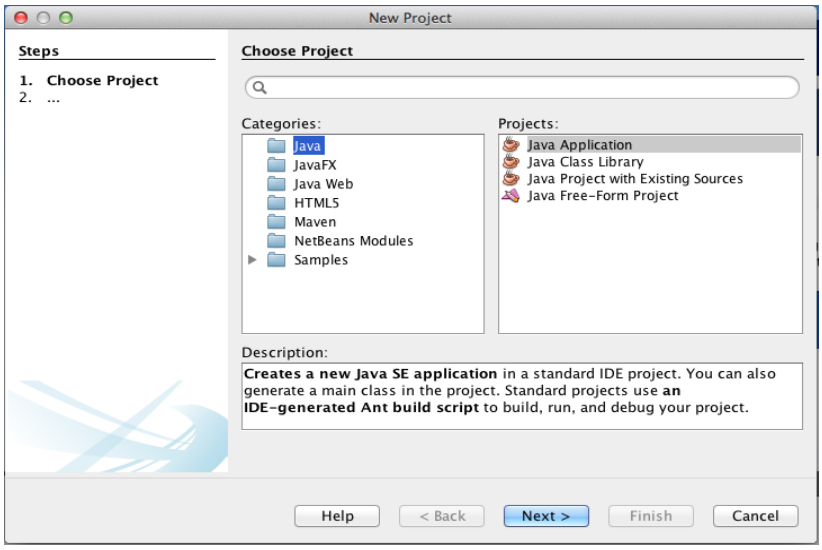
Data structure
Lab 1
Dr. Sophea PRUM
sopheaprum@gmail.com

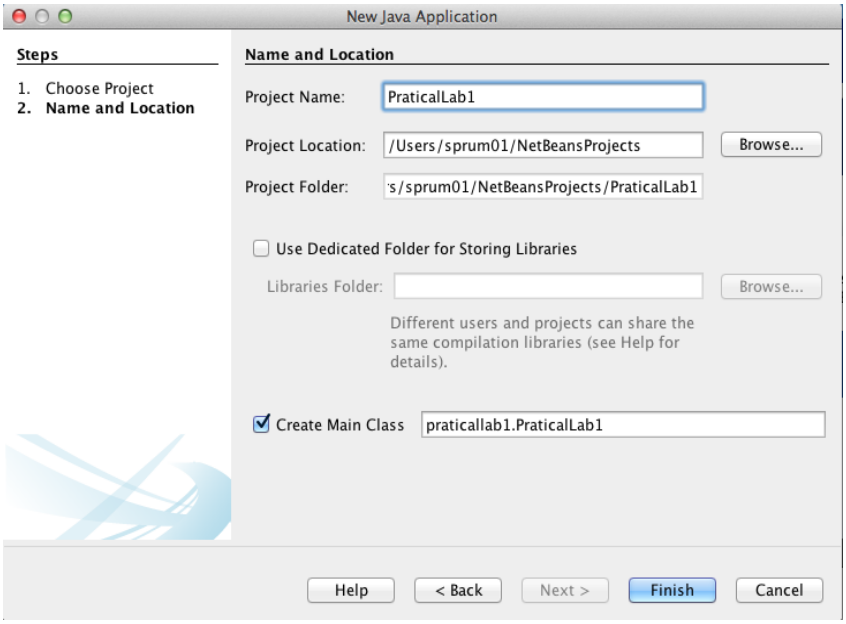
Hello world!

1. Run your NetBeans IDE

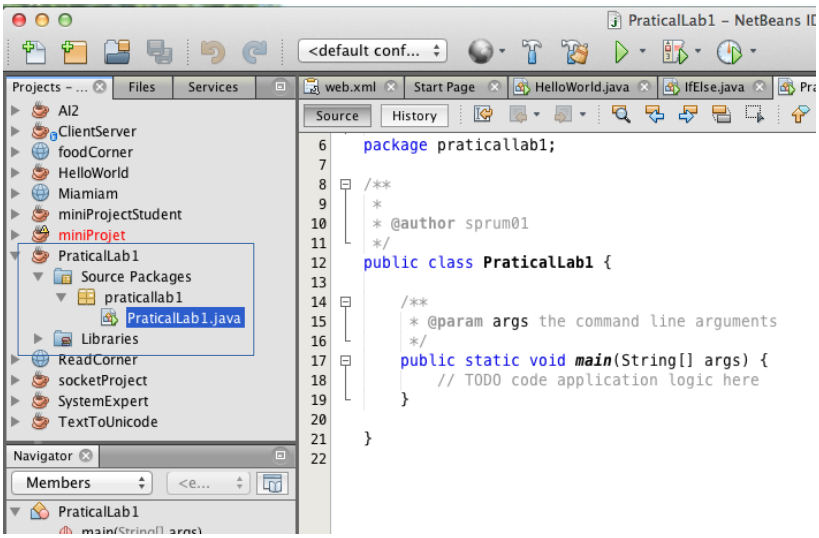
2. Create a new project: **File → New project**. In **Categories,** choose **java**. In **Projects,** choose **Java Application**



3. Enter your project name for example **PraticalLab1**. NetBeans automatically provides path to **Project Location** and **Project folder** where you can find your source code later.



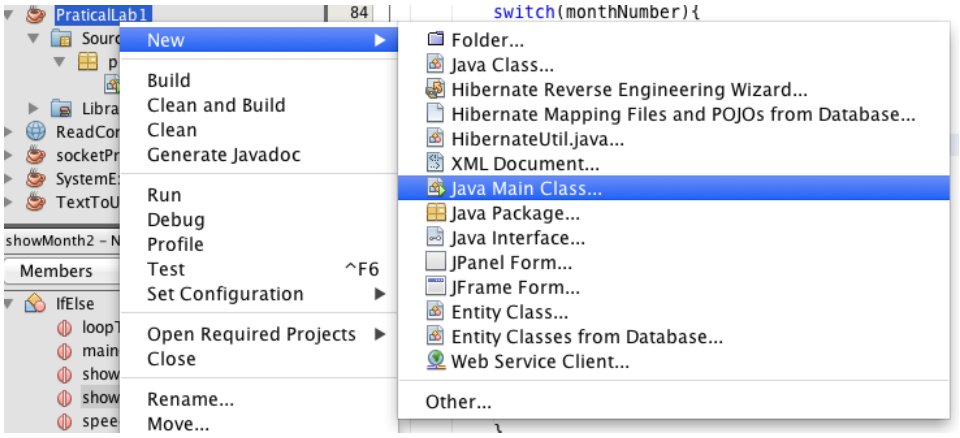
NetBeans automatically creates a java class having the same as as your project name.

Your Project 

Bravo! Your project has been created.

Todo: Add a line in your main method to print out “Hello world!”

In the following exercises, you will create one class for each exercise. In order to create a new class, right-click on your project → **new** → **java main class**



Observation, error detection and correction

Exercise 1: using if/else statement

We want to write a small program that will print the message according to the following conditions: If the user age is below 18, the system will print out “User is under 18”. If the user age is between 18 to 50, the system will print out “User is between 18 and 50”. Otherwise, the system will print out “User is above 50”. Correct the error in the code bellow.

```
String result;
int age = 10;

if(age < 18){
    result = "User is under 18";
}
else if((age >=18)||(age<=50)){
    result = "User is between 18 to 50";
}
else{
    result = "User is above 50";
}
System.out.println(result);
```

Hints: run your program with different values of variable *age* (10, 18, 50, 51, 60). Observing the results!

Exercise 2: while vs. do/while

Write and test these two blocks of code with different values of variable **initialValue** (1, 5, 10, 11). Observe and explain the difference outputs of these two blocks of code.

```
int maxValue = 10;
int initialValue = 1;

while(initialValue <= maxValue){
    initialValue++;
}

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

```
int maxValue = 10;
int initialValue = 1;

do{
    initialValue++;
}while(initialValue <= maxValue);

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

Design your own algorithm

Exercise 3: Write a program called *CheckOddEven* which prints "**Odd Number**" if the *int* variable *number* is odd, or “**Even Number**” otherwise. The program shall always print “BYE!” before exiting.

Hints: n is an even number if (n % 2) is 0; otherwise, it is an odd number.

Exercise 4: Write a program called *PrintNumberInWord* which prints "**ONE**", "**TWO**",... , "**NINE**", "**OTHER**" if the *int* variable *number* is 1, 2,... , 9, or other, respectively. Use (a) a "if/else" statement; (b) a "switch-case" statement.

Exercise 4: Write a program called **Bread/Rice/Noodle** which prints the numbers from 1 to 110, 11 numbers per line. The program will print "Bread" in place of the numbers which are divisible by 3, "Rice" for divisible by 5, "Noodle" for divisible by 7

More fun

Exercise 5: The guessing game

In this guessing game, the computer will come up with a random number between 1 and 1000. The player must then continue to guess the numbers until getting the correct number. For every guess, the computer will either say "Too high" or "Too low", and then ask for another input. At the end of the game, the number is revealed along with the number of tries to get the correct answer.

Hints: get rendom number in java:

```
Random rand = new Random(1000);
int numberToGuess = rand.nextInt();
```

get an int number of keyboard input:

```
Scanner input = new Scanner(System.in);
System.out.print("Guess a number between 1 to 1000 : ");
int guess = input.nextInt();
```

****

If a programmer codes just for fun he has all his skill.

If he codes for score his hand tremble and his breath is uneasy