

Java Programming 1

PROFESSOR: CÂI FILIAULT

Topics for this week

Math
Library

Boolean
Data Type

Boolean
Conditions

Conditional
Expressions

Logical
Operators

Conditional
Operators

Marty A. Theodore

MATH TEACHER

Email

Marty A. Theodore

School Teacher

Hello,

I need a new piece of software that can do the following:

- Calculate a number to an exponent
- Calculate the base 10 logarithm of a number
- Calculate the square root of a number

-Thanks

Plan of Action

Let's build a list of what we need to complete:

1. Display the user with a menu offering them a list of possible mathematical operations
2. Allow the user to select an operation within that list
3. Have the software ask for the appropriate number of inputs needed for the calculation
4. Have the software perform the appropriate calculation
5. Output the results to the screen

Next let's add the steps into our code in comments

Steps:

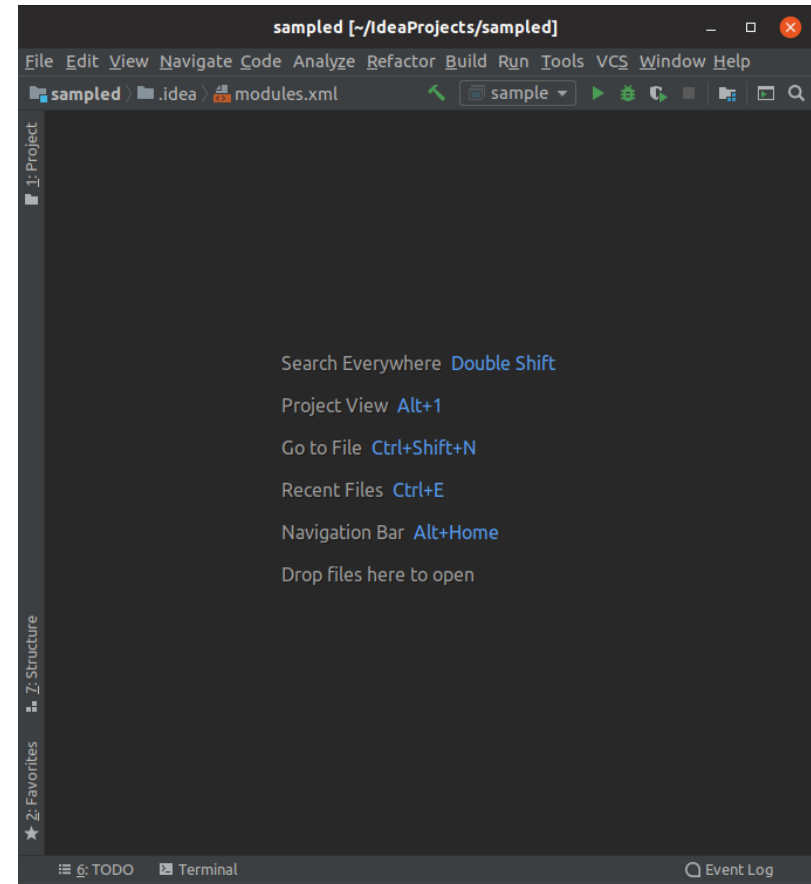
Let's build a list of what we need to complete:

1. Allow the user to input the shingle cost
2. Allow the user to input the size of the roof in square feet
3. Allow the user to input the installation cost per square feet
4. Inform the program about relevant taxes
5. Calculate two totals (one before and one after taxes)
6. Output the two totals on the console

Creating a new project

Start by opening IntelliJ and selecting the following:

File> New> Project



New Project

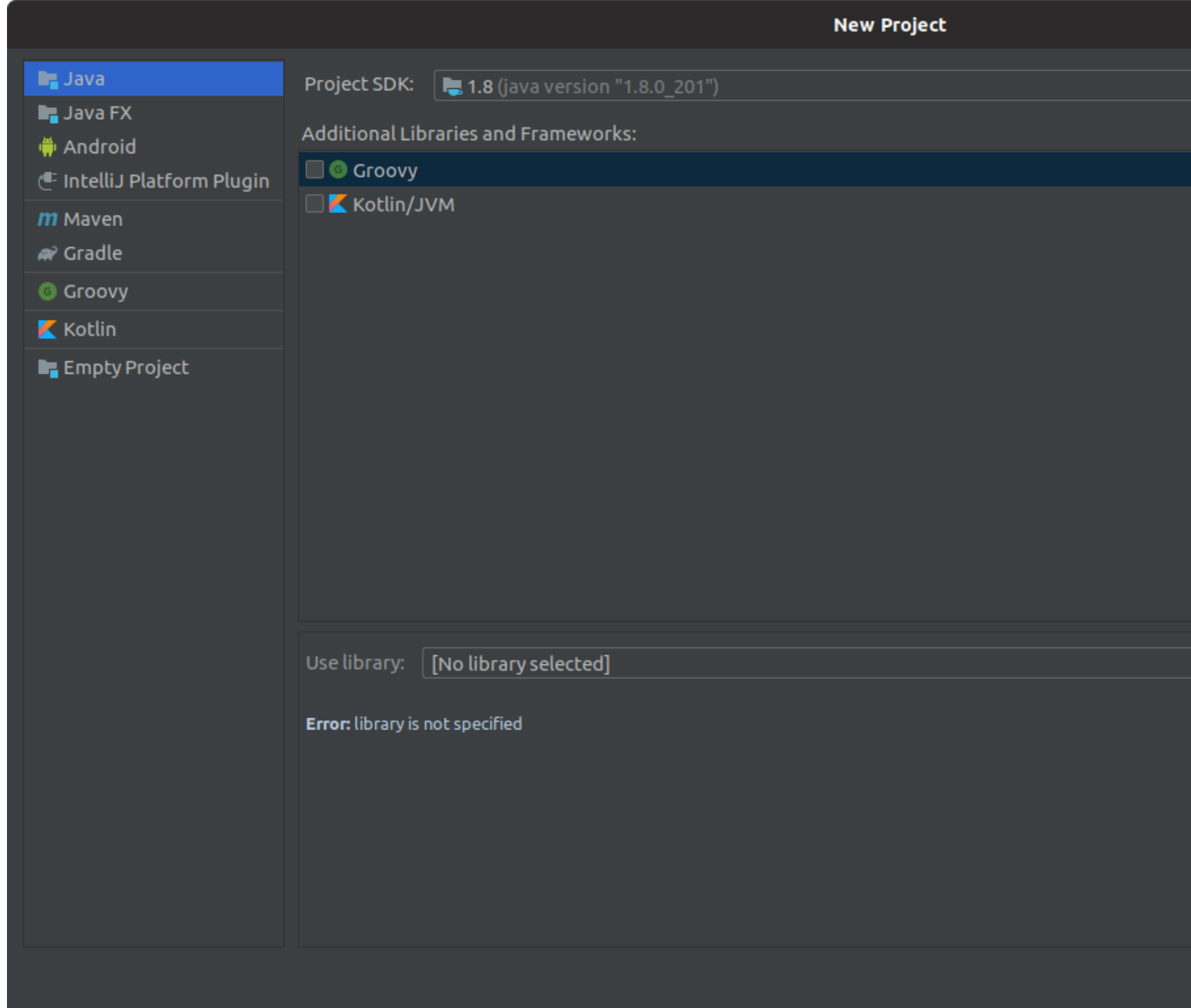
You will now see a dialog box appear *(It may be a little different than the one that I have shown)*

Starting on the left we will select a new Java Project

We will now want to select the project SDK of 1.8

SDK stands for Software Development Kit and it dictates which version of Java we are using.

At this point we can hit the next button twice



New Project

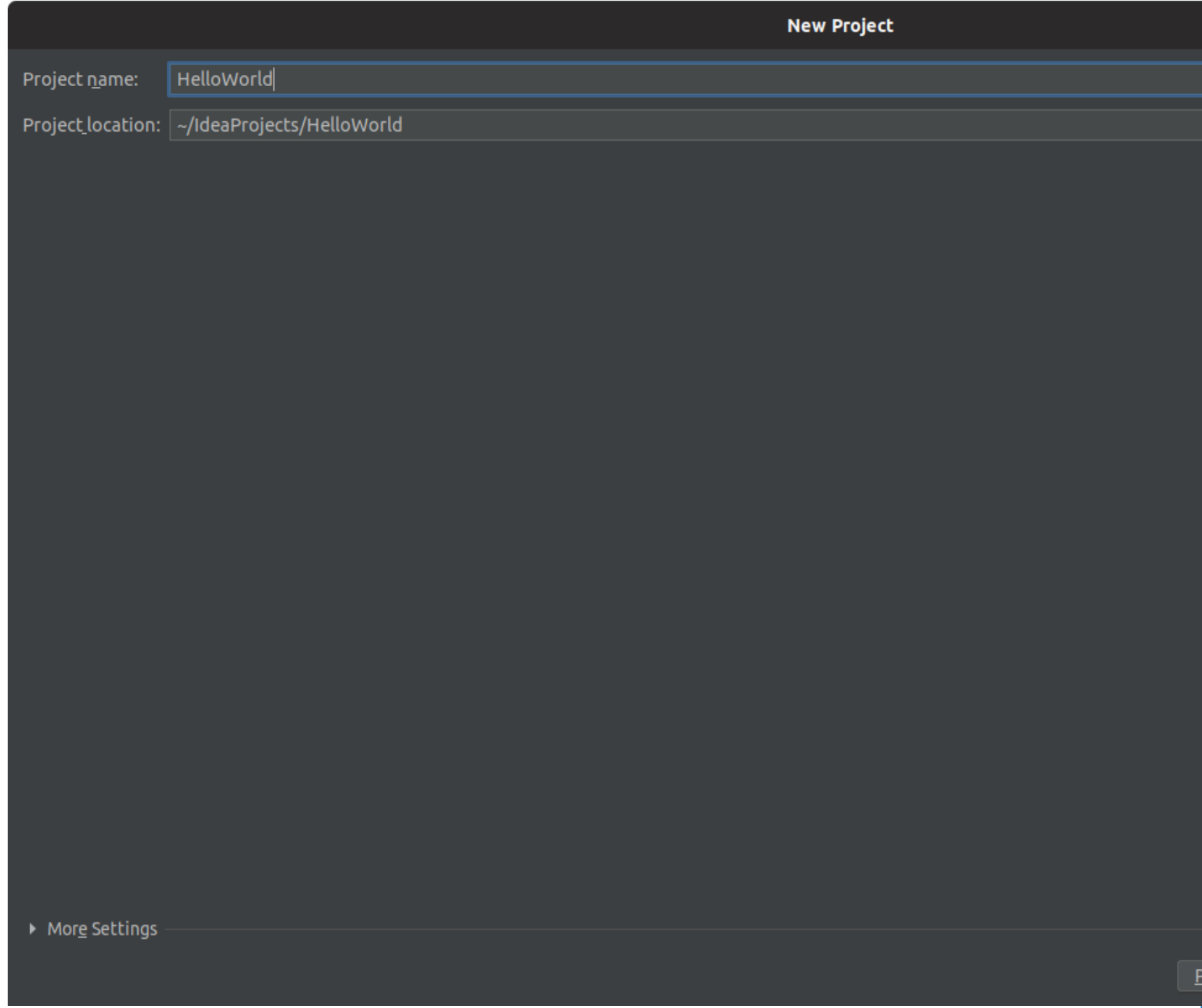
Give the project a name of:

FunctionCalculator

At this time you can change the project location to any convenient place you would like

Tip:

Keeping your programming projects on a flash/jump drive is perfectly fine. However you will want to avoid from directly working on the drive.

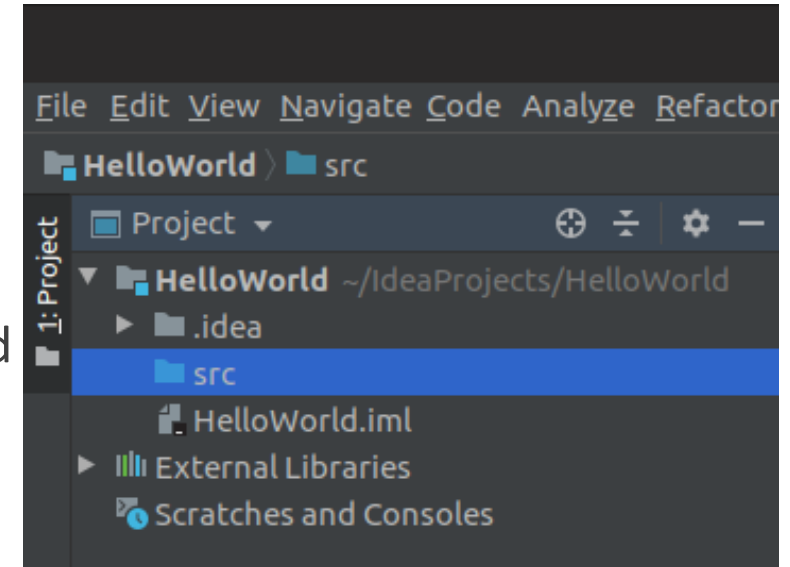


Creating a new project

You will now see in the top left-hand corner of your IDE a package explorer.

Within the package explorer you should see the **FunctionCalculator** project.

Expand the **FunctionCalculator** project to see a folder named **src**.



Creating a new class file

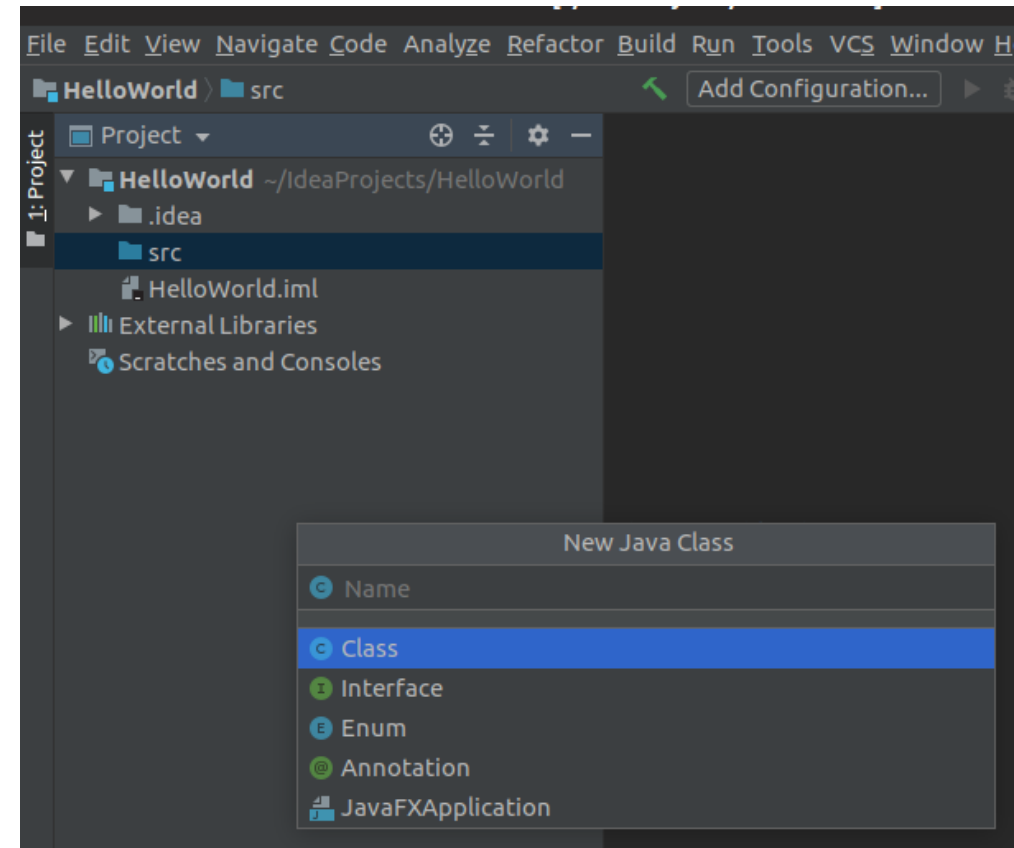
Next, we will need to create a new class file.

Right click on the **src** folder and select:

New> Java Class

Creating a new class is like creating a new program

Name the class **FunctionCalculator** and hit **enter**



Writing the program

You should now see a FunctionCalculator.java file opened on the screen.

- Refer to the image on the right to see what your file should look like
- The first thing we are going to do is type "psvm" and hit the enter key
- This is what's known as "auto-completion"
- Add the comments to match ours shown

```
 3 public class functionCalculator {  
 4  
 5     public static void main(String[] args) {  
 6         //show a menu to the user  
 7         //allow user to select item in menu  
 8         //perform appropriate action  
 9         //prompt for appropriate number of inputs  
10        //perform appropriate output  
11    }  
12  
13 }  
14
```

Writing the program

- Be sure to have a menu like the one provided on the slide.
- The next step is to allow the user to select an item within the list
- Recall in the last software we made a variable named choice and populated it using the Scanner class.
- Make an instance of the scanner class and read in an integer, store this result within a variable named choice.

```
1 package Week3;
2
3 public class functionCalculator {
4
5     public static void main(String[] args) {
6         //show a menu to the user
7         System.out.println("Welcome to the calculator app\n"
8             + "1) Calculate a number raised to a power of another number:\n"
9             + "2) Calculate logarithm 10 of a number:\n"
10            + "3) Calculate the square root of a number:\n"
11            + "Please select a number (1-3):");
12         //allow user to select item in menu
13         //perform appropriate action
14         //prompt for appropriate number of inputs
15         //perform appropriate output
16     }
17
18 }
19
```

Writing the program

Allow the user to select an item in the menu

- Line 8 creates an empty integer variable that will be used to record the user's choice
- Line 9 create an instance of the scanner class named input. This will be used to record the users input
- Finally line 17 takes the choice variable and populates it with the next integer entered by the keyboard
- The next step is to run the appropriate operation depending on what the user selected.

```
3 import java.util.Scanner;
4
5 public class functionCalculator {
6
7     public static void main(String[] args) {
8         int choice;
9         Scanner input = new Scanner(System.in);
10        //show a menu to the user
11        System.out.println("Welcome to the calculator app\n"
12            + "1) Calculate a number raised to a power of another number:\n"
13            + "2) Calculate logarithm 10 of a number:\n"
14            + "3) Calculate the square root of a number:\n"
15            + "Please select a number (1-3):");
16        //allow user to select item in menu
17        choice = input.nextInt();
18        //perform appropriate action
19        //prompt for appropriate number of inputs
20        //perform appropriate output
21    }
22
23 }
```

Writing the program

Step 3: Perform the appropriate mathematical operation

- Before we can begin to build a switch statement, we need to understand what will be executed within each case.
- How do we calculate an exponent? How would we program this?
- What about logarithm 10?
- Furthermore how do we calculate the square root of a number? Again, how would we program this?

Writing the program

The truth is we currently don't have enough knowledge about programming to answer these questions.

Writing the program

Functions:

- Functions are collections of statements that perform an operation. Through the next few slides we will explore the use of some functions and how they work.
- There is a good philosophy to follow when developing software and it “don’t reinvent the wheel”
- The easiest way to calculate exponents, squareroots and logarithms is to utilize the math library and math functions.
- Let's look at an example of the math library in use:

Writing the program

Perform the appropriate mathematical operation

The Math Library:

- Lines 19 – 22 gather the appropriate inputs needed for calculation from the user.
- Line 23 create a variable named solution and stores within it the solution produced from the Math.pow() function.
- Line 24 outputs the solution to the screen

```
7 public static void main(String[] args) {  
8     int choice;  
9     Scanner input = new Scanner(System.in);  
10    //show a menu to the user  
11    System.out.println("Welcome to the calculator app\n"  
12        + "1) Calculate a number raised to a power of another number:\n"  
13        + "2) Calculate logarithm 10 of a number:\n"  
14        + "3) Calculate the square root of a number:\n"  
15        + "Please select a number (1-3):");  
16    //allow user to select item in menu  
17    choice = input.nextInt();  
18    //perform appropriate action  
19    System.out.println("Please enter a number for calculation");  
20    double firstNumber = input.nextDouble();  
21    System.out.println(firstNumber + " to the power of: (please enter an integer):");  
22    int secondNumber = input.nextInt();  
23    double solution = Math.pow(firstNumber, secondNumber);  
24    System.out.println(firstNumber + " to the power of " + secondNumber + " evaluates to " + solution);  
25  
26  
27    //prompt for appropriate number of inputs  
28    //perform appropriate output  
29 }  
30  
31 }
```

Writing the program

- The math library allows us to utilize the Math.pow() function.
- Notice how in the Math.pow function we included two comma separated values.
- These values are known as parameters.
- Parameters are “passed” to functions and are information utilized by the function to produce the correct output.

```
7 public static void main(String[] args) {  
8     int choice;  
9     Scanner input = new Scanner(System.in);  
10    //show a menu to the user  
11    System.out.println("Welcome to the calculator app\n"  
12        + "1) Calculate a number raised to a power of another number:\n"  
13        + "2) Calculate logarithm 10 of a number:\n"  
14        + "3) Calculate the square root of a number:\n"  
15        + "Please select a number (1-3):");  
16    //allow user to select item in menu  
17    choice = input.nextInt();  
18    //perform appropriate action  
19    System.out.println("Please enter a number for calculation");  
20    double firstNumber = input.nextDouble();  
21    System.out.println(firstNumber + " to the power of: (please enter an integer):");  
22    int secondNumber = input.nextInt();  
23    double solution = Math.pow(firstNumber, secondNumber);  
24    System.out.println(firstNumber + " to the power of " + secondNumber + " evaluates to " + solution);  
25  
26  
27    //prompt for appropriate number of inputs  
28    //perform appropriate output  
29 }  
30  
31 }
```

Writing the program

Examples of functions:

- Think of how we use the `System.out.println()` function.
- We pass the function a string parameter
- Furthermore think of every time we use the `input.nextInt()`, `input.nextDouble()` etc.
- These are all examples of functions that un complicate very complex tasks.

Writing the program

Math function	Description
Math.sqrt()	Accepts a number as a parameter and calculates the square root
Math.pow	Accepts two parameters and calculates the first number to the power of the second number
Math.round	Accepts one parameter and rounds the number
Math.ceil()	Accepts one parameter and rounds the number up
Math.floor()	Accepts one parameter and rounds the number down
Math.log10	Accepts one parameter and calculates the logarithm10 of the number
Etc..	For complete listing refer to text or oracles website

Writing the program

Square root Solution:

- Line 39 calculates the square root of the number entered by the user
- The output is stored within the solution variable
- The output is produced and displayed on the screen in line 40

```
20 //perform appropriate output
21 System.out.println("Please enter a number for calculation");
22 double firstNumber = input.nextDouble();
23 double solution;
24 switch(choice){
25     case 1:
26         //calculate the power of one number to another
27         System.out.println(firstNumber + " to the power of:");
28         int power = input.nextInt();
29         solution = Math.pow(firstNumber, power);
30         System.out.println(firstNumber + " to the power of " + power + " evaluates to " + solution);
31         break;
32     case 2:
33         //calculate log10 of a number
34         solution = Math.log10(firstNumber);
35         System.out.println("Log10(" + firstNumber + ") evaluates to " + solution);
36         break;
37     case 3:
38         //calculate the square root of a number
39         solution = Math.sqrt(firstNumber);
40         System.out.println("The square root of " + firstNumber + " evaluates to " + solution);
41         break;
42     default:
43         break;
44 }
45 System.out.println("Program exiting! Goodbye");
46
47 }
48
49 }
```

Test your software

Take a moment to test your software

Marty A. Theodore

BACK AGAIN

Email

Marty A. Theodore

School Teacher

Hello,

I would like a software for my first graders that can do the following:

- Determine if one number is greater than another
- Determine if one number is less than another number
- Determine if two numbers are equal

-Thanks

Marty

Conditional Logic

- Similar to the last piece of software made for Marty, he has asked us to make a software that is able to make simple comparisons between numbers.
- The types of comparisons he would like to make can be addressed in a chart form
- The chart on the right is a list of all comparison operators we will use in Java.
- Comparison operators are usually use in conjunction with conditional logic.
- Conditional logic usually comes in the form of what is called an “if” statement

Operator	Name
>	Greater Than
<	Less Than
==	Equality
>=	Greater Than Equal To
<=	Less Than Equal To
!=	Not Equal To
!	Not

Conditional Logic

When dealing with conditional logic you often find yourself using the two words “if” and “then”

- If this is true, then do this
- If this is not true, then do this
- Or if it's not true do nothing

If statements take on many forms:

- If
- if else
- Else if

Before we begin to explain the “if”, “if else”, “else if” conditional logic lets explain Boolean values.

Conditional Logic

Boolean:

- Named after **George Boole** who invented **Boolean algebra**.
 - Boolean algebra in its simplest form is a method for representing data as true or false (1 or 0 respectively).
- Within java we have a data type called Boolean.
- Can we think of another type of computer technology that uses 1's and 0's?

Conditional Logic

If Statement

If statements operate on conditional expressions that evaluate to either true or false.

If (conditional expression){

the code inside here will only run if the conditional expression evaluates to true

}

If (hungry){

System.out.println("I am hungry");

}

if else

Else if

Conditional Logic

If Else Statement

If statements operate on conditional expressions that evaluate to either true or false.

The if else statement gives an alternative outcome for conditional situations

```
If (conditional expression){
```

```
    the code inside here will only run if the conditional expression evaluates to true
```

```
}
```

```
else{
```

```
    the code inside here will only run if the conditional expression evaluates to false
```

```
}
```

Conditional Logic

Else if Statement

If statements operate on conditional expressions that evaluate to either true or false.

The if else statement gives an alternative outcome for conditional situations

Finally the else if statement is used to provide alternative conditional expressions to check if the first one fails

```
If (hot){
```

```
    System.out.println("This porridge is too hot")
```

```
}
```

```
else if (warm){
```

```
    System.out.println("This porridge is just right"); }
```

```
}
```

```
else if (cold){
```

```
    System.out.println("This porridge is too cold"); }
```

```
}
```

Conditional Logic

Expression	Logical Operator	Result
(1 == 2)	?	?
(1 != 2)	?	?
(5 > 5)	?	?
(5 < 6)	?	?
(3 >= 0)	?	?
(4 <= 4)	?	?

Conditional Logic

Expression	Logical Operator	Result
(1 == 2)	Equality	False
(1 != 2)	Not equal	True
(5 > 5)	Greater than	False
(5 < 6)	Less than	True
(3 >= 0)	Greater than equal to	True
(4 <= 4)	Less than equal to	True

Conditional Logic

We can see that each conditional expression can be made into an “if” statement

Whether the inside of the if statement is run or not is determined by the result of the conditional expression that resides within it.

Expression	Result
<code>if(1 == 2){ }</code>	False
<code>if(1 != 2){ }</code>	True
<code>if(5 > 5){ }</code>	False
<code>if(5 < 6){ }</code>	True
<code>if(3 >= 0){ }</code>	True
<code>if(4 <= 4){ }</code>	True

Creating a new class file

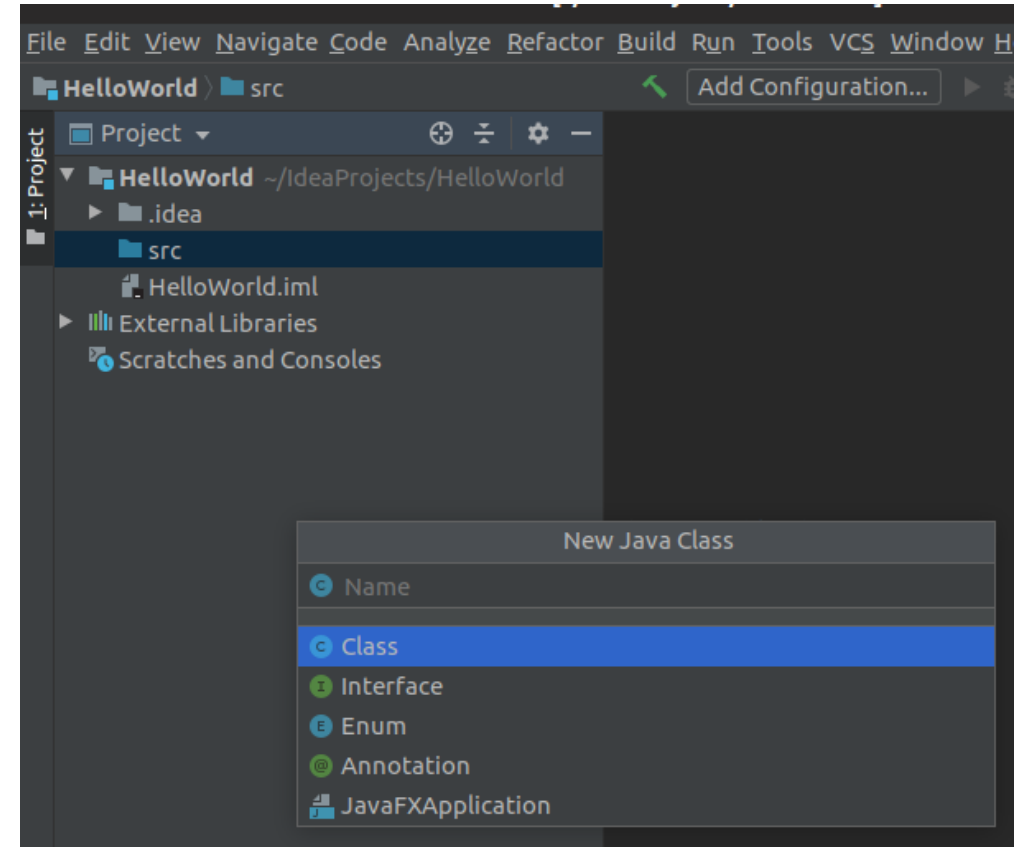
Next, we will need to create a new class file.

Right click on the **src** folder and select:

New> Java Class

Creating a new class is like creating a new program

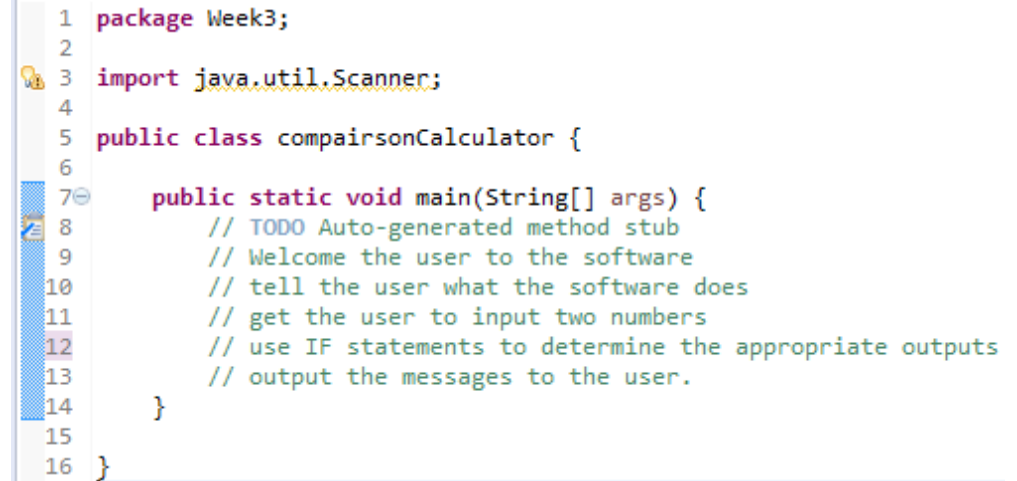
Name the class **ComparisonCalculator** and hit **enter**



Writing the program

You should now see a ComparisonCalculator.java file opened on the screen.

- Refer to the image on the right to see what your file should look like
- The first thing we are going to do is type "psvm" and hit the enter key
- This is what's known as "auto-completion"
- Add the comments to match ours shown



```
1 package Week3;
2
3 import java.util.Scanner;
4
5 public class compairsonCalculator {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         // Welcome the user to the software
10        // tell the user what the software does
11        // get the user to input two numbers
12        // use IF statements to determine the appropriate outputs
13        // output the messages to the user.
14    }
15
16 }
```

Writing the program

Line number 10 and 12-14 introduce the user:

- Introduce the user to the program by adding the `System.out.println` as seen in line number 10
- We then tell the user what the software is about to do.
- The next step is to retrieve the input of two numbers from the user
- We will need to prompt the user to enter two numbers. Then implement the scanner class and create variables to store the information gathered from the scanner class.

```
1 package Week3;
2
3 import java.util.Scanner;
4
5 public class compairsonCalculator {
6
7     public static void main(String[] args) {
8         // TODO Auto-generated method stub
9         // Welcome the user to the software
10        System.out.println("Welcome, this program makes the following comparisons");
11        // tell the user what the software does
12        System.out.println("1) if one number is larger than another\n"
13                           + "2) if one number is smaller than another\n"
14                           + "3) if two numbers are the same\n");
15        // get the user to input two numbers
16        // use IF statements to determine the appropriate outputs
17        // output the messages to the user.
18    }
19
20 }
```

Writing the program

Line number 25 creates an IF statement that checks to see whether input1 is greater than input2 or not. Then outputs the correct output.

Line number 32 creates an IF statement that checks to see whether input 1 is smaller than input2 or not. Then outputs the correct output.

Line number 38 creates an IF statement that checks to see if the two numbers (input1 and input2) are equal. Then outputs the correct output. value is true.

```
21 System.out.println("Please enter the second number for calculation:");
22 input2 = in.nextDouble();
23 // use IF statements to determine the appropriate outputs
24 // output the messages to the user.
25 if(input1 > input2){
26     System.out.println("" + input1 + " is larger than " + input2);
27 }
28 else{
29     System.out.println("" + input1 + " is not larger than " + input2);
30 }
31
32 if(input1 < input2){
33     System.out.println("" + input1 + " is smaller than " + input2);
34 }
35 else{
36     System.out.println("" + input1 + " is not smaller than " + input2);
37 }
38 if(input1 == input2){
39     System.out.println("" + input1 + " is equal to " + input2);
40 }
41 else{
42     System.out.println("" + input1 + " is not equal to " + input2);
43 }
44 }
45
46 }
```

Test your software

Take a moment to test your software

GNB Bank

NEW CLIENT

Email

GNB Bank

Local company

Giant National Bank (GNB)

Hello,

I would like a software that can do the following:

- Allow the user to enter the amount they would like to borrow and their credit rating.
- The software will then output the interest rate they can be approved for

-Thanks

Jâc

Information

A couple details talked about with the client

1. The company wants an online quoting software to let the client know whether they are eligible for a loan from them.
2. The client has provided us with all the necessary information to build the software:

More Information

The Bank has informed us it allows the user to borrow in the following increments of money

- \$10,000, \$20,000, \$30,000, \$40,000, and \$50,000

When borrowing money the bank needs to determine the interest rate, the following is used to determine the interest rate:

- Prime is currently 2%:
- Excellent credit rating is always prime + 0
- Good credit rating is prime + 0.5% for < 30,000 borrowed otherwise its prime + 1%
- Fair credit rating is prime + 1% for < 30,000 borrowed otherwise its prime + 2%
- Poor credit rating is prime + 2% for < 30,000 borrowed otherwise its prime + 5%
- Bad credit rating is prime + 5% for < 30,000 borrowed otherwise the application is denied

More Information

The chart provided is the credit rating ranges that a person's credit rating can be in.

We have more than enough information provided to build the desired software.

Let's break down the program down into steps. Then learn how to use the if statement as we go.

1. We will need to introduce the user to the program
2. Gather the amount of money the user wishes to borrow and their current credit rating
3. Calculate the interest rate the user will pay based on prime, credit score and amount borrowed
4. Display whether the user will receive a loan or not and if they do then how much the user will pay over 5 and 10 years.

Credit Rating	Range
Excellent	781 - 850
Good	661 - 780
Fair	601 - 660
Poor	501 - 600
Bad	Below 500

Creating a new class file

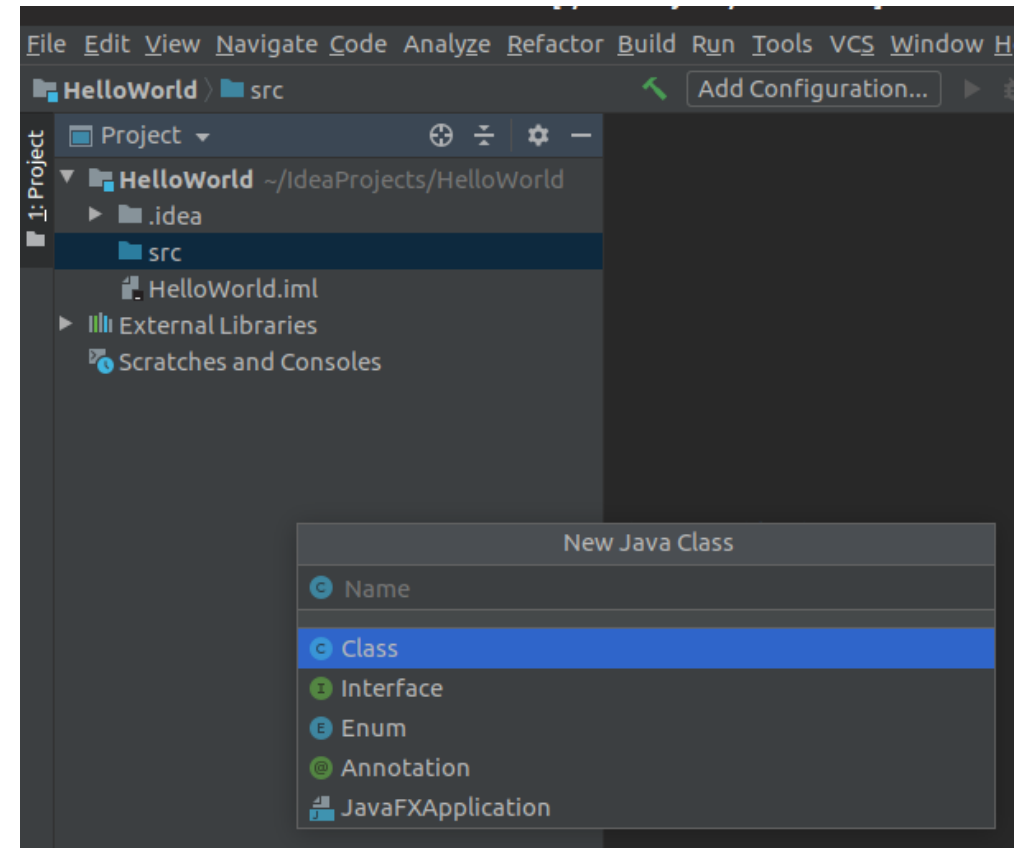
Next, we will need to create a new class file.

Right click on the **src** folder and select:

New> Java Class

Creating a new class is like creating a new program

Name the class **LoanEstimator** and hit **enter**



Writing the program

You should now see a LoanEstimator.java file opened on the screen.

- Refer to the image on the right to see what your file should look like
- The first thing we are going to do is type "psvm" and hit the enter key
- This is what's known as "auto-completion"
- Add the comments to match ours shown

```
1 package Week3;
2
3 public class LoanEstimator {
4     public static void main(String[] args) {
5         //Introduce the user to the program
6         //Gather inputs
7         //Calculate the interest rate |
8         //Display appropriate outputs
9     }
10 }
11
```

Writing the program

Line number 6 and 7 introduce the user:

- Introduce the user to the program by adding the `System.out.println` as seen in line number 6 and 7
- Notice how we prompt the user to enter the amount of money they would like to borrow
- We will need to implement the scanner class and create variables to store the information gathered from the scanner class.

```
1 package Week3;
2
3 public class LoanEstimator {
4     public static void main(String[] args) {
5         //Introduce the user to the program
6         System.out.println("Welcome to GNB loan quoting software:\n"
7             + "Please enter the amount you wish to borrow: ");
8         //Gather inputs
9         //Calculate the interest rate
10        //Display appropriate outputs
11    }
12 }
13
```

Writing the program

Gathering Inputs:

- Line 7 and 8 creates variables to store the users credit score and amount of money they want to borrow.
- Line 9 creates an instance of the scanner input so it can gather information from the keyboard
- Line 14 reads the amount of money the user wishes to borrow and stores it inside the loan amount variable
- Line 15 and 16 prompt the user to input their credit score and stores it within the credit score variable

```
1 package Week3;
2
3 import java.util.Scanner;
4
5 public class LoanEstimator {
6     public static void main(String[] args) {
7         Double loanAmount;
8         int creditScore;
9         Scanner input = new Scanner(System.in);
10        //Introduce the user to the program
11        System.out.println("Welcome to GNB loan quoting software:\n"
12            + "Please enter the amount you wish to borrow: ");
13        //Gather inputs
14        loanAmount = input.nextDouble();
15        System.out.println("Please enter your current credit score:");
16        creditScore = input.nextInt();
17        //Calculate the interest rate
18        //Display appropriate outputs
19    }
20 }
```

Edit: Please note double should be spelled with a lowercase d

Writing the program

Excellent credit rating is always prime + 0

Excellent (781 – 850)

Line number 7 creates a final double prime with the number 2.

The final keyword introduces a new concept called a constant. A constant is the opposite of a variable in the sense that the constant is a variable that does not change. Once assigned a value the constant cannot change within the program.

Line number 10 creates a new variable named interest that will store the interest rate

Lines 20 - 22 uses an if statement to set the value of interest rate.

If the credit score is greater than or equal to 781 then the interest rate will be equal to prime plus 0.

The program will first calculate (creditScore >= 781), this will either be true or false. The inside of the if statement will only be run if the value is true.

```
1 package Week3;
2
3 import java.util.Scanner;
4
5 public class LoanEstimator {
6     public static void main(String[] args) {
7         final double prime = 2;
8         Double loanAmount;
9         int creditScore;
10        double interest;
11        Scanner input = new Scanner(System.in);
12        //Introduce the user to the program
13        System.out.println("Welcome to GNB loan quoting software:\n"
14            + "Please enter the amount you wish to borrow: ");
15        //Gather inputs
16        loanAmount = input.nextDouble();
17        System.out.println("Please enter your current credit score:");
18        creditScore = input.nextInt();
19        //Calculate the interest rate
20        if(creditScore >= 781){
21            interest = prime + 0;
22        }
23        //Display appropriate outputs
24    }
25 }
26
27
```

Writing the program

Good credit rating is prime + 0.5% for < 30, 000
borrowed otherwise its prime + 1%

Good (661-780)

Line number 23 is known as an else if statement, else if statements are used if the initial if statements fail. For instance, in this situation line 23 will only execute if the condition on line 20 fails. Line 23 checks to see if the value is greater than or equal to 661 (giving the range 661 – 780)

Line 24 checks if the amount borrowed is greater than 30, 000 or not.

Line 25 executes only if the condition in line 24 is true.

Line 26 is a new concept. The else statement is what executes exclusively when the if statement that proceeds it fails. Else statements must follow if statements. Line 27 will only execute if the condition on line 24 is false.

```
3 import java.util.Scanner;
4
5 public class LoanEstimator {
6     public static void main(String[] args) {
7         final double prime = 2;
8         Double loanAmount;
9         int creditScore;
10        double interest;
11        Scanner input = new Scanner(System.in);
12        //Introduce the user to the program
13        System.out.println("Welcome to GNB loan quoting software:\n"
14            + "Please enter the amount you wish to borrow: ");
15        //Gather inputs
16        loanAmount = input.nextDouble();
17        System.out.println("Please enter your current credit score:");
18        creditScore = input.nextInt();
19        //Calculate the interest rate
20        if(creditScore >= 781){
21            interest = prime + 0;
22        }
23        else if (creditScore >= 661){
24            if(loanAmount < 30000){
25                interest = prime + 0.5;
26            }
27            else{
28                interest = prime + 1;
29            }
30        }
31        //Display appropriate outputs
32    }
```

Writing the program

Using the concepts learned from the previous slides try and build the remainder of the program with the following information:

Fair credit rating is prime + 1% for < 30,000 borrowed otherwise its prime + 2%

(Credit Score Fair 601 – 660)

Poor credit rating is prime + 2% for < 30,000 borrowed otherwise its prime + 5%

(Credit Score Poor 501 – 600)

Bad credit rating is prime + 5% for < 30,000 borrowed otherwise the application is denied

(Credit Score below 500)

Writing the program

Line numbers 31 – 38 uses else if statements to check the ranges from 601 – 660. The “nested if”* checks to see if the loan amount is less than 30,000. The if statements then assigns the appropriate interest rate.

Line numbers 39 - 46 uses else if statements to check the ranges from 501 – 600. the nested if statement checks to see if the loan amount is less than 30,000 and assigns the appropriate interest rate.

Line numbers 47 - 54 uses an else if statement to check if the score is less than 500. the nested if statement checks if the loan is less than 30, 000. then approves or denies the loan.

```
25         interest = prime + 0.5;
26     }
27     else{
28         interest = prime + 1;
29     }
30 }
31 else if (creditScore >= 601){
32     if(loanAmount < 30000){
33         interest = prime + 1;
34     }
35     else{
36         interest = prime + 2;
37     }
38 }
39 else if (creditScore >= 501){
40     if(loanAmount < 30000){
41         interest = prime + 2;
42     }
43     else{
44         interest = prime + 2;
45     }
46 }
47 else if (creditScore < 500){
48     if(loanAmount < 30000){
49         interest = prime + 5;
50     }
51     else{
52         System.out.println("Loan Denied (Too high of risk)");
53     }
54 }
```

Writing the program

Display appropriate outputs:

Now that we have the correct calculations in place we need to:

Display whether the user will receive a loan or not and if they do then how much the user will pay over 5 and 10 years.

To complete this task we will need to know whether the user has been approved for a loan or not.

Create a Boolean variable named `loanApproved` and set its default value to `true`.

Then on line 53 set the variable to `false` as this is the only instance in which the loan is denied.

```
32     else if (creditScore >= 601){
33         if(loanAmount < 30000){
34             interest = prime + 1;
35         }
36         else{
37             interest = prime + 2;
38         }
39     }
40     else if (creditScore >= 501){
41         if(loanAmount < 30000){
42             interest = prime + 2;
43         }
44         else{
45             interest = prime + 2;
46         }
47     }
48     else if (creditScore < 500){
49         if(loanAmount < 30000){
50             interest = prime + 5;
51         }
52         else{
53             System.out.println("Loan Denied (Too high of risk)");
54             loanApproved = false;
55         }
56     }
57 }
58 //Display appropriate outputs
59 }
60 }
```

Writing the program

Display appropriate outputs:

Finally on line 58 check to see if the loan has been approved.

If the loan has been approved, then output the final calculations which will show the amount the of interest that will be paid over 5- and 10-year durations. .

```
40     else if (creditScore >= 501){
41         if(loanAmount < 30000){
42             interest = prime + 2;
43         }
44         else{
45             interest = prime + 2;
46         }
47     }
48     else if (creditScore < 500){
49         if(loanAmount < 30000){
50             interest = prime + 5;
51         }
52         else{
53             System.out.println("Loan Denied (Too high of risk)");
54             loanApproved = false;
55         }
56     }
57     //Display appropriate outputs
58     if(loanApproved == true){
59         System.out.println("Based on your credit rating we can offer you a " + loanAmount + " at "
60             + interest + "% interest rate");
61         double totalFiveYears = (loanAmount * (interest / 100) * 5) + loanAmount;
62         System.out.println("Total Paid Over 5 Years: " + totalFiveYears);
63         double totalTenYears = (loanAmount * (interest / 100) * 10) + loanAmount;
64         System.out.println("Total Paid Over 10 Years: " + totalTenYears);
65     }
66 }
67 }
```

Homework

Read Chapter 3 and 4 of your textbook

Next Week

- Repetition Structures
- Sentinel values
- While loop
- For loop
- Do While Loop
- Nested for loops
- Nested loops
- Boolean conditions
- Character and String manipulation