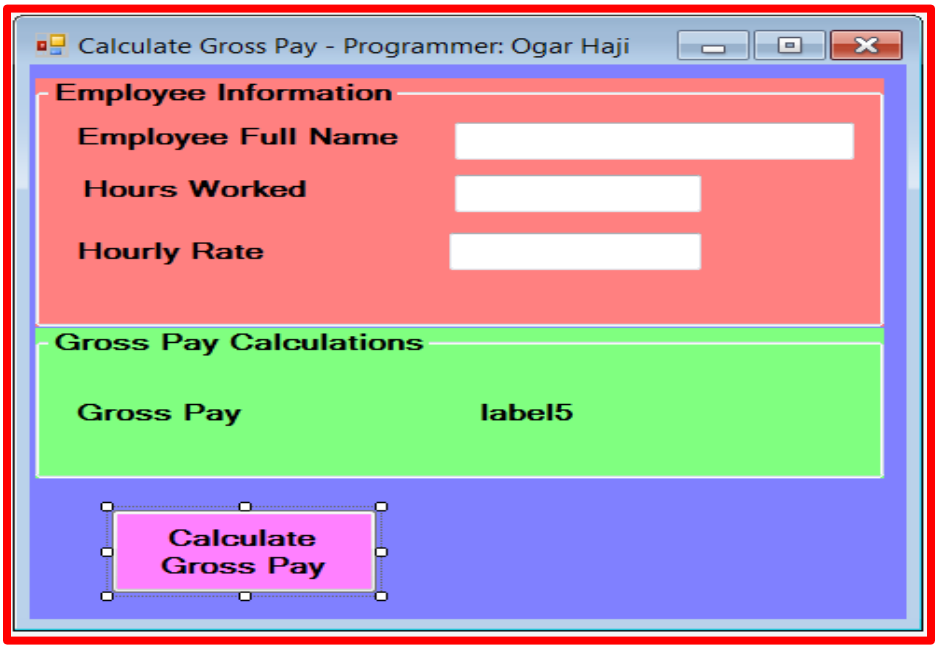
b) Click Swatches tab

c) Click on Color of your Choice (Light Colors)



- **Set BackGround Color Property** of **Form1** to **Light Gray.**
- **Set BackGround Color Property** of **Panel1** to **Light Red.**
- **Set BackGround Color Property** of **Panel2** to **Light Green.**
- **Set BackGround Color Property** of **jButton** to **Light Purple.**

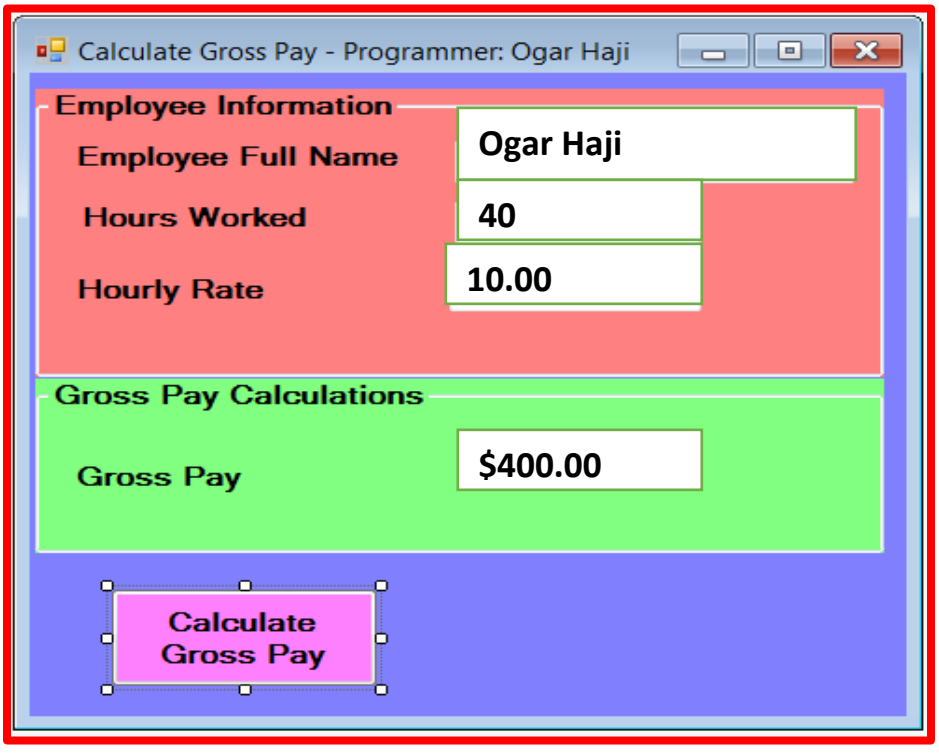
The User Interface Form will look like the Following:



Step 3) Processing and Calculation: The program will process each record and Calculate the Gross Pay: (5 Points)

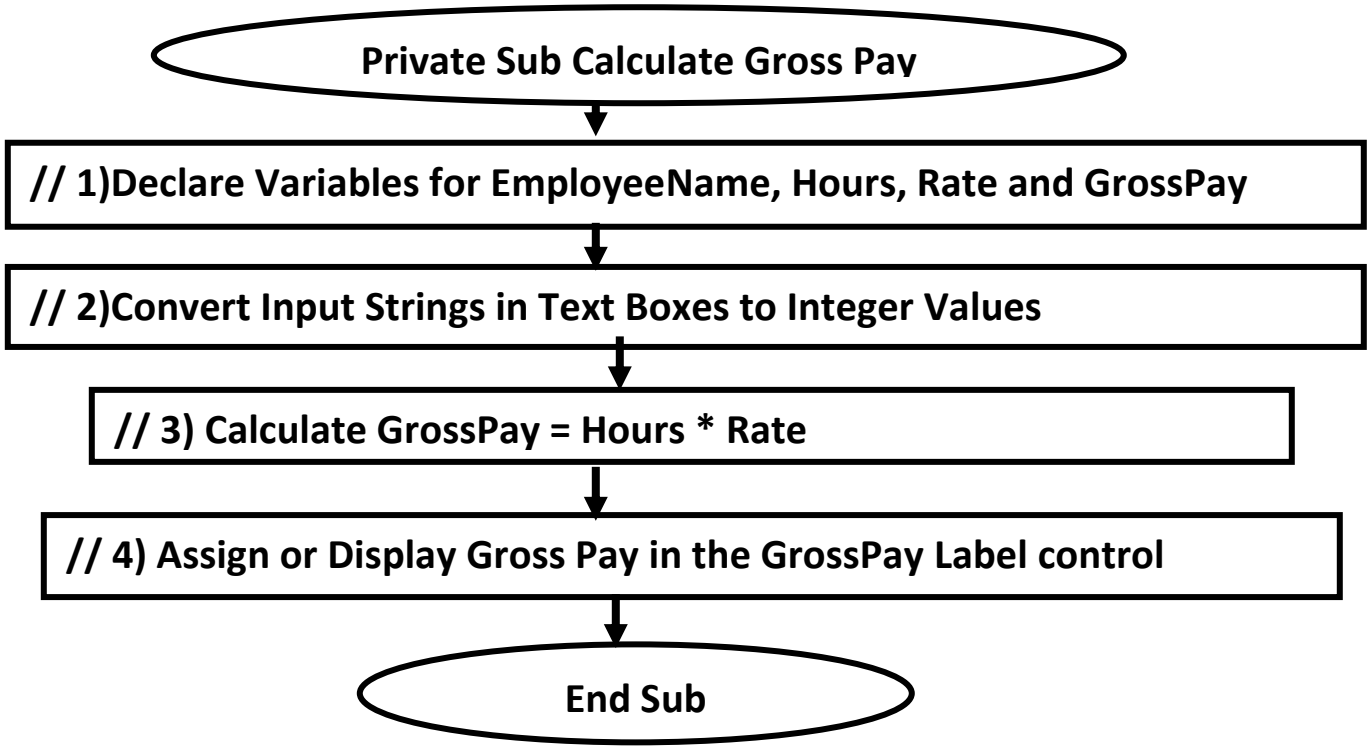
Gross Pay = hours * rate

Step 4) Output: You should know how output should look like: (5 Points)



Step 5) Flowchart: Draw a Flowchart to Calculate Gross Pay.

A) Draw a Flowchart



Step 6) PseudoCode: print a PseudoCode for the Program.(5 Points)

A) Write a Pseudo Code

Private Sub Calculate Gross Pay

Declare or Define Integer Variables for Hours. Rate, GrossPay
Convert Input Strings in Text Boxes to Integer Values
Calculate GrossPay = Hours * Rate
Assign or Display GrossPay in the GrossPay Label control

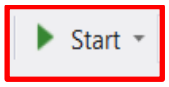
End Sub

Step 7) Code the Program in Java by referencing the Flowchart or Pseudocode you designed above and Save it as CalculateGrossPayForm.

a) Double-Click on **CalculateGrossPay Button** and **type** the **Java inside CalculateGrossPay() method**:

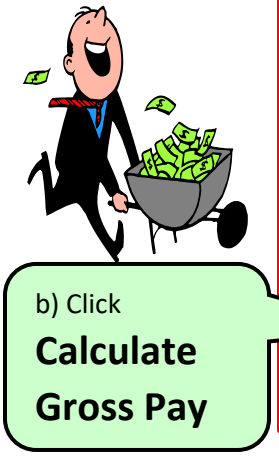
```
private void CalculateGrossPayActionPerformed(java.awt.event.ActionEvent evt){
```

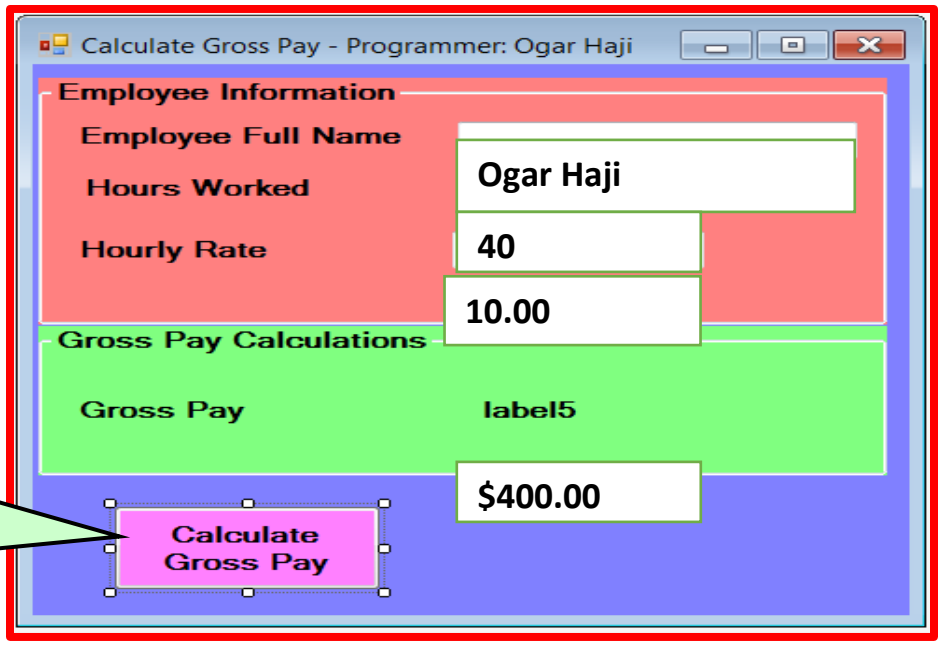
```
}
```

Step 8) Click **Start**  button to Start Running the program

a) Type Employee Name: Ogar Haji
Type Hours Worked: 40
Type Hourly Rate: 10.00

The following output appears on the Left side of the screen with the Input you entered and the correct Gross Pay $400.00.





If any Syntax Errors Found Do Next Step 9:

b) Click **Calculate Gross Pay** Button, the correct **Gross Pay $400.00** is displayed.

Step 9) Debug the Program: Debug or Correct any Syntax Errors until you have a clean Compiled program. (5 Points) (Clean compiled program means No Errors in the program).

Step 10) Test the Program: Test the Program with Test Data. (5 Points)

Repeat Step 10) Test the program many Times and Test the Program again and again until All conditions are tested:

Step 11) Documentation (5 Points): You have to add more comments to the Program (like Comments about the Purpose of the Program, Your Name and the Date the Program was written.)

```
##################################################################
# Purpose of the Program:
#  a) This Program will Calculate Gross Pay.
#  b) It will let the User to Enter User's Full Name:
#  c) It will let the User to Enter Hours Worked
#  d) It will let the User to Enter Hourly Rate.
#  e) The program will calculate the Gross Pay.
#       Gross Pay = Hours * Rate
#  f) Display Gross Pay
##################################################################
```

Step 12) Print a Copy of Java Code along with screen printout of the Running program. Submit to your Instructor the Print Copy and the screen Printout (Snaps) along with the following: (Which you did on Paper)

Copy the Java Code and the result of the program and Paste it in Microsoft Word program:

1) Purpose of the Program.
2) Input: how the Input looks like
3) Processing and Calculations
4) Output: how the Output will look like
5) Flowchart 6) Pseudocode
7) Java Code and
8) Print out copy of Java code and Output after running the program.

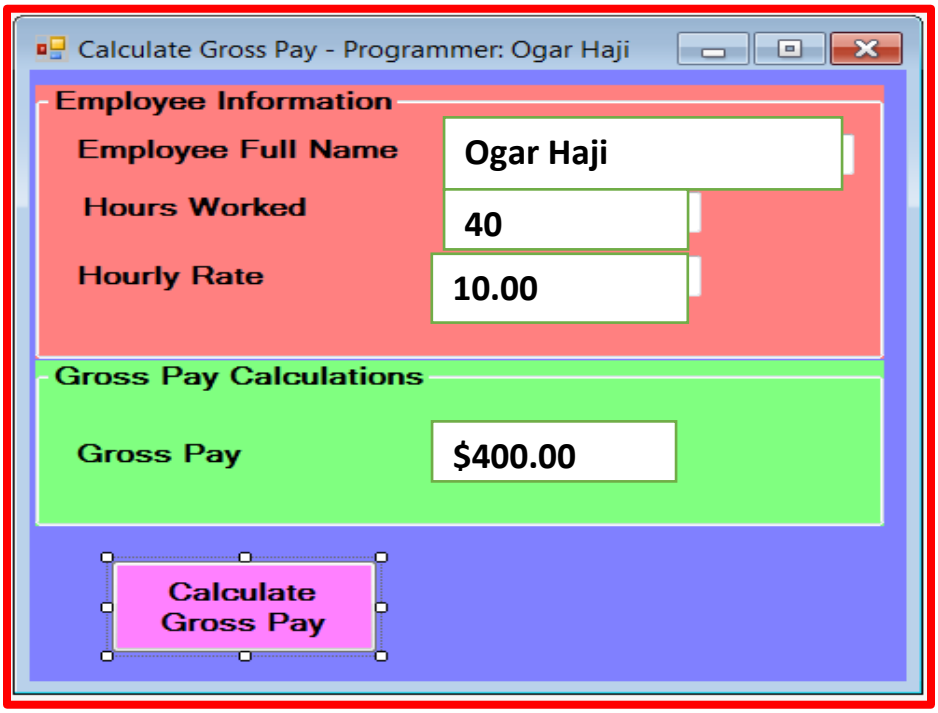
Submit the Programs on Time.

Remember Points will be deducted (20%) for Programs submitted Late.

Important Note:

1) Do Steps 1 thru 7 on Paper.
2) Then Get into NetBeans IDE
3) Type the Java code.
4) Save All the Files
5) Run the Program and Test it with Test Data for All Conditions.

strGrossPay = String.format("$ %7.2f", grossPay);



Modify the Calculate Gross Pay Java Project

+++(Modify Lab Exercise 3)+++

Modify Lab Exercise 3

Lesson 266 Ex: Steps to follow to Design, Code and Run a program to Calculate Gross Pay.

Do the following 12 Must Steps to Design, Code and Solve a project using Java Language.

Modify the program to Calculate and Display the Results of the following:

Note: Declare Constant Names

1) Calculate State Tax at 8% of the GrossPay

2) Calculate Federal Deductions at 15% of the GrossPay.

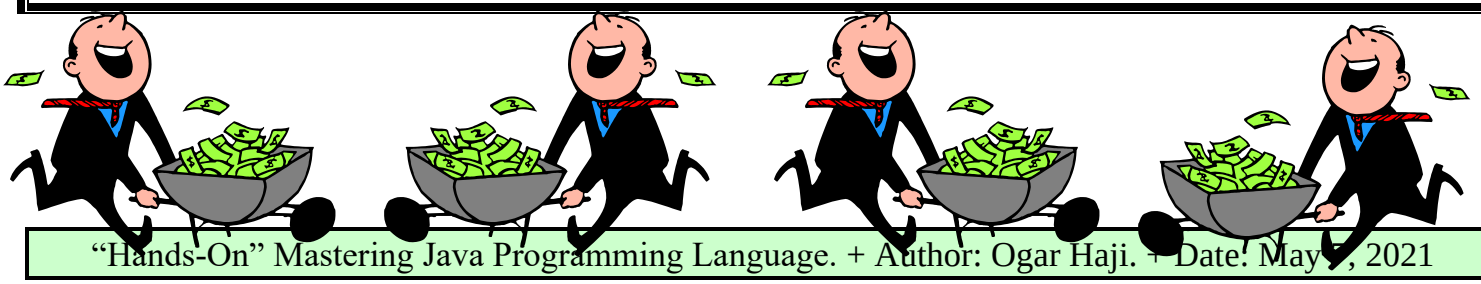
3) Calculate Total Deductions by adding State Tax and Federal Deductions.

4) Calculate Net Pay = GrossPay – Total Deductions

5) Format All Calculations with $ and 2 Decimal Places.

6) Add 2 Buttons (Clear Screen and Exit), set the Button Names and Text Properties and write the Java Code and Make sure it works correctly:

Note: use Constant Names to Declare State Tax and Federal Deductions Variables.



Modifying Gross Pay Project: Add Show/Hide Picture

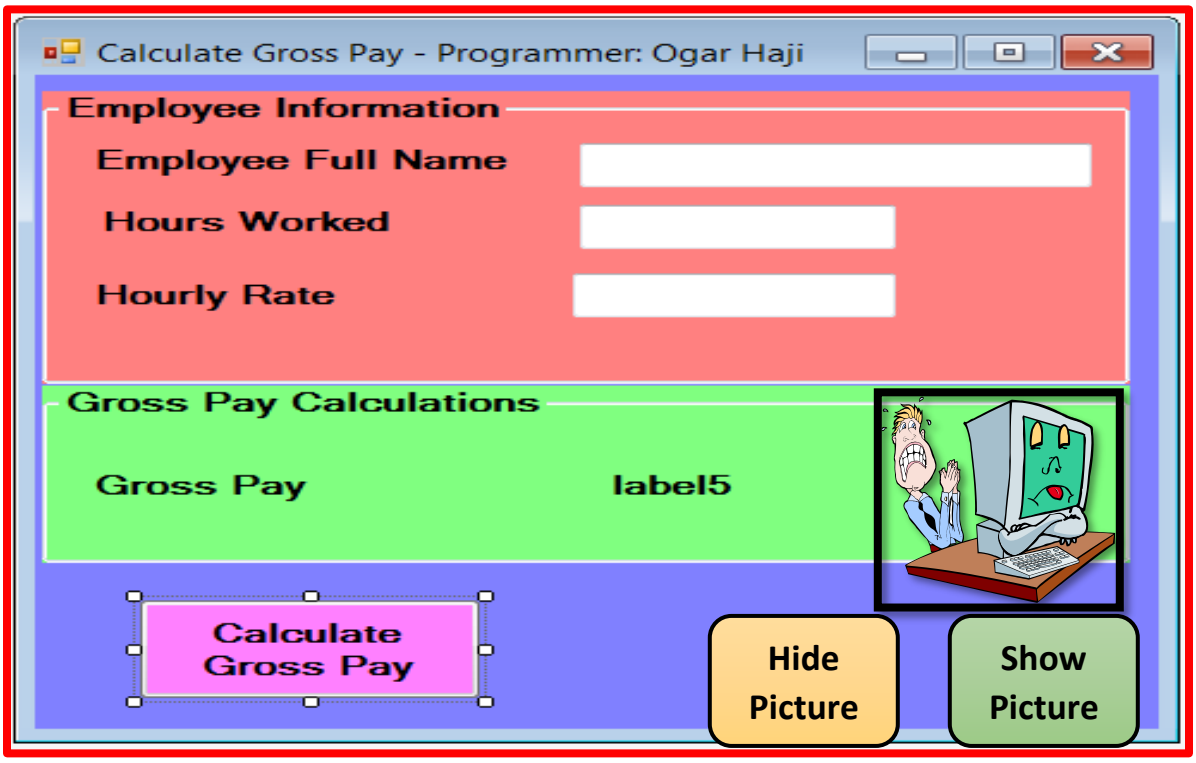
+++(Modify Lab Exercise 3)+++

Modify Lab Exercise 3

Lesson 267 Ex: Modify Gross Pay Project and Add Hide and Show a Picture?

1) Open File Calculate Gross Pay

2) Modify the Calculate Gross Pay program and add a Picture and 2 Buttons (Show Picture) and (Hide Picture)



3) Change the Text property and Name property of the Picture and Buttons: Also Add Exit Button.

4) Code the Buttons in Java Language.

5) Insert Date and Time Labels and write the Java Code to insert Date and Time automatically when the Form is loaded.

6) Run the program and make sure All the Buttons work correctly.

7) Upload to Brightspace.

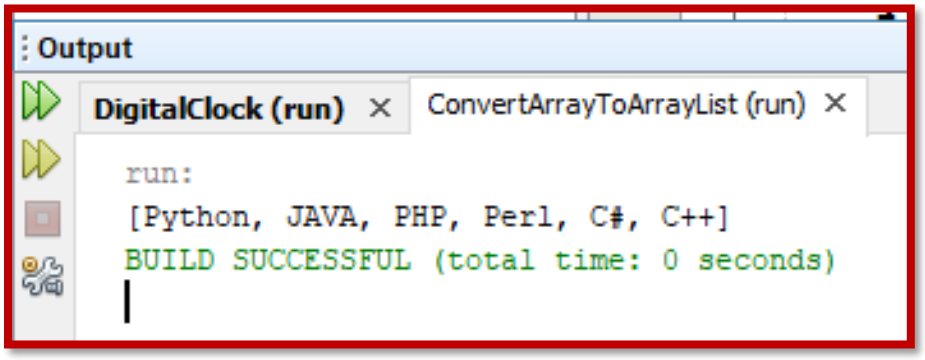
Convert an Array
To ArrayList
Java Project
+++(Do Lab Exercise 4) 100 Points+++

Do Lab Exercise 4

Lesson 268 Ex : How to Convert an Array to ArrayList using Arrays.asList() Method Java Project ?

Do the 12 Must Steps to Design, Code and Solve a project using Java Language.

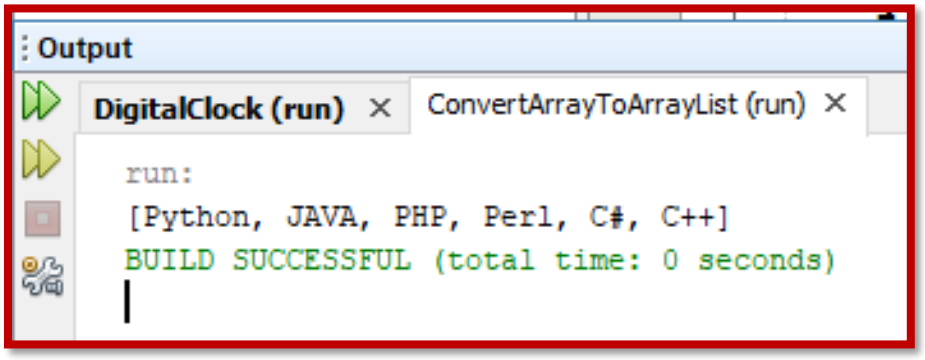
This Java project will Display the following .

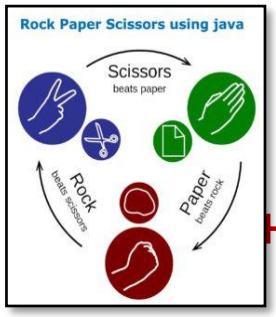


1. Launch **NetBeans IDE** program
2. Type following **Java project, Save** as **ConvertArrayToArrayList.**

```
/*
 This Java Project will convert an Array to ArrayList
Project Name: ConvertArrayToArrayList
 */
package convertarraytoarraylist;
```

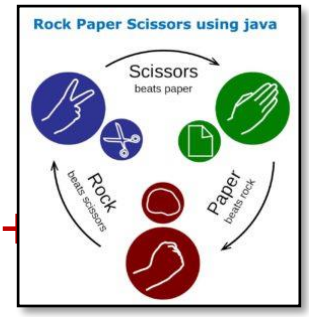
```
/**
 * @author Ogar's Laptop
 */

import java.util.ArrayList;
 import java.util.Arrays;

public class ConvertArrayToArrayList {

   /**
    * @param args the command line arguments
    */
   public static void main(String[] args) {

        // 1) Declare a String Array called languagesArray and initialize it with these
        // languages
       String[]  languagesArray = new String[]
                   {"Python", "JAVA", "PHP",  "Perl", "C#", "C++"};
       // 2) Declare an ArrayList called languagesArrayList and Convert the
      // 3) Array languagesArray to an ArrayList using Arrays.asList() method
         ArrayList<String> languagesArrayList= new ArrayList<>
                (Arrays.asList(languagesArray));

       // 4) Print out the ArrayList to the Console
     System.out.println(languagesArrayList);
  }
}
```


Output
DigitalClock (run)
ConvertArrayToArrayList (run)
run:
[Python, JAVA, PHP, Perl, C#, C++]
BUILD SUCCESSFUL (total time: 0 seconds)



Simulate Rock, Paper and Scissors Game using Java

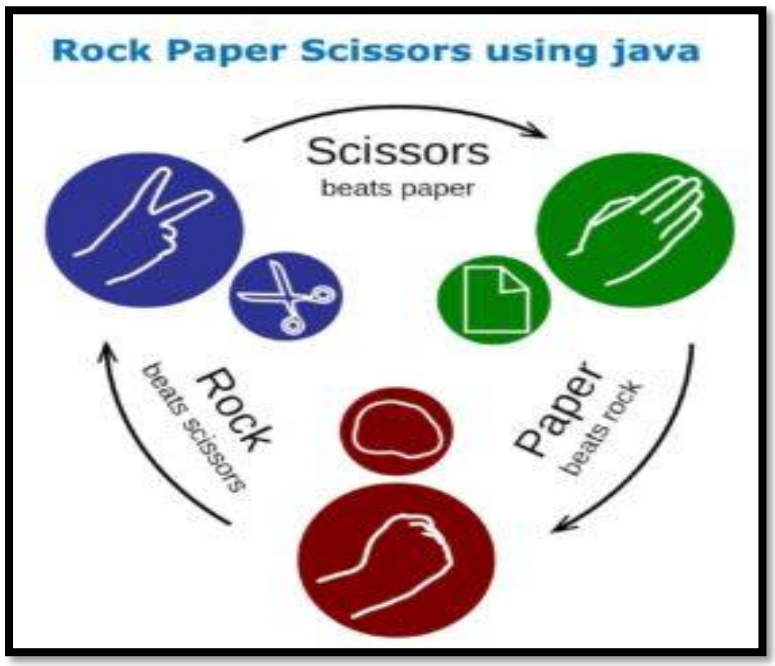
+++(Do Lab Exercise 8) 100 Points+++



Do Lab Exercise 8

Lesson 269 - How to Design a Project to Simulate Rock, Paper and Scissors Game in Java?

In this example we are going to develop a Rock Paper Scissors Game using Java. Developing games while learning java enhances our programming skills.



The Rules of the Game are as follows:

1. Rock beats Scissors.
2. Scissors beats Paper.
3. Paper beats Rock.
4. Whoever wins after playing 3 times is the Winner.

5. **In this example the game will be between a User and the Computer.**

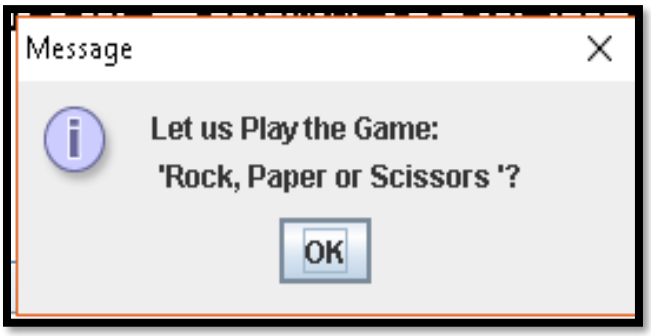
Steps to follow to Design the Rock Paper Scissors Java Program:

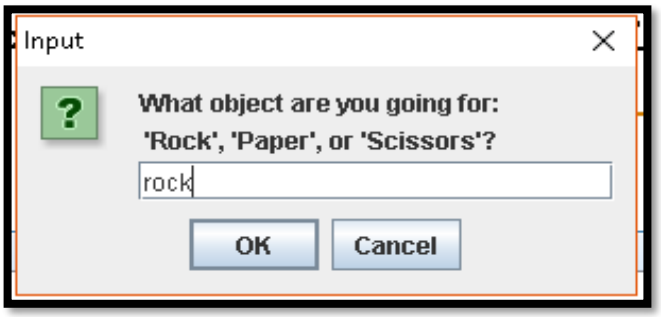
1. **We will create a class Player.java that represents the user who is going to play the game with the computer.**
2. **The class Computer.java represents the Computer that is playing with the user.**
3. **Driver.java is a helper class containing the Rock Paper Scissor constant variables and the business logic code.**
4. **RPS.java is the class containing the main function where the program execution will begin.**
5. **We will ask the user to enter his name. Then select one among ROCK PAPER SCISSORS.**
6. **Then we will let the computer randomly select ROCK PAPER SCISSORS.**
7. **We will use the business logic function to find out who wins. And every time we will display the selections made and who won.**
8. **They can decide to end the game any time.**
9. **We will keep a track of who win 5 times first. And then display the winner.**

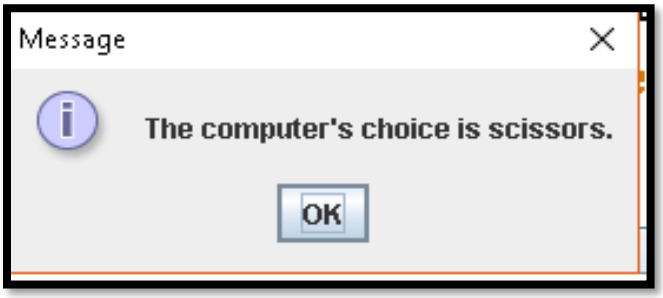
Player.java will represent the User playing with the computer.

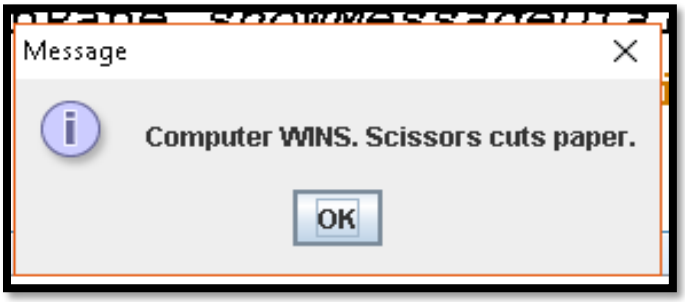
Rock Paper Scissors

The Input/Output of this Java project looks like the following:







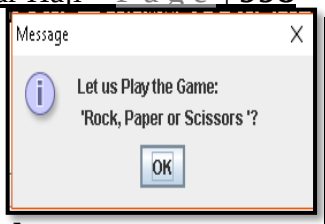


1) Type the following java project in **Netbeans,**

2) Save as **RockPaperOrScissors**

Do Lab Exercise

```
/*
 Rock, Paper or Scissors Game
Programmer: Ogar Haji
 */
package paper;

// Import the following Java classes
import javax.swing.JOptionPane;
import java.util.Random;

public class Paper {
   public static void main(String[] args)  {
      // 1) Declare the variables to be used in the project
      String computerPlayer, userPlayer;
```




```
//2) Show Message Dialog to play the game
JOptionPane.showMessageDialog(null,
    "Let us Play the Game:\n 'Rock, Paper or Scissors '?");
```

```
    // 3) Call computerChoice() method to simulate computer player
    computerPlayer = computerChoice();
```

```
    // 4) Call userChoice() method to prompt for user player
    userPlayer = userChoice();

    // 5) Check if  user Player has entered a choice
    if (userPlayer != null)  {
       // 6) show the computer choice
       JOptionPane.showMessageDialog(null,
            "The computer's choice is " + computerPlayer + ".");

       // 7) Call the winner() method to check who the winner is
       winner(computerPlayer, userPlayer);
    }

    // 8) Error entry - Prompt the user to enter 'Rock, Paper or Scissors
    else {
      JOptionPane.showMessageDialog(null, "Error: Improper User Entry. "
            + "Please enter either"
            + " 'Rock', 'Paper', or 'Scissors'.");
     }
}
```

```
// 3) Define and code computerChoice method
public static String computerChoice()  {
   // 1) Declare the local variables to be used in this method
   byte computerChoice;

   String computerChoiceString = "";

   // 2) Create an object choiceGenerator from class Random()
   Random choiceGenerator = new Random();
```

```
    // 3) Generate 3 numbers: 1, 2, 3 and assign (1=rock, 2=paper, 3=scissors
    computerChoice = (byte)(choiceGenerator.nextInt(3) + 1);
```

```
    // 4) Use switch statement to check computerChoice
    switch (computerChoice)   {
       case 1:  {
```

```
          computerChoiceString = "rock";
          break;
       }
```


```
       case 2:  {
          computerChoiceString = "paper";
          break;
       }
       case 3:  {
```

```
          computerChoiceString = "scissors";
          break;
       }
    }
```

```
    // 5) Return the value chosen by computer
    return computerChoiceString;
 }
```

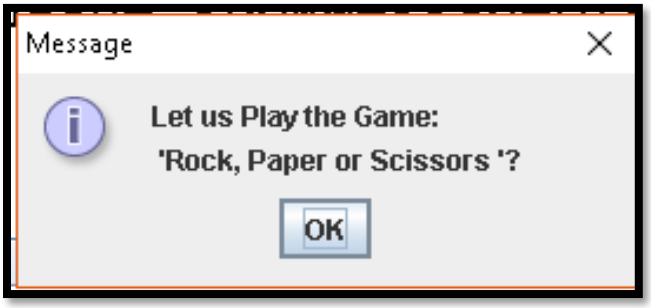
```
 // 4) Define and code userChoice method
 public static String userChoice()  {
    // 1) Declare the local variables to be used in this method
    String userChoice, userChoiceLowerCase;
    // 2) Prompt user to enter his/her choice (Rock, Paper or Scissors)
 userChoice = JOptionPane.showInputDialog("What object are you going
for:\n"
                + " 'Rock', 'Paper', or 'Scissors'?");
```

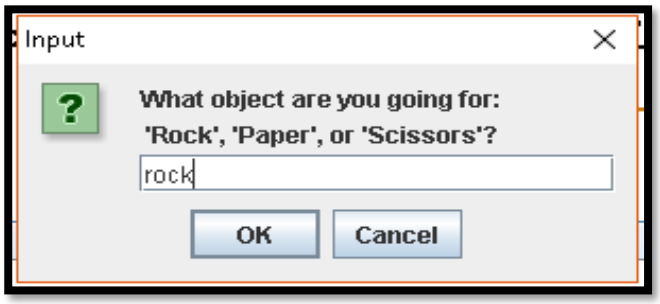
```
    // 3) Check the user choice of (rock, paper or scissors)
    if (userChoice.equalsIgnoreCase("rock")
```

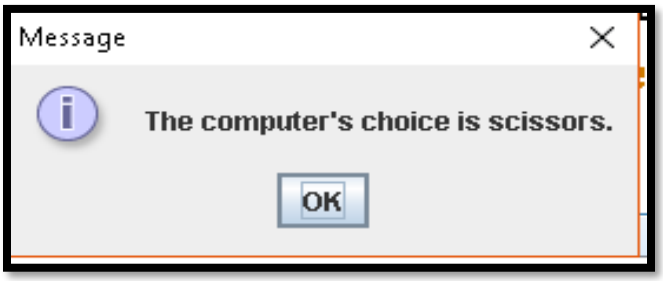
```
            || userChoice.equalsIgnoreCase("paper")
            || userChoice.equalsIgnoreCase("scissors")) {

     // 4) Convert user choice to lowercase
        userChoiceLowerCase = userChoice.toLowerCase();
    }
    else  {
        // 5) if not correct choice, Assign null to user choice
        userChoiceLowerCase = null;
    }
    // 6) Return the value chosen by user
    return userChoiceLowerCase;
}
// 7) Define and code the winner method
public static void winner(String computerSide, String userSide) {
    // 1) Compare if computer choice is same as player ( a TIE)
    if (computerSide.equals(userSide)) {
        JOptionPane.showMessageDialog(null,
                "The game has to be played again; we have a TIE.");
    }
    // 2) Check if computer choice is 'rock' and player is 'paper'
    // then the Player wins
    else if (computerSide.equalsIgnoreCase("rock")
            && userSide.equalsIgnoreCase("paper")) {
        JOptionPane.showMessageDialog(null,
                "The Player WINS. Paper covers rock.");
    }
    // 3) Check if computer choice is 'rock' and player is 'scissors'
    // then the Computer wins
    else if (computerSide.equalsIgnoreCase("rock")
            && userSide.equalsIgnoreCase("scissors")) {
```

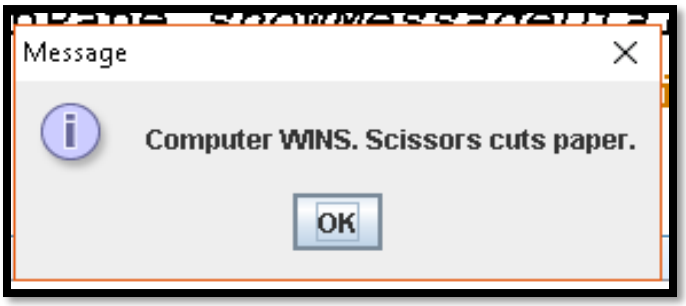
```
            JOptionPane.showMessageDialog(null,
                    "Computer WINS. Rock crushes scissors.");
         }
        // 4) Check if computer choice is 'paper' and player is 'rock'
        // then the Computer wins
        else if (computerSide.equalsIgnoreCase("paper")
                && userSide.equalsIgnoreCase("rock")) {
            JOptionPane.showMessageDialog(null,
                    "Computer WINS. Paper covers rock.");
        }
        // 5) Check if computer choice is 'paper' and player is 'scissors'
        // then the Player wins
        else if (computerSide.equalsIgnoreCase("paper")
                && userSide.equalsIgnoreCase("scissors")) {
            JOptionPane.showMessageDialog(null,
                    "The Player WINS. Scissors cuts paper.");
        }
        // 6) Check if computer choice is 'scissors' and player is 'rock'
        // then the Player wins
        else if (computerSide.equalsIgnoreCase("scissors")
                && userSide.equalsIgnoreCase("rock")) {
            JOptionPane.showMessageDialog(null,
                    "The player WINS. Rock crushes scissors.");
         }
        else {
            JOptionPane.showMessageDialog(null,
                    "Computer WINS. Scissors cuts paper.");
        }
    }
}
```

The Input/Output of this Java project looks like the following:









Modified program:

1) **Use for statement to run the project 3 times and display the Final winner for the game:**
2) **Use while statement and prompt the user if he/she want to continue after the 3 times have finished:**
   ("Do you want to Continue Playing the Game (Paper, Rock and Scissor: (Y/N));

```
/*
 Rock, Paper or Scissors Game
 */
package paper;

// Import the following classes
import javax.swing.JOptionPane;
import java.util.Random;
public class Paper {
// Declare Global static variables to be used in the entire project
public static int computerWinCount = 0;
```

```
public static int playerWinCount = 0;
public static int tiesCount = 0;
  public static void main(String[] args)  {
     // 1) Declare the variables to be used in the project
     String computerPlayer, userPlayer;
     for ( int i = 1; i <=3; i++)  {
     //2) Show Message Dialog to play the game
     JOptionPane.showMessageDialog(null,
           "Let us Play the Game:\n 'Rock, Paper or Scissors '?");
     // 3) Call computerChoice() method to simulate computer player
     computerPlayer = computerChoice();
     // 4) Call userChoice() method to prompt for user player
     userPlayer = userChoice();
     // 5) Check if  user Player has entered a choice
     if (userPlayer != null)  {
        // 6) show the computer choice
        JOptionPane.showMessageDialog(null,
              "The computer's choice is " + computerPlayer + ".");
        // 7) Call the winner() method to check who the winner is
        winner(computerPlayer, userPlayer);
     }
     // 8) Error entry - Prompt the user to enter 'Rock, Paper or Scissors
     else  {
       JOptionPane.showMessageDialog(null, "Error: Improper User Entry. "
             + "Please enter either"
             + " 'Rock', 'Paper', or 'Scissors'.");
     }
  }
   // 9) Call theFinalWinnerIs() method to announce the winner
   theFinalWinnerIs();
```

```
}
  // 3) Define and code computerChoice method
  public static String computerChoice()  {
     // 1) Declare the local variables to be used in this method
     byte computerChoice;  // a byte can store up to 128
     String computerChoiceString = "";
     // 2) Create an object choiceGenerator from class Random()
     Random choiceGenerator = new Random();
     // 3) Generate 3 numbers: 1, 2, 3 and assign (1=rock, 2=paper, 3=scissors
     computerChoice = (byte)(choiceGenerator.nextInt(3) + 1);
     // 4) Use switch statement to check computerChoice
     switch (computerChoice) {
        case 1:  {
           computerChoiceString = "rock";
           break;
        }
        case 2:  {
           computerChoiceString = "paper";
           break;
        }
        case 3:  {
           computerChoiceString = "scissors";
           break;
        }
     }
// 5) Return the value chosen by computer
     return computerChoiceString;
  }
  // 4) Define and code userChoice method
  public static String userChoice()  {
```

```
    // 1) Declare the local variables to be used in this method
    String userChoice, userChoiceLowerCase;
    // 2) Prompt user to enter his/her choice (Rock, Paper or Scissors)
userChoice = JOptionPane.showInputDialog("What object are you going for:\n"
                + " 'Rock', 'Paper', or 'Scissors'?");
    // 3) Check the user choice of (rock, paper or scissors)
    if (userChoice.equalsIgnoreCase("rock")
          || userChoice.equalsIgnoreCase("paper")
          || userChoice.equalsIgnoreCase("scissors"))  {
       // 4) Convert user choice to lowercase
       userChoiceLowerCase = userChoice.toLowerCase();
    }
    else  {
       // 5) if not correct choice, Assign null to user choice
       userChoiceLowerCase = null;
    }
    // 6) Return the value chosen by user
    return userChoiceLowerCase;
 }
 // 7) Define and code the winner method
 public static void winner(String computerSide, String userSide) {
    // 1) Compare if computer choice is same as player ( a TIE)
    if (computerSide.equals(userSide)) {
       JOptionPane.showMessageDialog(null,
             "The game has to be played again, we have a TIE.");
       tiesCount++;
    }
    // 2) Check if computer choice is 'rock' and player is 'paper'
    // then the Player wins
    else if (computerSide.equalsIgnoreCase("rock")
          && userSide.equalsIgnoreCase("paper"))  {
```

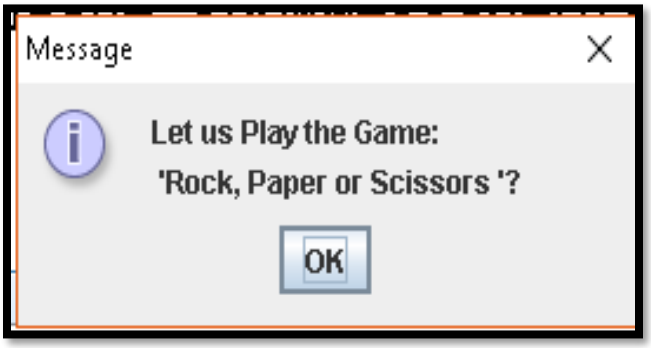
```
    JOptionPane.showMessageDialog(null,
          "The Player WINS. Paper covers rock.");
    playerWinCount++;
}
// 3) Check if computer choice is 'rock' and player is 'scissors'
// then the Computer wins
else if (computerSide.equalsIgnoreCase("rock")
      && userSide.equalsIgnoreCase("scissors"))  {
    JOptionPane.showMessageDialog(null,
          "Computer WINS. Rock crushes scissors.");
    computerWinCount++;
}
```

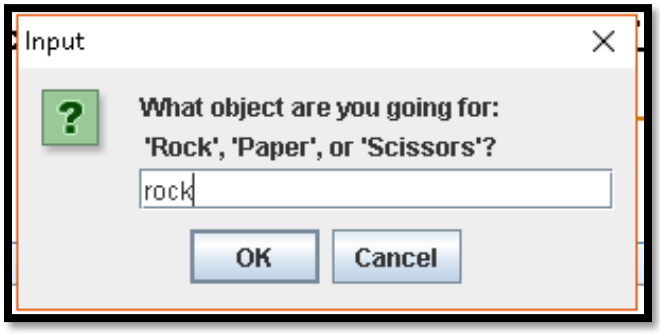
```
// 4) Check if computer choice is 'paper' and player is 'rock'
// then the Computer wins
else if (computerSide.equalsIgnoreCase("paper")
      && userSide.equalsIgnoreCase("rock"))  {
    JOptionPane.showMessageDialog(null,
          "Computer WINS. Paper covers rock.");
    computerWinCount++;
}
```

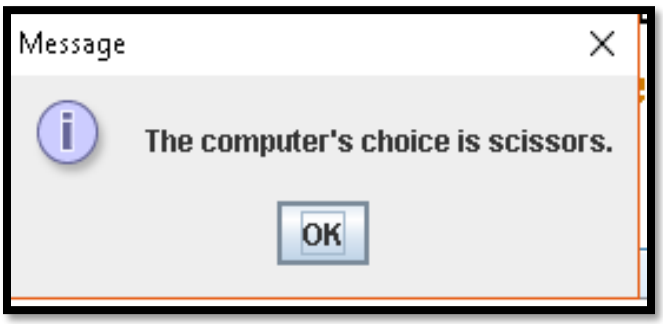

```
// 5) Check if computer choice is 'paper' and player is 'scissors'
// then the Player wins
else if (computerSide.equalsIgnoreCase("paper")
      && userSide.equalsIgnoreCase("scissors")) {
    JOptionPane.showMessageDialog(null,
          "The Player WINS. Scissors cuts paper.");
    playerWinCount++;
}
// 6) Check if computer choice is 'scissors' and player is 'rock'
// then the Player wins
else if (computerSide.equalsIgnoreCase("scissors")
      && userSide.equalsIgnoreCase("rock"))  {
```

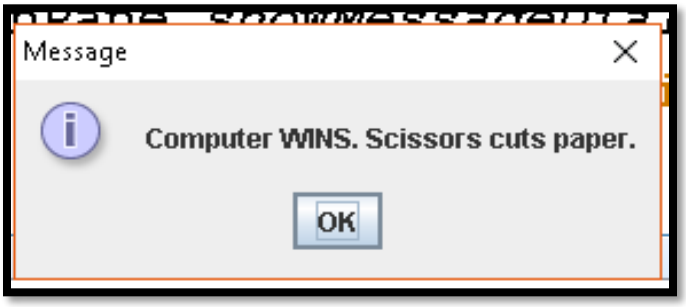
```
            JOptionPane.showMessageDialog(null,
                    "The player WINS. Rock crushes scissors.");
            playerWinCount++;
        }
        else  {
            JOptionPane.showMessageDialog(null,
                    "Computer WINS. Scissors cuts paper.");
            computerWinCount++;
        }
    }
    // 9) Define and code the Final Winner Is () method to see who the winner is
    public static void theFinalWinnerIS() {
        // 1) Check who the winner is
        if (playerWinCount > computerWinCount)  {
            JOptionPane.showMessageDialog(null,
                    "The Winner is the Player with wins = " + playerWinCount);
        }
        else if (computerWinCount > playerWinCount) {
             JOptionPane.showMessageDialog(null,
                  "The Winner is the Computer with wins = " + computerWinCount);
        }
        else  {
            JOptionPane.showMessageDialog(null,
                    "\nThere is a TIE with ties = " + tiesCount +
                    "\nThe Player Win Count = " + playerWinCount +
                    "\nThe Computer Win Count = " + computerWinCount);
        }
    }
}
```

The Input/Output of this Java project looks like the following:









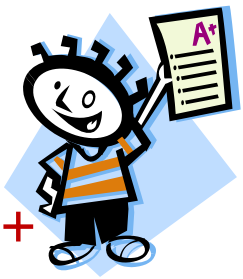
Please, Read, Study and Practice the Lessons in the Java Handout

Multiple Choice Quiz: (1 Question)

if (answer.equalsIgnoreCase("a"))

Using Methods ()

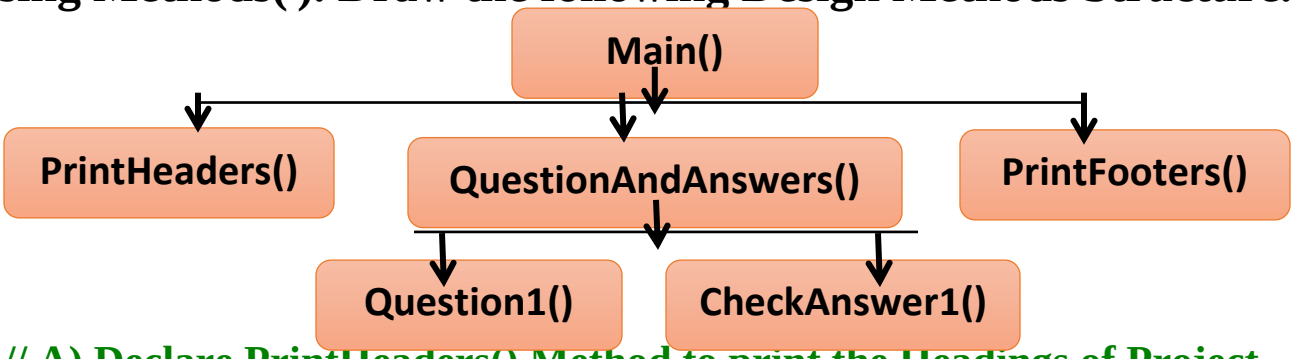
++(Do Lab Exercise 3) 100 Points +++



Do Lab Exercise 3

Lesson 270 Ex: How to Create Multiple Choice Test in Java language?

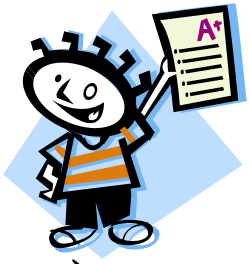
You can Design and Create Multiple Choice Test in Java language using Methods( ). Draw the following Design Methods Structure.

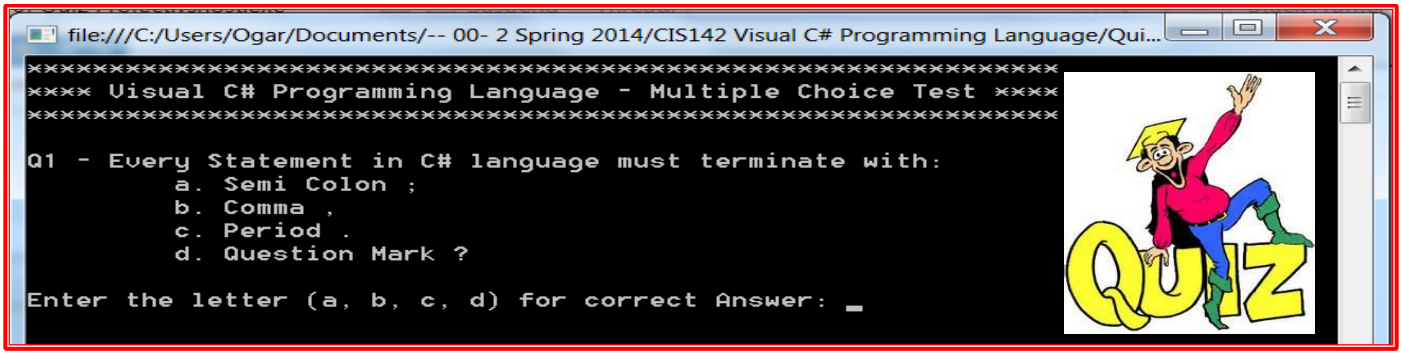



```
// A) Declare PrintHeaders() Method to print the Headings of Project
    public static void printHeaders()
 // B) Declare QuestionsAndAnswers() Method to display Question and let the
 //  User Answer it and then Call CheckAnswer() method
    public static void questionsAndAnswers() {
        // C) Declare Question1() Method which contains Question1
            public static string question1()
        // D) Declare CheckAnswer1() Method to check the Answer1
            public static void checkAnswer1(string answer)
 // G) Declare PrintFooters() Method to print the Footers of Project
        public static void printFooters()
 // H) Call the Methods from the Main() Method
    public static void main(String[] args) {
     // 1) Call the Methods to be used in the project from the Main() method
        printHeaders();
        questionsAndAnswers();
        printFooters();
    }
// 2) Display the Question 1 on the Console
```

```
System.out.println("Q1-Every Statement in Java language must terminate with:");
  System.out.println(" \t a. Semi Colon ; ");
  System.out.println(" \t b. Comma , ");
  System.out.println(" \t c. Period . ");
  System.out.println(" \t d. Question Mark ? ");
  System.out.println();
```


```
 // 3) Prompt User to Enter the letter (a, b, c, d) for Correct Answer
 System.out.print("Enter the letter (a, b, c, d) for correct Answer: ");
// 4) Read the User Answer from the console
  answer = input.nextLine();
 // 5) Check if the answer is Correct which is Answer 'a'
  if  (answer.equalsIgnoreCase ("a") ) {
     System.out.printf("Your Answer %s is Correct.", answer);
  }
  else {
     System.out.printf("Your Answer %s is Wrong.", answer);
  }
```



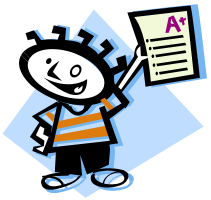


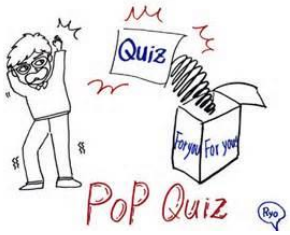
1) Ex. Type the following Java Code program,
2) Save the Java Project as **JavaQuiz**.

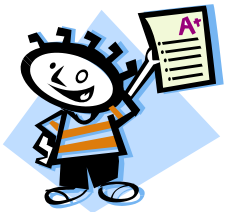
Do Lab Exercise

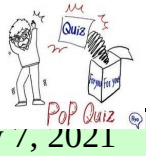
```
/*      Write your Comments here
        Date: Friday, November 17, 2017
        Programmer: Ogar Haji
        Project Name: Quiz
*/
```

```
package multiplechoicequiz;
// 1) Import the Classes needed for the project
import java.util.Scanner;

public class MultipleChoiceQuiz {
// 1) Declare the global variable answer to be used in the project
    public static String answer = "";
    public static int correctAnswersCount = 0;
    public static int wrongAnswersCount = 0;
    public static int numberOfQuestions = 0;
    public static int score = 0;

   // Create a public static Scanner only Once here to be Globa
    public static Scanner input = new Scanner(System.in);
public static void main(String[] args) {
     // A) Call the Methods from the main() Method
    // 1) Call print Headers method to print the headings of project
      printHeaders();
    // 2) Call Questions and Answers method to display Questions and Answers
      questionsAndAnswers();
      calculateStudentScore();
    // 3) Call print Footers method to print the Footers of project
      printFooters();
} // End of main() method
 // B) Define & Code PrintHeaders() Method to print the Headings of Project
    public static void printHeaders()  {
```

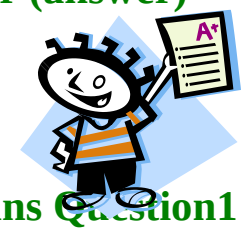








```
    // 1) Print the Headings of the project
    System.out.println("***************************************");
    System.out.println("****** Java Programming Language" +
                       " Multiple Choice Test *******");
    System.out.println("**Read Questions carefuly, enter Correct answer" +
                       " Letters (a,b,c,d)**");
    System.out.println("***************************************");
    System.out.println();
}

 // C) Declare and code questionsAndAnswers() Method to display Question
//  and let the User Answer it and then Call checkAnswer() method
public static void questionsAndAnswers() {
    // 1) Call question1() Method and store the answer in variable answer
  answer = question1();
  answer = checkForValidAnswer(answer);

 // 2) Call checkAnswer1() Method passing the parameter (answer)
  checkAnswer1(answer);
}
```



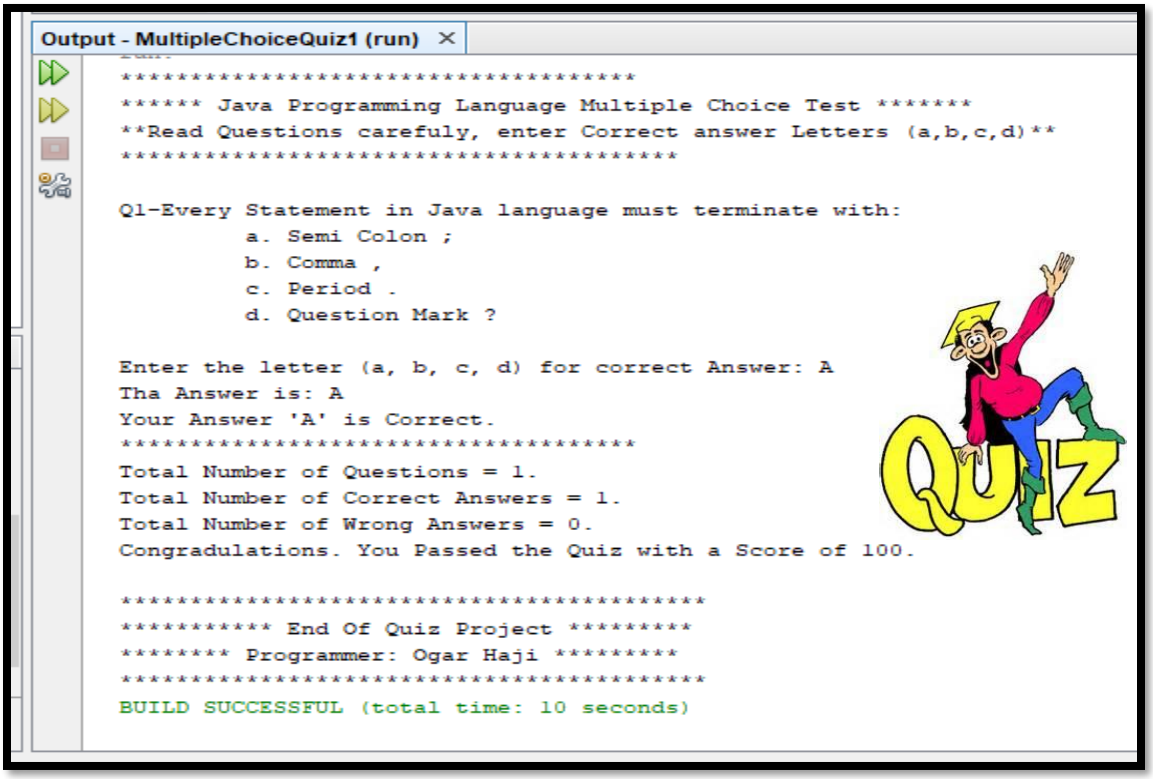
```
// D) Declare and Code question1() Method which contains Question1
public static String question1() {
    // 1) Display the Question 1 on the Console
    System.out.println("Q1-Every Statement in Java language must" +
                       " terminate with: ");
    System.out.println(" \t a. Semi Colon ; ");
    System.out.println(" \t b. Comma , ");
    System.out.println(" \t c. Period . ");
```

```
        System.out.println(" \t d. Question Mark ? ");
        System.out.println();
        // 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
          System.out.print("Enter the letter (a, b, c, d) for correct Answer: ");
        // 3) Read the User Answer from the console
          answer = input.nextLine();
          System.out.println("Tha Answer is: "+ answer);
          return answer;    // return the answer to the calling method
    }
// B) Define & Code checkForValidAnswer() Method
public static String checkForValidAnswer(String Answer) {
      while ( ! ( answer.equalsIgnoreCase("A") )
                || ( answer.equalsIgnoreCase("B") )
                || ( answer.equalsIgnoreCase("C") )
                || ( answer.equalsIgnoreCase("D") )  ) {
      // 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
     System.out.println("********** Invalid Answer ********** " );
   System.out.print("Please, Enter the letter (a, b, c, d) for correct Answer: ");
// 3) Read the User Answer from the console
     answer = input.nextLine();
   }
   return answer;
}
    // F) Declare CheckAnswer1() Method to check Answer1 submitted by user
    public static void checkAnswer1(String answer)  {
        // 1) Check if the answer 1 is Correct Answer 'a'
        if (answer.equalsIgnoreCase ("a") ) {
```

```
            System.out.printf("Your Answer '%s' is Correct.", answer);
            correctAnswersCount++;
        }
        else {
            System.out.printf("Your Answer '%s' is Wrong.", answer);
            wrongAnswersCount++;
        }
        System.out.println();
        System.out.println("****************************************");
    }
// ) Define and  Code calculateStudentScore() Method
public static void calculateStudentScore() {
        // 1) Find total number of questions
        numberOfQuestions = correctAnswersCount + wrongAnswersCount;
        System.out.printf("Total Number of Questions = %d. %n",
                                                    numberOfQuestions);
        System.out.printf("Total Number of Correct Answers = %d. %n",
                                                    correctAnswersCount);
        System.out.printf("Total Number of Wrong Answers = %d. %n",
                                                     wrongAnswersCount);
    // 2)    Find student Score on the Test.
    //  score = (100 / numberOfQuestions) * correctAnswersCount;
    score = (100 / numberOfQuestions) * correctAnswersCount;
    // 3) Check if the student is passing
    if (score >= 70) {
       System.out.printf("Congradulations. You Passed the Quiz" +
            " with a Score of %d. %n", score );
    }
    else {
       System.out.printf("Sorry. You Failed the Quiz" +
            " with a Score of %d. %n", score);
    }
```

```
    }
    // H) Declare and Code printFooters() Method to print Footers of Project
    public static void printFooters() {
        System.out.println();
        System.out.println("*****************************************");
        System.out.println("********** End Of Quiz Project *********");
        System.out.println("******** Programmer: Ogar Haji *********");
        System.out.println("*****************************************");
    }
} // End of Class
```

The Output of this Java project will look like the following:



Chapter 13 + Java Lab Assignment #13 (Due Next Week) 100 Points

Name: _____ **CIS144 Java language + Wright College**

Multiple Choice Quiz + Using Arrays + Java Project (Do Lab Exercise)+++ (DUPLICATE)

Do Lab Exercise + Lab 9

Null	A	B	C
0	**1**	**2**	**3**

correctAnswersArray

Null	A	B	D
0	**1**	**2**	**3**

studentAnswersArray

This is a **multiple-choice quiz using Arrays** Java project. It will prompt the user to enter his/her Id, First Name and Last Name. It will display the First Question and prompt the user to enter only options ‘a’, ‘b’, ‘c’ or ‘d’ and it will check for valid answer (a, b, c, d) and will repeat it until user enters a valid answer.

Once the student answers the first question, the project will store the result in the **studenAnswersArray**. Notice that the first element in the array Index[0] is Null. After the student finishes answering all the questions, then the Java project will method **gradeStudentAnswers** in which will use a for statement to compare the student answers array with the correct answers array. I will also store the Student Id, First Name, Last Name, Score and whether is Passing or Failing in an Array **studentNamesAndScoresArray** and at the End, the Java project will print out the List of Student Name in the **studentNamesAndScoresArray.**

1) Get into NetBeans IDE and type the following java project.

2) Save As MultipleChoiceQuizUsingArraysLab13

```
/*      Write your Comments here
```

Do Lab Exercise

Date: Friday, November 17, 2017

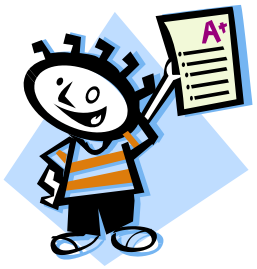
Programmer: Ogar Haji

Project Name: Multiple Choice Quiz using Arrays

*/

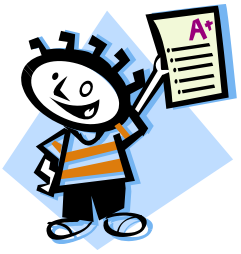
// 1) Import the Classes needed for the project

```
package multiplechoicequizUsingArrays;
import java.util.Scanner;

public class MultipleChoiceQuizUsingArrays {
```

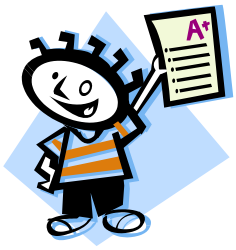


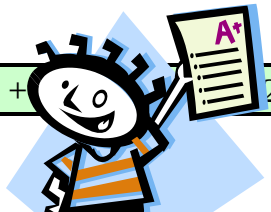
// 1) Declare the global variable answer to be used in the project

```
    public static String answer = "";
    private static  String[] correctAnswersArray = {" ","A","B","C"};
    private static String[] studentAnswersArray = new String[4];
    private static String[] studentScoresArray = new String[4];
    private static String[] studentNamesAndScoresArray = new String[4];
    private static String studentId = "";
    private static String firstName = "";
    private static String lastName = "";
    private static String studentIdAndFullName = "";
    private static int indexId = 0;
    private static String studentStatus = "";
    private static String studentInfo = "";
    private static int q;  // q means question
    public static String moreStudents = "Y";
```

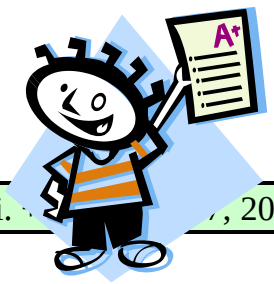


```
    public static int correctAnswersCount = 0;
    public static int wrongAnswersCount = 0;
    public static int numberOfQuestions = 0;

public static void main(String[] args) {
   // A) Call the Methods from the main() Method
    // 1) Call print Headers method to print the headings of project
     printHeaders();

    // Use while loop to prompt the user if there are more students
     while (moreStudents.equalsIgnoreCase ("Y"))  {
         // initialize counter variables to 0 for new student
        correctAnswersCount = 0;
        wrongAnswersCount = 0;
        numberOfQuestions = 0;

        Scanner input = new Scanner(System.in);

       // 2) Call Questions and Answers method to display Questions and Answers
        promptForStudentName();
        questionsAndAnswers();

        gradeStudentAnswers();
        calculateStudentScore();
         System.out.print("Do you have more Students to Check Grades (Y/N): ");
        moreStudents = input.nextLine();
     }

    // call the Student Name and Score arrays
     printStudentNamesAndScoreArrays();
```

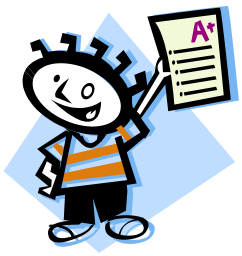




```
    //  Call print Footers method to print the Footers of project
      printFooters();

} // End of main() method

    // B) Define and Code PrintHeaders() Method to print the Headings of Project
    public static void printHeaders() {
        // 1) Print the Headings of the project
        System.out.println("*****************************************");
        System.out.println("****** java Programming Language" +
                           " Multiple Choice Test *******");
        System.out.println("**Read Questions carefuly, enter Correct answer" +
                           " Letters (a,b,c,d)**");
        System.out.println("*****************************************");
        System.out.println();
    }

// C) Define and Code promptForStudentName() method
    public static void promptForStudentName() {
        Scanner input = new Scanner(System.in);

        // increment the index id each time you enter new student
        indexId++;


        //  Prompt the user to Enter Student ID
        System.out.print("Please, Enter Student ID: ");

        // Read Student ID and Store it in studentId variable
```



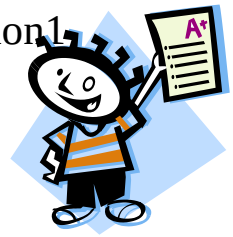
```
    studentId = input.nextLine();

    // Prompt the user to Enter Student First Name
    System.out.print("Please, Enter Student First Name: ");

    // Read Student First Name and Store it
    firstName = input.nextLine();

    // Prompt the user to Enter Student Last Name
    System.out.print("Please, Enter Student Last Name: ");
```



```
    // Read Student First Name and Store it
    lastName = input.nextLine();

    // Combine Student Id, a tab, firstName, a space and lastName
    studentIdAndFullName = studentId + "\t" + firstName + " " + lastName;
}

// D) Define and code questionsAndAnswers() Method to display Question and
// let the User Answer it and then Call checkAnswer() method
public static void questionsAndAnswers() {
// 1) Call question1() Method and store the answer in variable answer
  answer = question1();
  answer = checkForValidAnswer(answer);
```


```
// 3) Call question2() Method and store the answer in variable answer
  answer = question2();
  answer = checkForValidAnswer(answer);
// 5) Call question2() Method and store the answer in variable answer
  answer = question3();
```

```
    answer = checkForValidAnswer(answer);
}
```



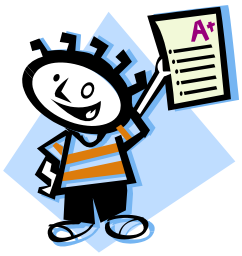
// E) Define and Code question1() Method which contains Question1

```
public static String question1() {
    Scanner input = new Scanner(System.in);
    // 1) Display the Question 1 on the Console
    System.out.println("Q1-Every Statement in Java language must" +
                                    " terminate with: ");
    System.out.println(" \t a. Semi Colon ; ");
    System.out.println(" \t b. Comma , ");
    System.out.println(" \t c. Period . ");
    System.out.println(" \t d. Question Mark ? ");
    System.out.println();

    // 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
     System.out.print("Enter the letter (a, b, c, d) for correct Answer: ");

    // 3) Read the User Answer from the console
     answer = input.nextLine();
     q = 1;
     studentAnswersArray[q] = answer;
     return answer;    // return the answer back to the calling method
}
```

// F) Define Question2() Method which contains Question2

```
public static String question2() {
    Scanner input = new Scanner(System.in);

   // 1) Display the Question 2 on the Console
   System.out.println("Q2 - Block Comments in Java language are: ");
```

```
System.out.println(" \t a. // ..... // ");
System.out.println(" \t b. /* ..... */ ");
System.out.println(" \t c. ?? ..... ?? ");
System.out.println(" \t d. ##..... ## ");
System.out.println();
```

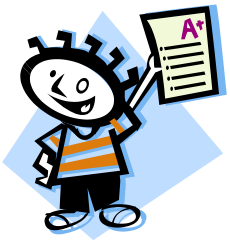


```
// 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
System.out.print("Enter the letter (a, b, c, d) for correct Answer: ");
// 3) Read the User Answer from the console
answer = input.nextLine();
q = 2;
studentAnswersArray[q] = answer;
return answer;   // return the answer to the calling method
}
```


```
// G) Define and code Question3() Method which contains Question2
public static String question3() {
    Scanner input = new Scanner(System.in);
    // 1) Display the Question 2 on the Console
System.out.println("Q3 - One of the following is NOT a valid variable name in Java: ");
System.out.println(" \t a. hoursWorked ");
System.out.println(" \t b. HoursWorked ");
System.out.println(" \t c. hours %  & Worked ");
System.out.println(" \t d. hours_worked ");
System.out.println();
```


```
// 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
```

```
      System.out.print("Enter the letter (a, b, c, d) for correct Answer: ");

      // 3) Read the User Answer from the console
      answer = input.nextLine();

      q = 3;
      studentAnswersArray[q] = answer;
      return answer;   // return the answer to the calling method
     }
```



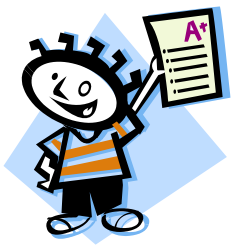
```
// H) Define and code checkForValidAnswer
public static String checkForValidAnswer(String Answer) {
    Scanner input = new Scanner(System.in);

  while ( !((answer.equalsIgnoreCase("A"))
            || (answer.equalsIgnoreCase("B"))
            || (answer.equalsIgnoreCase("C") )
            || (answer.equalsIgnoreCase("D")) ))
       {
      // 2) Prompt the User to Enter the letter (a, b, c, d) for Correct Answer
     System.out.println("********** Invalid Answer ********** " );
    System.out.print("Please, Enter the letter (a, b, c, d) for correct Answer: ");

     // 3) Read the User Answer from the console
     answer = input.nextLine();
   }

  return answer;
}
```


```
// I) Define, code CheckAnswer1() Method to check the Answer1 submitted by user
```


```
public static void checkAnswer1(String answer) {
    // 1) Check if the answer 1 is Correct Answer 'a'
    if (answer.equalsIgnoreCase ("a") ) {
       System.out.printf("Your Answer '%s' is Correct.", answer);
    }
    else {
       System.out.printf("Your Answer '%s' is Wrong.", answer);
    }
    System.out.println();
    System.out.println("****************************************");
}
```

// J) Declare, code CheckAnswer2() Method to check the Answer2 submitted by user

```
public static void checkAnswer2(String answer) {
    // 1) Check if the answer 2 is Correct Answer 'b'
    if (answer.equalsIgnoreCase ("b") ) {
       System.out.printf("Your Answer '%s' is Correct.", answer);
    }
    else {
       System.out.printf("Your Answer '%s' is Wrong.", answer);
    }
    System.out.println();
    System.out.println("****************************************");
}
```

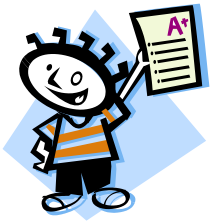


// K) Define, code CheckAnswer2() Method to check the Answer2 submitted by user

```
public static void checkAnswer3(String answer) {
```


```
  // 1) Check if the answer 3 is Correct Answer 'c'
   if (answer.equalsIgnoreCase ("c") )  {
      System.out.printf("Your Answer '%s' is Correct.", answer);
   }
   else  {
      System.out.printf("Your Answer '%s' is Wrong.", answer);
   }
   System.out.println();
   System.out.println("****************************************");
}
```

// L) Define and code calculateStudentScore() Method to calculate student Score

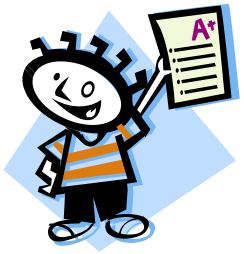



```
public static void calculateStudentScore() {
   double score = 0.0 ;
   // 1) Find total number of questions
   numberOfQuestions = correctAnswersCount + wrongAnswersCount;
   System.out.printf("Total Number of Questions = %d. %n",
                        numberOfQuestions);
System.out.printf("Total Number of Correct Answers = %d. %n",
                        correctAnswersCount);
System.out.printf("Total Number of Wrong Answers = %d. %n",
                        wrongAnswersCount);
```

// 2) Find student Score on the Test.

```
    score =  (double) ((100.00 / numberOfQuestions) * correctAnswersCount);
```


```
   // 3) Check if the student is passing
   if (score >= 70) {
      studentStatus = "Passing";
      System.out.printf("Congratulations. You Passed the Quiz" +
                                    " with a Score of %.2f %n", score );
   }
   else  {
      studentStatus = "Failing";
      System.out.printf("Sorry. You Failed the Quiz" +
            " with a Score of %.2f %n", score);
   }
    String  studentScoreAndStatus= score + "\t" + studentStatus;
    studentInfo = studentIdAndFullName + "\t" + studentScoreAndStatus;
 studentNamesAndScoresArray[indexId] = studentInfo; // Store score in array
}
  // M) Define and Code gradeStudentAnswers() Method
public static void gradeStudentAnswers() {
    System.out.println("\n**** Printing the Correct Answers Array ****");
    for (int i = 1; i < correctAnswersArray.length; i++) {
       System.out.printf("The Correct Answer for Question %d is %s. %n",
                    i,correctAnswersArray[i]);
    }
      System.out.println("\n*** Printing the Students Answers Array **");
    for (int i = 1; i < studentAnswersArray.length; i++) {
```



```
            System.out.printf("The Student Answer for Question %d is %s. %n",
                              i,studentAnswersArray[i]);
        }
      System.out.println("\n*** Printing the Graded Students Answers **");
      for (int i = 1; i < correctAnswersArray.length ; i++) {
        if (studentAnswersArray[i].equalsIgnoreCase(correctAnswersArray[i])) {
    System.out.printf("Excellent. Your Answer for Question %d is %s Correct.%n"
                              ,i , studentAnswersArray[i] );
            correctAnswersCount++;
          }
          else {
        System.out.printf("Sorry. Your Answer for Question %d is %s Wrong.%n"
                              ,i , studentAnswersArray[i] );
            wrongAnswersCount++;
          }
        System.out.printf("The Student Answer for Question %d is %s. %n",
                              i,studentAnswersArray[i]);
        }
    }
```

// N) Define and Code printNameAndScoreArrays() Method

```
  public static void printStudentNamesAndScoreArrays() {
      // Print the contents of studentNames array and Scores array
    System.out.println("The following is the List of  students in CIS144 Java language\n");
      for (String n : studentNamesAndScoresArray)  {
          System.out.printf("%s %n", n);
      }
```

```
  }
   // O) Define and Code printFooters() Method to print the Footers of Project
  public static void printFooters() {
     System.out.println();
     System.out.println("*****************************************");
     System.out.println("********** End Of Quiz Project *********");
     System.out.println("******** Programmer: Ogar Haji *********");
     System.out.println("*****************************************");
  }
}
```

The Output is like the following:

```
run:
*********************************************
****** java Programming Language Multiple Choice Test *******
**Read Questions carefully, enter Correct answer Letters (a,b,c,d)**
*********************************************

Please, Enter Student ID: 111
Please, Enter Student First Name: Mary
Please, Enter Student Last Name: Smith
Q1-Every Statement in Java language must terminate with:
        a. Semi Colon ;
        b. Comma ,
        c. Period .
        d. Question Mark ?

Enter the letter (a, b, c, d) for correct Answer: a
Q2 - Block Comments in Java language are:
        a. // ..... //
        b. /* ..... */
```

c. ?? ..... ??

d. ##..... ##

Enter the letter (a, b, c, d) for correct Answer: b

Q3 - One of the following is NOT a valid variable name in Java, :

a. hoursWorked

b. HoursWorked

c. hours % & Worked

d. hours_worked

Enter the letter (a, b, c, d) for correct Answer: c

********** Printing the Correct Answers Array ********

The Correct Answer for Question 1 is A.

The Correct Answer for Question 2 is B.

The Correct Answer for Question 3 is C.

********** Printing the Students Answers Array ********

The Student Answer for Question 1 is a.

The Student Answer for Question 2 is b.

The Student Answer for Question 3 is c.

********** Printing the Graded Students Answers ********

Excellent. Your Answer for Question 1 is a Correct.

The Student Answer for Question 1 is a.

Excellent. Your Answer for Question 2 is b Correct.

The Student Answer for Question 2 is b.

Excellent. Your Answer for Question 3 is c Correct.

The Student Answer for Question 3 is c.

Total Number of Questions = 3.

Total Number of Correct Answers = 3.

Total Number of Wrong Answers = 0.

Congratulations. You Passed the Quiz with a Score of 100.00

Do you have more Students to Check Grades (Y/N): y

Please, Enter Student ID: 222

Please, Enter Student First Name: Ogar

Please, Enter Student Last Name: Haji

Q1-Every Statement in Java language must terminate with:

a. Semi Colon ;

b. Comma ,

c. Period .

d. Question Mark ?

Enter the letter (a, b, c, d) for correct Answer: a

Q2 - Block Comments in Java language are:

a. // ..... //

b. /* ..... */

c. ?? ..... ??

d. ##..... ##

Enter the letter (a, b, c, d) for correct Answer: b

Q3 - One of the following is NOT a valid variable name in Java, :

a. hoursWorked

b. HoursWorked

c. hours % & Worked

d. hours_worked

Enter the letter (a, b, c, d) for correct Answer: d

********** Printing the Correct Answers Array ********

The Correct Answer for Question 1 is A.

The Correct Answer for Question 2 is B.

The Correct Answer for Question 3 is C.

********** Printing the Students Answers Array ********

The Student Answer for Question 1 is a.

The Student Answer for Question 2 is b.

The Student Answer for Question 3 is d.

********** Printing the Graded Students Answers ********

Excellent. Your Answer for Question 1 is a Correct.

The Student Answer for Question 1 is a.

Excellent. Your Answer for Question 2 is b Correct.

The Student Answer for Question 2 is b.

Sorry. Your Answer for Question 3 is d Wrong.

The Student Answer for Question 3 is d.

Total Number of Questions = 3.

Total Number of Correct Answers = 2.

Total Number of Wrong Answers = 1.

Sorry. You Failed the Quiz with a Score of 66.67

Do you have more Students to Check Grades (Y/N): y

Please, Enter Student ID: 333

Please, Enter Student First Name: Bobby

Please, Enter Student Last Name: Knight

Q1-Every Statement in Java language must terminate with:

a. Semi Colon ;

b. Comma ,

c. Period .

d. Question Mark ?

Enter the letter (a, b, c, d) for correct Answer: c

Q2 - Block Comments in Java language are:

a. // ..... //

b. /* ..... */

c. ?? ..... ??

d. ##..... ##

Enter the letter (a, b, c, d) for correct Answer: c

Q3 - One of the following is NOT a valid variable name in Java, :

a. hoursWorked

b. HoursWorked

c. hours % & Worked

d. hours_worked

Enter the letter (a, b, c, d) for correct Answer: c

********** Printing the Correct Answers Array ********

The Correct Answer for Question 1 is A.

The Correct Answer for Question 2 is B.

The Correct Answer for Question 3 is C.

********** Printing the Students Answers Array ********

The Student Answer for Question 1 is c.

The Student Answer for Question 2 is c.

The Student Answer for Question 3 is c.

********** Printing the Graded Students Answers ********

Sorry. Your Answer for Question 1 is c Wrong.

The Student Answer for Question 1 is c.

Sorry. Your Answer for Question 2 is c Wrong.

The Student Answer for Question 2 is c.

Excellent. Your Answer for Question 3 is c Correct.

The Student Answer for Question 3 is c.

Total Number of Questions = 3.

Total Number of Correct Answers = 1.

Total Number of Wrong Answers = 2.

Sorry. You Failed the Quiz with a Score of 33.33

Do you have more Students to Check Grades (Y/N): n

The following is the List of the students in CIS144 Java language

null

111 Mary Smith 100.0 Passing

222 Ogar Haji 66.66666666666667 Failing

333 Bobby Knight 33.333333333333336 Failing

****

********** End Of Quiz Project *********

******** Programmer: Ogar Haji *********

```
****************************************
BUILD SUCCESSFUL (total time: 2 minutes 5 seconds)
```

(Remember to Do One Modification at a Time)

1) **Modify the project to Add 2 More Questions**
2) **Find and print the Class Average.**
3) **Find and print out the Number of Passing students.**
4) **Find and print out the Number of Failing students.**
5) **Find and print out the Percentage of Passing students.**

Do Only 1 Modification at a Time

CIS144 Java Programming Language: Instructor: Ogar Haji