DASAR PEMROGRAMAN JAVA

Tipe Data

- Tipe data merupakan sekumpulan nilai dan operasi-operasi yang diasosiasikan pada nilai-nilai tersebut
- Bahasa pemrograman java mengenal dua tipe data yakni :
  - Tipe data primitif
  - Tipe data reference

Tipe Data Primitif

- Merupakan tipe data dasar yang sudah disediakan tanpa harus didefinisikan dahulu sebelum digunakan
- Java mendefinisikan 8 tipe data primitif :

1. boolean
2. char
3. byte
4. short
5. int
6. long
7. double
8. float

- **Boolean**
  - Tipe data boolean diwakili oleh dua pernyataan yakni : true dan false.

  Contoh :

```
boolean hasil = true;
```

  Contoh yang ditunjukkan diatas, mendeklarasikan variabel yang diberi nama **hasil** sebagai tipe data **boolean**, dan memberinya nilai **true.**

- **Char**
  - Tipe data char (character) diwakili oleh karakter single unicode, tipe data ini harus memiliki ciri yakni berada pada tanda single quotes (‘ ‘).

‘A’ ⟶ Huruf A

‘\t’ ⟶ Tab

- **Integral – byte, short, int & long**
  - Tipe integral memiliki default tipe data yaitu int. tipe data integral memiliki range sbb :

Integer Length	*Name or Type*	*Range*
8 bits	byte	-2^7 to 2^7-1
16 bits	short	-2^{15} to $2^{15}-1$
32 bits	int	-2^{31} to $2^{31}-1$
64 bits	long	-2^{63} to $2^{63}-1$

- **Floating Point – float dan double**
  - Tipe floating point digunakan untuk menyimpan nilai yang bersifat pecahan.
  - Memiliki double sebagai default tipe datanya.
  - Tipe floating point memiliki range nilai sbb :

Panjang Float	*Nama atau Tipe*	*Range*
32 bits	float	-2^{31} to $2^{31}-1$
64 bits	double	-2^{63} to $2^{63}-1$

Tipe data Reference

- Berbeda denga tipe data primitif, nilai pada tipe data reference merupakan reference untuk (alamat) nilai atau seperangkat nilai yg diwakili oleh variabel
- Reference merupakan pointer atau alamat memori
- Karena java tidak mendukung penggunaan pointer maka reference dilakukan menggunakan nama variabel.

Contoh tipe data reference

- Contoh tipe reference adalah :
  - Array
  - Classes
  - Interface

Variabel

- Variabel merupakan item yang digunakan data untuk menyimpan pernyataan objek
- Variabel harus memiliki nama dan tipe data untuk dapat digunakan.
- Nama variabel harus mengikuti aturan untuk identifier.
- Penulisan : <type data> <nama variabel>

Contoh deklarasi variabel

```
char huruf;
int nilai;
```

Variable Reference dan Variabel Primitif

- **Variabel primitif**
  - Merupakan variabel dengan tipe data primitif
  - Variabel tsb menyimpan data dalam lokasi sebenarnya pada memori.

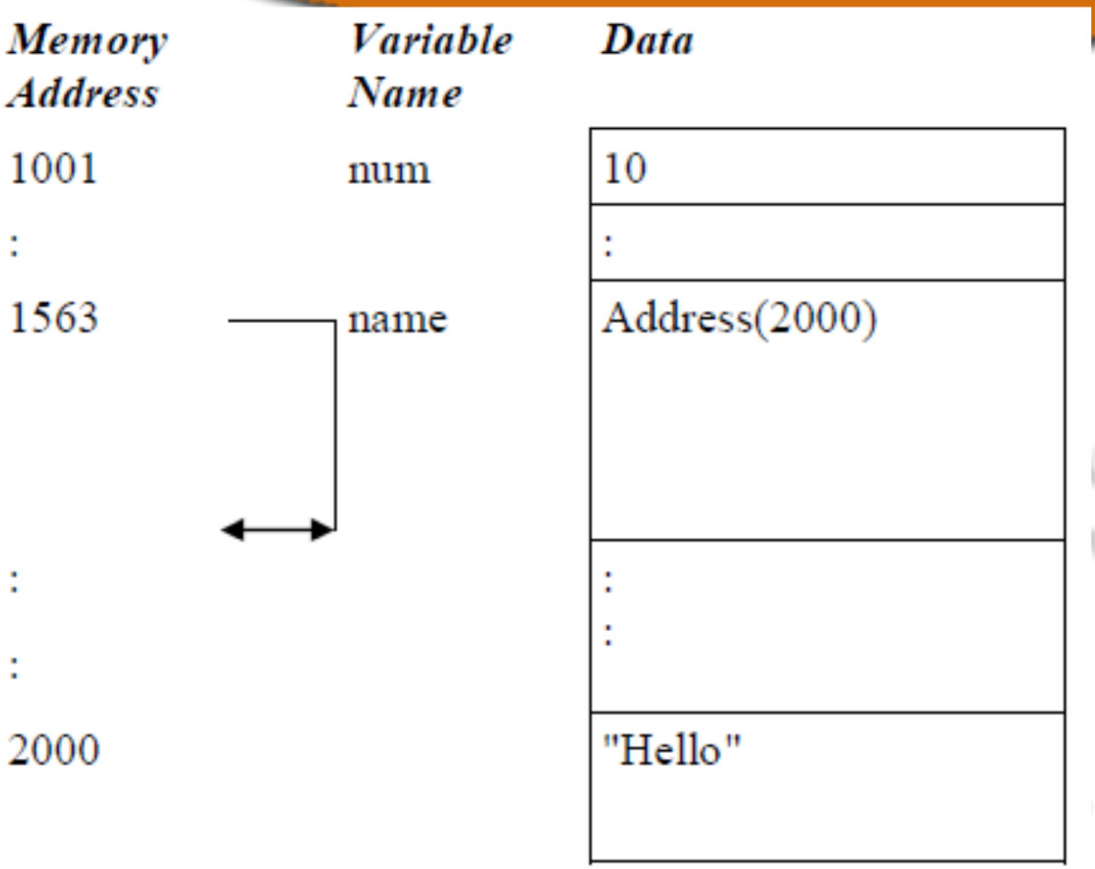
Contoh :

```
int num=20;
```

- **Variabel reference**
  - Adalah variabel yang menyimpan alamat dalam lokasi tertentu, yang menunjuk ke lokasi memori dimana data sebenarnya berada.

Contoh :

```
String name=“Hello”;
```


Memory Address
Variable Name
Data
1001
num
10
:
:
1563
name
Address(2000)
:
:
:
2000
"Hello"

What???

Ada Pertanyaan ?

Why ???

Operator

- Merupakan simbol yang memiliki fungsi untuk mengubah operand sehingga menjadi transformasi.
- Java memiliki beberapa operator dintaranya :
  - Operator Penugasan
  - Operator Aritmatika
  - Operator Relasi
  - Operator Logika
  - Operator Unary
  - Operator Shorthand
  - Operator Bitwise
  - Operator Kondisi

Operator Penugasan

- Merupakan operator yang paling sederhana dan hanya dilambangkan karakter “=“.
- Operator ini berfungsi untuk menugaskan suatu nilai ke suatu variabel

Contoh :

```
int angka=23;
```

Operator Aritmatika

- Adalah Operator yang digunakan untuk melakukan fungsi aritmetika

Operator	*Penggunaan*	*Keterangan*
+	op1 + op2	Menambahkan op1 dengan op2
*	op1 * op2	Mengalikan op1 dengan op2
/	op1 / op2	Membagi op1 dengan op2
%	op1 % op2	Menghitung sisa dari pembagian op1 dengan op2
-	op1 - op2	Mengurangkan op2 dari op1

Praktikum 2.1

- Buatlah kelas aritmatika dan tulis kode berikut:

```
12  import java.util.Scanner;
13  public class aritmatika {
14
15      double bil1=0;
16      double bil2=0;
17
18      void show()
19      {
20          Scanner input = new Scanner(System.in);
21          System.out.print("Input Bilangan 1   : ");
22          bil1=input.nextDouble();
23          System.out.print("Input Bilangan 2   : ");
24          bil2=input.nextDouble();
25
```

```
26          System.out.println("Hasil Penjumlahan : "+(bil1+bil2));
27          System.out.println("Hasil Pengurangan : "+(bil1-bil2));
28          System.out.println("Hasil Perkalian   : "+(bil1*bil2));
29      }
30
31
32      public static void main(String[] args)
33      {
34          aritmatika obj1 = new aritmatika();
35          obj1.show();
36      }
37
38  }
```

Operator Relasi

- Digunakan untuk membandingkan dua nilai dan menentukan keterhubungan antar nilai-nilai tersebut.
- Hasil keluaran dari operator ini adalah nilai true atau false

Operator	*Penggunaan*	*Keterangan*
>	op1 > op2	op1 lebih besar dari op2
>=	op1 >= op2	op1 lebih besar dari atau sama dengan op2
<	op1 < op2	op1 kurang dari op2
<=	op1 <= op2	op1 kurang dari atau sama dengan op2
==	op1 == op2	op1 sama dengan op2
!=	op1 != op2	op1 tidak sama dengan op2

Praktikum 2.2

- Buatlah kelas op_relasi dan tulis kode berikut:

```
 */
import java.util.Scanner;
public class op_relasi {
    double bil1=0;
    double bil2=0;

    void show()
    {
        Scanner input = new Scanner(System.in);
        System.out.print("Input Angka 1       : ");
        bil1=input.nextDouble();
        System.out.print("Input Angka 2       : ");
        bil2=input.nextDouble();

        System.out.println("Angka1 > Angka2      : "+(bil1>bil2));
        System.out.println("Angka1 < Angka2      : "+(bil1<bil2));
```

```
27            System.out.println("Angka1 >= Angka2     : "+(bil1>=bil2));
28            System.out.println("Angka1 <= Angka2     : "+(bil1<=bil2));
29            System.out.println("Angka1 == Angka2     : "+(bil1==bil2));
30            System.out.println("Angka1 != Angka2     : "+(bil1!=bil2));
31        }
32
33        public static void main(String[] args)
34        {
35            op_relasi relasi = new op_relasi();
36            relasi.show();
37        }
38    }
```

Operator Logika

- Operator logika memiliki satu atau lebih operand yang bernilai boolean dan menghasilkan nilai boolean
- Terdapat enam operator logika antaralain :
  - Logika AND (&&)
  - Boolean logika AND (&)
  - Logika OR (||)
  - Boolean inclusive OR (|)
  - Boolean exclusive OR (^)
  - Logika NOT (!)

- **Logika AND (&&) dan boolean logika AND (&).**
  - Perbedaan antara "&&" dan "&" adalah operator "&&" mensupport short-circuit evaluation sementara operator "&" tidak.

x1	*x2*	*Hasil*
TRUE	TRUE	TRUE
TRUE	FALSE	FALSE
FALSE	TRUE	FALSE
FALSE	FALSE	FALSE

- **Logika OR (||) dan boolean logika OR (|).**
  - Sama seperti logika AND, logika OR / "||" mendukung short-circuit evaluations, sementara "|" tidak.

x1	*x2*	*Hasil*
TRUE	TRUE	TRUE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

- **boolean logika ExclusiveOR (^)**
  - Tabel kebenaran untuk logika exclusiveOR sbb :

x1	*x2*	*Hasil*
TRUE	TRUE	FALSE
TRUE	FALSE	TRUE
FALSE	TRUE	TRUE
FALSE	FALSE	FALSE

- **Logika NOT (!)**
  - Logika not digunakan dalam satu argumen dimana argumen tersebut dapat menjadi suatu pernyataan, variabel atau konstanta. Berikut ini adalah tabel kebenaran untuk operator NOT.

x1	*Hasil*
TRUE	FALSE
FALSE	TRUE

Praktikum 2.3

- Dengan menggunakan operator logika, buatlah kelas op_logical dan tulis kode berikut:

```
12  import java.util.Scanner;
13  public class op_logical {
14
15      int bil1=0;
16      int bil2=0;
17
18      void show()
19      {
20          Scanner input = new Scanner(System.in);
21          System.out.print("Input Angka 1       : ");
22          bil1=input.nextInt();
23          System.out.print("Input Angka 2       : ");
24          bil2=input.nextInt();
25
```

```
26              System.out.println("Angka1 & Angka2      : "+(bil1&bil2));
27              System.out.println("Angka1 | Angka2      : "+(bil1|bil2));
28              System.out.println("Angka1 ^ Angka2      : "+(bil1^bil2));
29          }
30
31          public static void main(String[] args)
32          {
33                  op_logical logical = new op_logical();
34                  logical.show();
35          }
36      }
```

Operator Unary

- Operator unary dapat dikenakan hanya pada satu operand

Arti Operator	Operator	Contoh Pemakaian	Keterangan
Pre-Increment	++operand	int i = 8 ; int j = ++i; i bernilai 8, j bernilai 8	

Post-Increment	operand++	int i = 8;
		int j = i++;
		i bernilai 9, j bernilai 8	

Pre-Decrement	--operand	int i = 8 ; int j = --i; i bernilai 7 , j bernilai 7	
Post-Increment	operand--	int i=8; int j = i--; i bernilai 7, j bernilai 8	

Operator Shorthand

- Operator ini digunakan untuk menyingkat penulisan

Operator	Nama Operator	Penggunaan	Ekuivalen
+=	Addition assignment	i+=2	i=i+2
-=	Subtraction assignment	i-=2	i=i-2
*=	Multiplication assignment	i*=2	i=i*2
/=	Division assignment	i/=2	i=i/2
%=	Remainder assignment	i%=2	i=i%2

Operator Bitwise

- Digunakan untuk melakukan operasi pada tingkat digital

Arti Operator	Operator	Contoh Pemakaian	Keterangan

Shift Kiri	<<	int b = -16;
		int c = b<<2;
		nilai c = -64	

Shift Kanan	>>	int b = -16;
		int c = b>>2;
		nilai c = -4
			Ketika digeser ke kanan 1 kali, maka bit paling kiri terisi dengan angka yang sama dengan *sign* bit sebelumnya.

Shift Kanan, dengan pengisian "0" pada bit-bit sebelah kiri	>>>	int b = -16;
		int c = -16>>>2;
		nilai c = 1073741820
			Ketika digeser ke kanan 1 kali, maka bit paling kiri terisi dengan 0

AND	&	int a = 12;
		int b = - 13;
		int c = a & b;
		nilai c = 0
			Yang di-AND adalah setiap bit dari a dan b yang menempati posisi bit yang sama, misalnya bit ke-2 variabel a di-AND dengan bit ke-2 variabel b.

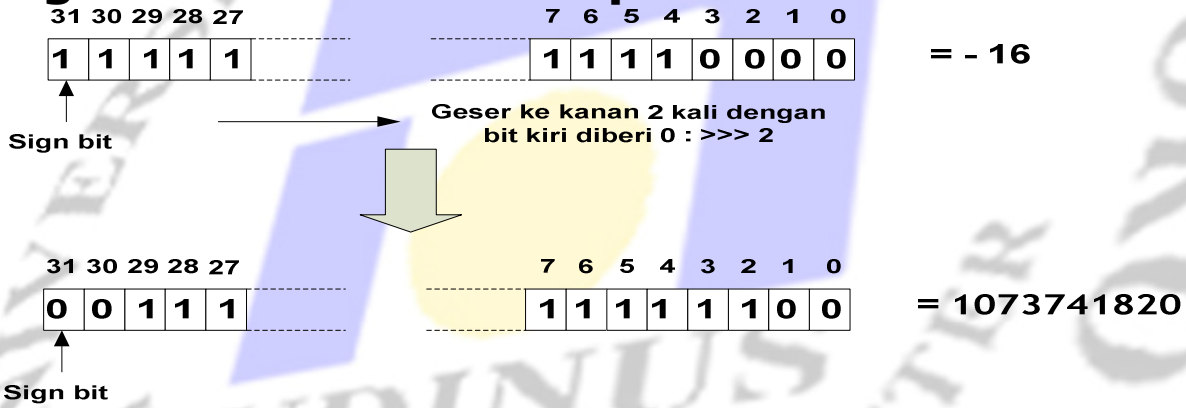
OR	|	int a = 12;
		int b = -13;
		int c = a | b;
		nilai c = -1
			Yang di-OR adalah setiap bit dari a dan b yang menempati posisi bit yang sama

Exclusive OR	^	int a = 13;
		int b = -13;
		int c = a ^ b ;
		nilai c = -2
			Yang di-exclusive OR adalah setiap bit dari a dan b yang menempati posisi bit yang sama

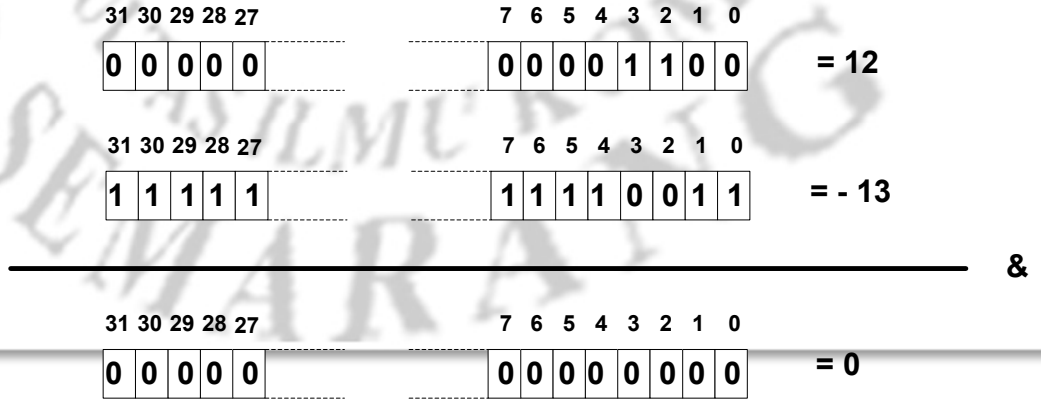
Complement	~	int a = 12;
		int c = ~a;
		nilai c = -13
			Nilai setiap bit diganti dengan lawannya. Jika bit bernilai 1, maka nilai tersebut akan dirubah menjadi 0

Operator Bitwise (1)

- **Shift Kanan Dengan Penambahan '0' pada Bit-bit Kiri**

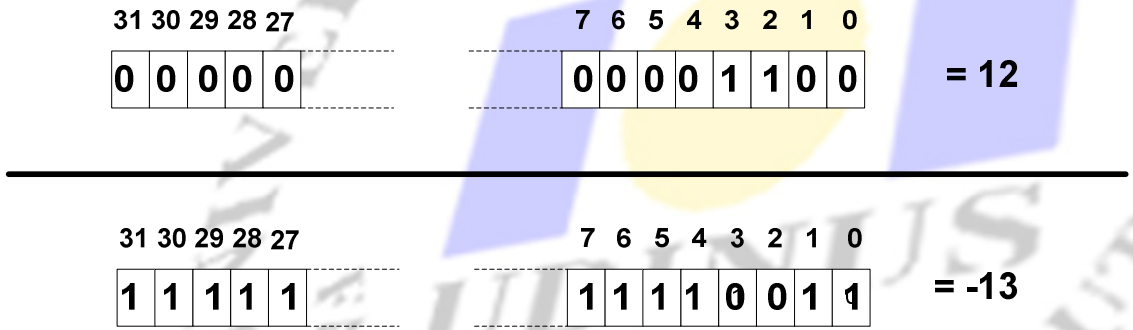


- **Operator '&'**



Operator Bitwise (2)

- **Operator '~'**



Operator Kondisi

- Operator kondisi “?”,”:”. Merupakan operator ternary, artinya operator ini membawa tiga argumen yang membentuk suatu ekspresi bersyarat.

Contoh :

```
exp1?exp2:exp3
```

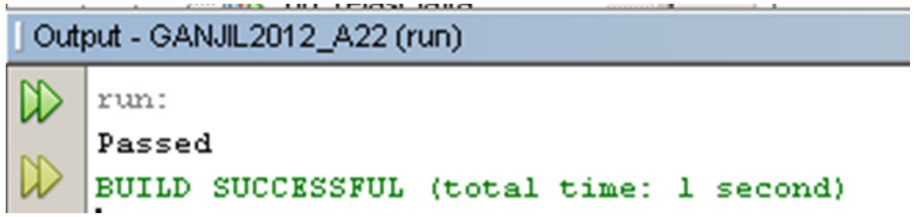
- Dimana exp1 adalah pernyataan boolean yang memiliki hasil true / false
- Jika exp1 bernilai true maka, exp2 adalah hasil operasi, jika exp1 bernilai false maka exp3 adalah hasil operasi

Praktikum 2.4

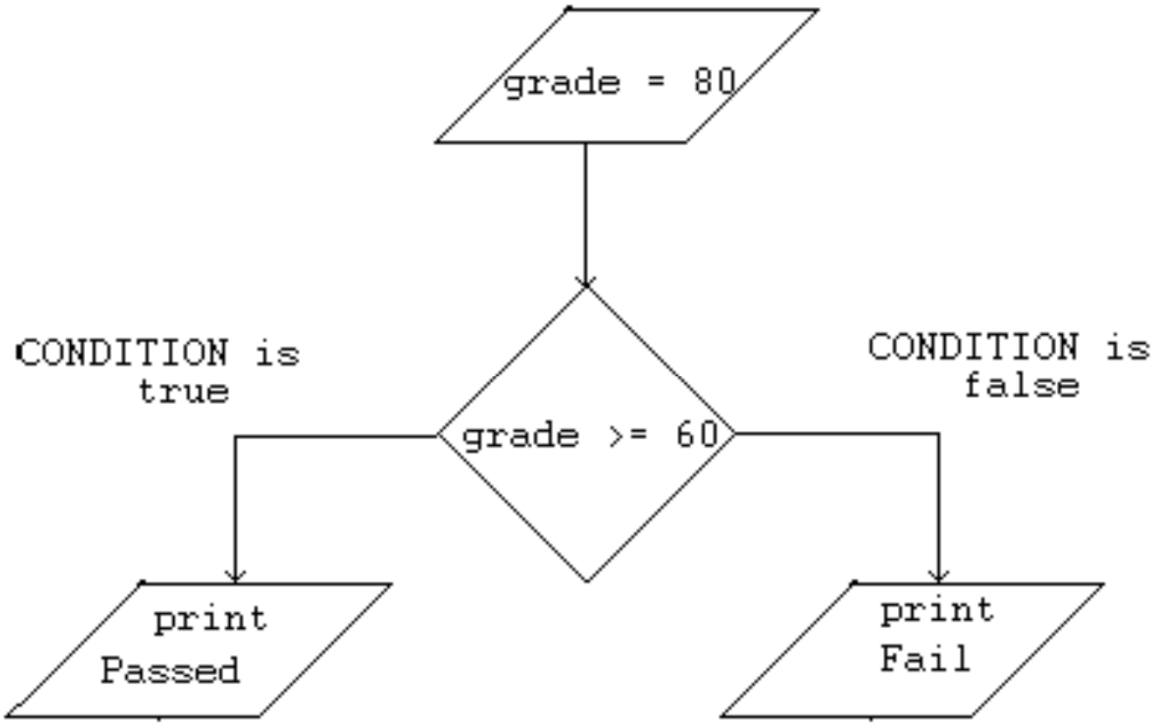
- Dengan menggunakan operator kondisi, buatlah kelas op_conditional dan tulis kode berikut:

```
12  public class op_conditional {
13
14      void show()
15      {
16          String status;
17          int grades=80;
18
19          status = (grades>=60)?"Passed":"Fail";
20          System.out.println(status);
21      }
22      public static void main(String[] args)
23      {
24          op_conditional kondisi = new op_conditional();
25          kondisi.show();
26      }
27  }
```

Hasil keluaran dari program ini akan menjadi,



Flowchart

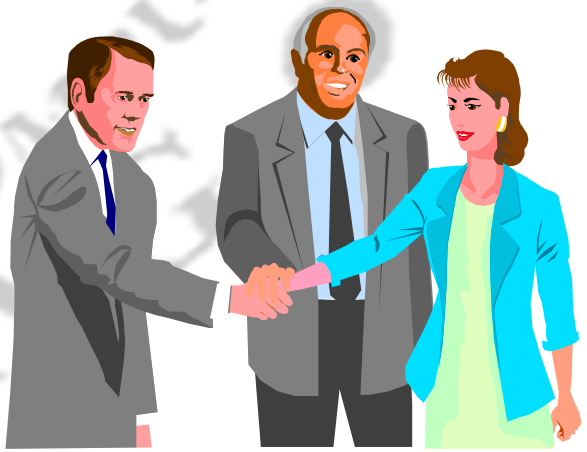

grade = 80
grade >= 60
CONDITION is true
CONDITION is false
print Passed
print Fail

What???

Ada Pertanyaan ?

Why ???

Terima kasih



Daftar Pustaka

- Java™ Tutorial, Third Edition: A Short Course on the Basics, Addison Wesley , 2000.
- Kathy Sierra & Bert Bates, "Sun Certified Programer for Java tm 6 Study Guide", McGraw-Hill Companies, 2008.