# How to calculate Birthday

//New feature LocalDate in jdk 1.8

**import** java.time.LocalDate;

**import** java.time.Period;

//year, month and date are instance variable

**public** **int** calculateAge() {

**if** (year == 0 || month == 0 || date == 0) {

**return** 0;

}

**else** {

// calculate birthday

LocalDate today = LocalDate.*now*();

LocalDate birthday = LocalDate.*of*(year, month, date);

Period p = Period.*between*(birthday, today);

**return** p.getYears();

}

}

# How to input Date

Date myDate = **new** Date(1979, 12, 31);

System.***out***.println(myDate.getYear());

System.***out***.println(myDate.getMonth());

System.***out***.println(myDate.getDate());

# How to convert int to double

**double** heightInMeters = (**double**) height/100;

**return** weight/(heightInMeters\*heightInMeters);

# How to round

Int I = (int) Math.round(double i);

# Use other package

//Current package name

**package** Exercise\_3\_17;

//Other package name+Class name

**import** Exercise\_3\_16.HeartRates;

# How to get individual number

Scanner s = **new** Scanner(System.***in***);

System.***out***.println("Input thousand value:");

**int** nbr = s.nextInt();

**int** [] x={0,0,0,0};

**int** n=3;

**while** (nbr>0){

x[n]=nbr%10;

nbr/=10;

System.***out***.println(x[n]);

n--;}

# Reset value when use Nested while or for

**int** a = 1, b = 1, c = 1;

**while** (a <= 500) {

**while** (b <= 500) {

**while** (c <= 500) {

**if** (a \* a == b \* b + c \* c) {

System.***out***.print("This is Pythagoran Triples a = " + a

+ " b = " + b + " c = " + c + "\n");

}

c++;

}

c = 1;

b++;

}

b = 1;

a++;

}

# Calculate 3rd side of triangle

Math.*hypot*(a, b));

# Manually calculate exponent

**private** **static** **int** calculateIntergerPower(**int** base, **int** exponent){

**int** value = 1;

**for**(**int** i=0; i<exponent;i++){

value\*=base;

}

**return** value;

# Get integer length

int length = String.valueOf(1000).length();

# GCD

In [mathematics](http://en.wikipedia.org/wiki/Mathematics), the **greatest common divisor** (**gcd**), also known as the **greatest common factor** (**gcf**), **highest common factor** (**hcf**), or **greatest common measure** (**gcm**), of two or more [integers](http://en.wikipedia.org/wiki/Integer) (when at least one of them is not zero), is the largest positive integer that [divides](http://en.wikipedia.org/wiki/Divisor) the numbers without a [remainder](http://en.wikipedia.org/wiki/Remainder). For example, the GCD of 8 and 12 is 4.

Use Euclid’s algorithm to calculate GCD

# Convert String to Int, and convert Int to String

Integer.parseInt(String value)

Double.parseDouble(String value)

Integer value.toString()

# Use Args, multi dimentions and convert String to double

**public** **class** EnhancedFor {

/\*\*

\* **@param** args

\*/

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**if** (args.length == 0) {

System.***out***.println("Enter Double values");

} **else** {

System.***out***.println("The product is :" + *calculateProduct*(args));

}

}

**static** **double** calculateProduct(String... args) {

**double** product = 1.0;

**for** (String value : args) {

product \*= Double.*parseDouble*(value);

}

**return** product;

}

}