Primitive data types and operations

เราจะมาเรียนรู้เกี่ยวกับ Java primitive data types และเรื่องที่เกี่ยวข้อง เช่น ตัวแปร, ชนิดของตัวแปร (ชนิดของข้อมูล), operators, expressions (นิพจน์), และ input and output.

วัตถุประสงค์

- ให้รู้จักการใช้ตัวแปรเก็บข้อมูล
- ให้รู้จักคำสั่งกำหนดค่าตัวแปร (assignment statements) และ นิพจน์ที่เกี่ยวข้อง
- ให้รู้จักการใช้ค่าคงที่ (constants) เก็บข้อมูลที่มีค่าถาวร
- ให้รู้จักการประกาศ Java primitive data types: byte, short, int, long, float, double, and char
- ให้รู้จักการใช้ operators ในการเขียนนิพนจ์ที่เกี่ยวกับตัวเลข (numeric expressions)
- ให้รู้จักการใช้ตัวอักขระ (char) โดยใช้ชนิด char
- ให้รู้จักการใช้สายอักขระ (string) โดยใช้ชนิด String
- ให้รู้จักการรับข้อมูลเข้าจาก console โดยใช้ class Scanner

Identifiers

- identifier เป็นคำที่ประกอบที่ประกอบไปด้วย ตัวอักษร, ตัวเลข, ขีดเส้นใต้ (_), และ dollar signs ($).
- identifier ต้องเริ่มต้นด้วยตัวอักษร, ขีดเส้นใต้ (_), หรือ dollar sign ($) เริ่มด้วยตัวเลขไม่ได้
  - identifier เป็นคำที่ Javฤ จองไว้ (reserved word) ไม่ได้
- identifier ไม่สามารถเป็น `true`, `false`, หรือ `null`.
- identifier จะมีความยาวเท่าไหร่ก็ได้

Variables

```
// Compute the first area
radius = 1.0;
area = radius * radius * 3.14159;
System.out.println("The area is " +
  area + " for radius "+radius);

// Compute the second area
radius = 2.0;
area = radius * radius * 3.14159;
System.out.println("The area is " +
  area + " for radius "+radius);
```

การประกาศตัวแปร (Declaring Variables)

```
int x;              // ประกาศตัวแปร x ชนิด int(integer)
                    //  (เก็บค่าจำนวนเต็ม)
double radius;      // ประกาศตัวแปร radius ชนิด double
                    //  (เก็บค่าทศนิยม)
char a;            // ประกาศตัวแปร a ชนิด char (character)
                    //  (เก็บค่าอักขระ)
```

Assignment Statements (คำสั่งการกำหนดค่า)

```
x = 1;              // Assign 1 to x;

radius = 1.0;       // Assign 1.0 to radius;

a = 'A';            // Assign 'A' to a;
```

= **<u>แปลว่า นำค่าทางด้านขวามือไปใส่ไว้ในตัวแปร (หน่วยความจำ) ด้านซ้ายมือ</u>**

ประกาศ (Declaring) และ กำหนดค่าเริ่มต้น (Initializing) ใน step เดียว

- `int x = 1;`
- `double d = 1.4;`

ค่าคงที่ (Constants)

```
final datatype CONSTANTNAME = VALUE;

final double PI = 3.14159;
final int SIZE = 3;
```

ชนิดของข้อมูลชนิดตัวเลข (Numerical Data Types)

Name	Range	Storage Size
byte	-2^7 (-128) to 2^7-1 (127)	8-bit signed
short	-2^{15} (-32768) to $2^{15}-1$ (32767)	16-bit signed
int	-2^{31} (-2147483648) to $2^{31}-1$ (2147483647)	32-bit signed
long	-2^{63} to $2^{63}-1$ (i.e., -9223372036854775808 to 9223372036854775807)	64-bit signed
float	Negative range: -3.4028235E+38 to -1.4E-45 Positive range: 1.4E-45 to 3.4028235E+38	32-bit IEEE 754
double	Negative range: -1.7976931348623157E+308 to -4.9E-324 Positive range: 4.9E-324 to 1.7976931348623157E+308	64-bit IEEE 754

ตัวดำเนินการสำหรับตัวเลข (Numeric Operators)

Name	Meaning	Example	Result
+	Addition	34 + 1	35
-	Subtraction	34.0 - 0.1	33.9
*	Multiplication	300 * 30	9000
/	Division	1.0 / 2.0	0.5
%	Remainder	20 % 3	2

การหารจำนวนเต็ม (Integer Division)

+, -, *, /, and %

5 / 2 yields an integer 2.

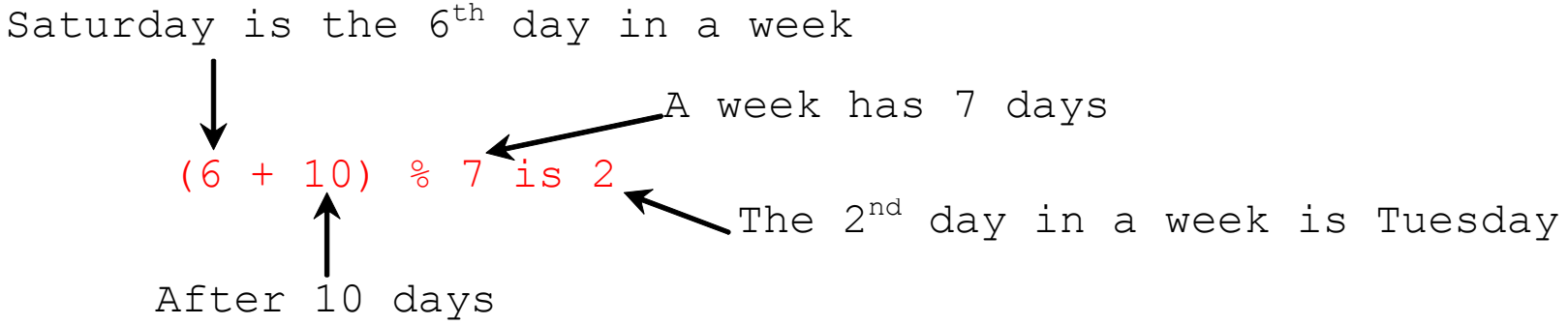
5.0 / 2 yields a double value 2.5

5 % 2 yields 1 (the remainder of the division)

การหาเศษ (Remainder Operator)

เศษจากการหารเป็นเรื่องสำคัญสำหรับการเขียนโปรแกรม เช่น

- ถ้าเราจะตรวจสอบว่า ตัวเลขเป็นเลขคู่หรือปล่าว เราสามารถทำได้โดยดูว่า เศษเหลือจากการหาร 2 แล้วได้ 0 หรือปล่าว (เศษเหลือจากการเอาเลขคี่หาร 2 จะได้ 1 เสมอ)
- สมมตว่า วันนี้เป็นวันเสาร์แล้วเราอยากรู้ว่าอีก 10 วันเป็นวันอะไร เราสามารถหาได้โดยใช้นิพจน์ด้านล่างนี้:



Exercise: Displaying Time

เขียนนิพจน์หา ชั่วโมง กับ นาที จากจำนวนวินาที

Literals ตัวเลข (Number Literals)

literal เป็นค่าคงที่ที่ปรากฏขึ้นในโปรแกรมโดยตรง (ตัวอักขระที่ไม่ใช่ตัวแปร ไม่ใช่คำสั่ง ไม่ใช่ operators) ดังตัวอย่างด้านล่าง

34, 1,000,000, and 5.0 เป็น literal

i, x, d เป็นชื่อตัวแปร

int, long, double เป็นชนิดของตัวแปร

= เป็น operator สำหรับการกำหนดค่า

```
int i = 34;
long x = 1000000;
double d = 5.0;
```

Literal เลขจำนวนเต็ม (Integer Literals)

- Literal เลขจำนวนเต็มจะถูกกำหนดให้กับตัวแปรที่ประกาศไว้รองรับเลขจำนวนเต็มเท่านั้น (นั่นคือ ชนิด byte, short, int, long ขึ้นอยู่กับขนาดของตัวเลข)
- จะเกิดข้อผิดพลาดเมื่อ compile ถ้าเรากำหนดตัวแปรโดยใช้ literal ที่ใหญ่กว่าหรือเล็กกว่าค่าที่ตัวแปรจะรับได้ เช่น byte b = 1000 จะทำให้เกิด compile error, เพราะ 1000 ใหญ่เกินกว่าตัวแปรชนิด byte จะเก็บมันได้
- ช่วงที่เป็นไปได้ของค่าที่จะถูกเก็บอยู่ในตัวแปรชนิด int คือ ตั้งแต่ -2^{31} (-2147483648) ถึง $2^{31}-1$ (2147483647)
- ใน Java, Java จะให้ literal ที่เป็นจำนวนเต็มเป็นชนิด int หมด
- ถ้าเราอยากจะเขียน literal เลขจำนวนเต็ม แต่จะให้ Java มองเป็นชนิด long เราต้องใส่ตัวอักษร L หรือ l ตามหลัง literal นั้น (L ดีกว่าเพราะเราจะได้ไม่สับสน l กับเลข 1)

Literal ทศนิยม (Floating-Point Literals)

- ตัวอย่างการเขียน literal ทศนิยม เช่น 10.5
- โดยปกติ, Java จะให้ literal ทศนิยม เป็นชนิด double เช่น 10.5 เป็นชนิด double ไม่ใช่ชนิด float
- แต่เราสามารถกำหนดให้ literal เป็นชนิด float ได้โดยการเติม f หรือ F ต่อท้าย, หรือให้เป็นชนิด double ได้โดยการเติม d หรือ D ต่อท้าย เช่น 100.2f หรือ 100.2F เป็นทศนิยมชนิด float, และ 100.2d หรือ 100.2D เป็นทศนิยมชนิด double

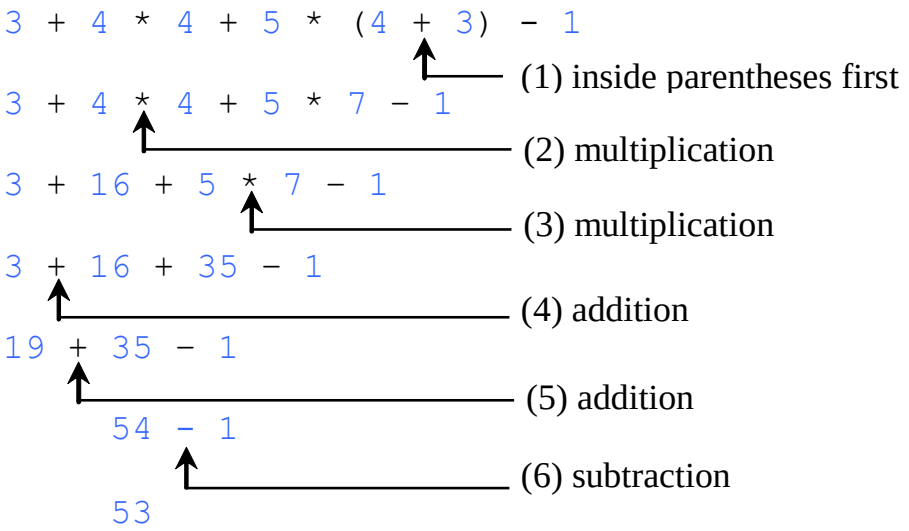
ตัวเลขทางวิทยาศาสตร์ (Scientific Notation)

Literal ทศนิยมสามารถถูกเขียนในรูปแบบตัวเลขทางวิทยาศาสตร์ได้เช่น

- 1.23456e+2 หรือ 1.23456e2 หรือ 123.456 มีค่าเท่ากัน
- 1.23456e-2 เท่ากับ 0.0123456
- E (หรือ e) แทนการยกกำลัง 10

วิธีการคำนวณค่านิพจน์ (Expression Evaluation)

Java จะมีวิธีคำนวณ expression ของตัวเอง แต่ผลลัพธ์ก็จะเหมือนกับผลลัพธ์ทาง arithmetic ดังนั้นเราใช้วิธีการคำนวณทางคณิตศาสตร์เพื่อหาค่านิพจน์ของ Java ได้



Exercise: แปลงอุณหภูมิ

จงเขียนโปรแกรมเพื่อแปลง องศา Fahrenheit ไปเป็น Celsius โดยใช้สูตรต่อไปนี้:

$$celsius = (\frac{5}{9})(\ fahrenheit - 32)$$

ตัวดำเนินการการกำหนดค่าอย่างย่อ (Shortcut Assignment Operators)

Operator	*Example*	*Equivalent*
`+=`	`i += 8`	`i = i + 8`
`-=`	`f -= 8.0`	`f = f - 8.0`
`*=`	`i *= 8`	`i = i * 8`
`/=`	`i /= 8`	`i = i / 8`
`%=`	`i %= 8`	`i = i % 8`

Increment and Decrement Operators

Operator	ชื่อ	คำอธิบาย
++var	preincrement	เพิ่มค่าตัวแปร var ไปหนึ่ง แล้วตีค่านิพจน์เท่ากับค่าใหม่ที่ได้
var++	postincrement	จะตีค่านิพจน์นี้เท่ากับค่าแรกของตัวแปร var จากนั้นจะเพิ่มค่าของตัวแปร var ขึ้นหนึ่ง
--var	predecrement	ลดค่าตัวแปร var ไปหนึ่ง แล้วตีค่านิพจน์เท่ากับค่าใหม่ที่ได้
var--	postdecrement	จะตีค่านิพจน์นี้เท่ากับค่าแรกของตัวแปร var จากนั้นจะลดค่าของตัวแปร var ลงไปหนึ่งค่า

Increment and Decrement Operators, cont.

```
int i = 10;
int newNum = 10 * i++;
```

Same effect as

```
int newNum = 10 * i;
i = i + 1;
```

```
int i = 10;
int newNum = 10 * (++i);
```

Same effect as

```
i = i + 1;
int newNum = 10 * i;
```

การแปลงชนิดของตัวเลข (Numeric Type Conversion)

พิจารณา **statement** ต่อไปนี้:-

```
byte i = 100;
long k = i * 3 + 4;
double d = i * 3.1 + k / 2;
```

กฎการแปลง (Conversion Rules)

เมื่อเราใช้ตัวดำเนินการแบบ binary (มีตัวถูกดำเนินการ 2 ตัว) กับตัวถูกดำเนินการที่มีชนิดต่างกัน, Java จะแปลงชนิดของตัวถูกดำเนินการโดยอัตโนมัติตามกฎต่อไปนี้:

1. ถ้าตัวถูกดำเนินการตัวใดตัวหนึ่งเป็นชนิด double, อีกตัวก็จะถูกแปลงเป็น double
2. ถ้าไม่ใช่ข้อข้างต้น, ถ้าตัวถูกดำเนินการตัวใดตัวหนึ่งเป็นชนิด float, อีกตัวก็จะถูกแปลงเป็น float
3. ถ้าไม่ใช่ข้อข้างต้น, ถ้าตัวถูกดำเนินการตัวใดตัวหนึ่งเป็นชนิด long, อีกตัวก็จะถูกแปลงเป็น long
4. ถ้าไม่ใช่ข้อข้างต้น, ตัวถูกดำเนินการทั้งสองจะถูกแปลงเป็น int

Type Casting

การ cast โดยปริยาย (Implicit casting)

```
double d = 3;
```
(แปลงชนิดให้ใหญ่ขึ้น จาก literal ชนิด int ไปเป็นชนิด double ก่อน แล้วค่อยไปกำหนดค่าในตัวแปร d)

การ cast โดยโปรแกรมเมอร์ตั้งใจ (Explicit casting)

```
int i = (int)3.0;
```
(แปลงชนิดให้เล็กลง จาก literal ชนิด double เป็นชนิด int)

```
int i = (int)3.9;
```
(ส่วนที่เป็นทศนิยมจะถูกตัดไป)

What is wrong? `int x = 5 / 2.0;`

range increases

byte, short, int, long, float, double

ชนิดข้อมูลตัวอักขระ (Character Data Type)

Four hexadecimal digits.

```
char letter = 'A'; (ASCII)
char numChar = '4'; (ASCII)
char letter = '\u0041'; (Unicode)
char numChar = '\u0034'; (Unicode)
```

NOTE: The increment and decrement operators can also be used on char variables to get the next or preceding Unicode character. For example, the following statements display character b.

```
char ch = 'a';
System.out.println(++ch);
```

Escape Sequences for Special Characters

Description	*Escape Sequence*	*Unicode*
Backspace	`\b`	`\u0008`
Tab	`\t`	`\u0009`
Linefeed	`\n`	`\u000A`
Carriage return	`\r`	`\u000D`
Backslash	`\\`	`\u005C`
Single Quote	`\'`	`\u0027`
Double Quote	`\"`	`\u0022`

ASCII Character Set

ASCII Character Set is a subset of the Unicode from \u0000 to \u007f

TABLE B.1 ASCII Character Set in the Decimal Index

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht
1	nl	vt	ff	cr	so	si	dle	dcl	dc2	dc3
2	dc4	nak	syn	etb	can	em	sub	esc	fs	gs
3	rs	us	sp	!	″	#	$	%	&	’
4	(	)	*	+	,	-	.	/	0	1
5	2	3	4	5	6	7	8	9	:	;
6	<	=	>	?	@	A	B	C	D	E
7	F	G	H	I	J	K	L	M	N	O
8	P	Q	R	S	T	U	V	W	X	Y
9	Z	[	\	]	^	_	‘	a	b	c
10	d	e	f	g	h	i	j	k	l	m
11	n	o	p	q	r	s	t	u	v	w
12	x	y	z	{	|	}	~	del		

ASCII Character Set, cont.

ASCII Character Set is a subset of the Unicode from \u0000 to \u007f

TABLE B.2 ASCII Character Set in the Hexadecimal Index

	0	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*A*	*B*	*C*	*D*	*E*	*F*
0	nul	soh	stx	etx	eot	enq	ack	bel	bs	ht	nl	vt	ff	cr	so	si
1	dle	dcl	dc2	dc3	dc4	nak	syn	etb	can	em	sub	esc	fs	gs	rs	us
2	sp	!	“	#	$	%	&	’	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	‘	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	del

Casting ระหว่าง ชนิด char และ ชนิดตัวเลข

```
int i = 'a'; // เหมือนกับ int i = (int)'a';

char c = 97; // เหมือนกับ char c = (char)97;
```

ชนิดสายอักขระ (String Type)

ตัวแปรชนิด char เก็บข้อมูลได้แค่หนึ่งตัวอักขระ ถ้าเราจะเก็บข้อมูลที่เป็นสายอักขระเราต้องใช้ตัวแปรชนิด String. เช่น,

String message = "Welcome to Java";

การต่อสายอักขระ (String Concatenation)

```
// นำสายอักขระ 3 สายมาต่อกัน
String message = "Welcome " + "to " + "Java";

// นำสายอักขระ Chapter มาต่อด้วยเลข 2
String s = "Chapter" + 2; // s กลายเป็น Chapter2

// สายอักขระ Supplement มาต่อด้วยอักขระ B
String s1 = "Supplement" + 'B'; // s1 กลายเป็น SupplementB
```

การรับ input จาก keyboard โดยใช้ Scanner

1. สร้าง object Scanner

```
Scanner scanner = new Scanner(System.in);
```

2. เรียกใช้ methods next(), nextByte(), nextShort(), nextInt(), nextLong(), nextFloat(), nextDouble(), or nextBoolean() เพื่อรับค่า string, byte, short, int, long, float, double, หรือ boolean เช่น,

```
System.out.print("Enter a double value: ");
Scanner scanner = new Scanner(System.in);
double d = scanner.nextDouble();
```

TestScanner

Trace ComputeChange

Suppose amount is 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount	1156

remainingAmount initialized

Trace ComputeChange

Suppose amount is 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

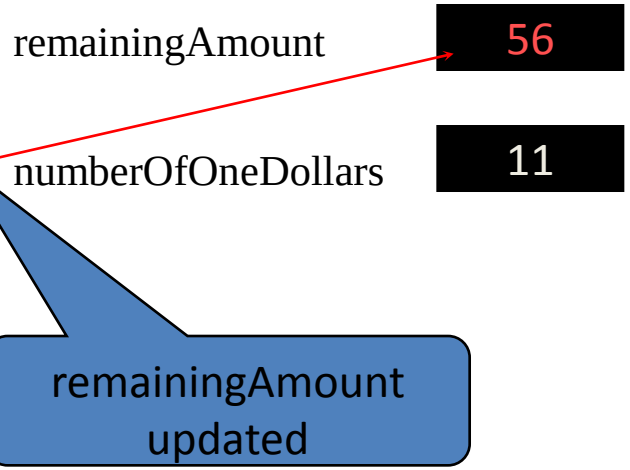
remainingAmount	1156
numberOfOneDollars	11

numberOfOneDollars assigned

Trace ComputeChange

Suppose amount is 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```



Trace ComputeChange

Suppose amount is 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount	56
numberOfOneDollars	11
numberOfOneQuarters	2

numberOfOneQuarters assigned

Trace ComputeChange

Suppose amount is 11.56

```
int remainingAmount = (int)(amount * 100);

// Find the number of one dollars
int numberOfOneDollars = remainingAmount / 100;
remainingAmount = remainingAmount % 100;

// Find the number of quarters in the remaining amount
int numberOfQuarters = remainingAmount / 25;
remainingAmount = remainingAmount % 25;

// Find the number of dimes in the remaining amount
int numberOfDimes = remainingAmount / 10;
remainingAmount = remainingAmount % 10;

// Find the number of nickels in the remaining amount
int numberOfNickels = remainingAmount / 5;
remainingAmount = remainingAmount % 5;

// Find the number of pennies in the remaining amount
int numberOfPennies = remainingAmount;
```

remainingAmount	6
numberOfOneDollars	11
numberOfQuarters	2

remainingAmount updated

การแปลง String ให้เป็น Integer

เพื่อที่จะแปลงค่าชนิด String ให้เป็นค่าชนิด int, เราสามารถใช้ method parseInt ของ class Integer ได้ดังนี้:

int intValue = Integer.parseInt(“123”);

การแปลง String ให้เป็น Double

เพื่อที่จะแปลงค่าชนิด string ให้เป็นค่าชนิด double, เราสามารถใช้ method parseDouble ของ class Double ได้ดังนี้:

double doubleValue =Double.parseDouble(“123.45”);