Tugas 1 PBO Makassar, 28 Februari 2025

**PRAKTIKUM**

**Pemograman Berorientasi Objek**

Nama : Wahyuni

Stambuk : 13020230110

Frekuensi : TI\_PBO-12 (B2)

Dosen : Mardiyyah Hasnawi, S.Kom.,M.T.

**PROGRAM STUDI TEKNIK INFORMATIKA**

**FAKULTAS ILMU KOMPUTER**

**UNIVERSITAS MUSLIM INDONESIA**

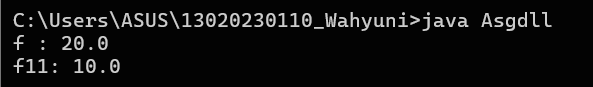
**MAKASSAR**

**2024**

**Tugas1\_13020230110\_Wahyuni**

**Output 16 Kode dengan penjelasannya**

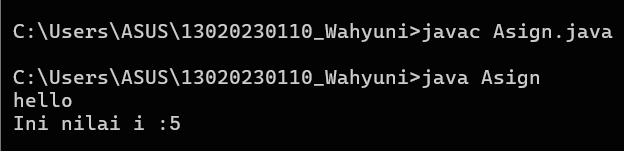




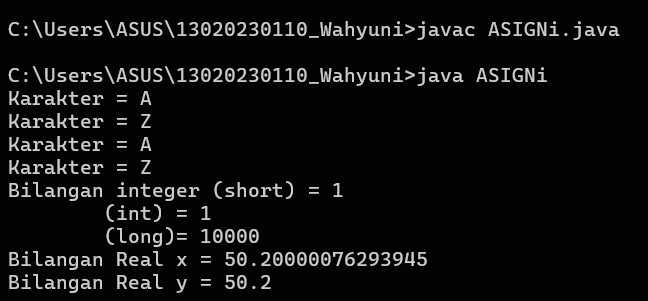
**Penjelasan soal**

Dua variabel dideklarasikan. Variabel pertama yaitu f, rtipe float dan diinisialisasi dengan nilai 20.0f. Ini berarti bahwa variabel f menyimpan angka desimal 20.0. Nah variabel kedua adalah fll, yang bertipe double, tetapi belum diinisialisasi pada saat deklarasi. Selanjutnya, dalam algoritmanya, nilai diberikan kepada variabel fll. Di sini, fll diisi dengan nilai 10.0f . Meskipun nilai ini ditulis dengan akhiran f, yang menunjukkan bahwa itu adalah tipe float, Java secara otomatis mengonversi nilai tersebut menjadi tipe double saat disimpan dalam variabel fll. Ini adalah salah satu fitur dari Java yang memungkinkan konversi tipe data secara otomatis. Setelah itu, program mencetak nilai dari kedua variabel ke konsol menggunakan perintah system.out.orintln Dalam perintah ini, teks yang dicetak adalah "f : " diikuti oleh nilai dari variabel f, dan kemudian diakhiri dengan baris baru sebelum mencetak "fll: " diikuti oleh nilai dari variabel fll. Tanda /n digunakan untuk membuat baris baru dalam output. Output ini menunjukkan bahwa variabel f memiliki nilai 20.0 dan variabel fll memiliki nilai 10.0. Meskipun dalam kode fll diisi dengan 10.0f, Java berhasil mengonversi nilai tersebut menjadi tipem double, sehingga outputnya tetap konsisten. Secara keseluruhan, kode ini memberikan contoh yang baik tentang cara mendeklarasikan dan menggunakan variabel dengan tipe data yang berbeda dalam Java, serta cara mencetak hasilnya ke konsol. Output yang dihasilkan memberikan informasi yang jelas tentang nilai yang disimpan dalam masing-masing variabel.



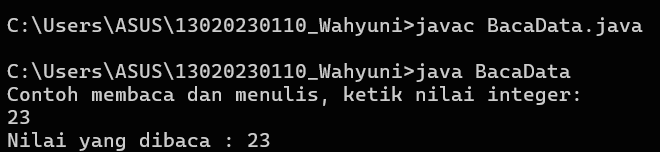


Program di atas adalah contoh sederhana dari penggunaan bahasa pemrograman Java. Pada bagian main  kode pertama kali mengeluarkan teks "hello" di layar, diikuti dengan nilai dari variabel i variabel i ditetapkan dengan nilai 5 sebelum ditampilkan. Proses yang terjadi dimulai dengan perintah javac asign.java yang digunakanuntuk mengomplikasikan kode sumber java. Setelah itu programa di eksekusi dengan perintah java asign. Hasil dari eksekusi program tersebut adalah tampilan di konsol yang menunjukkan dua baris output. Baris pertama menampilkan "hello", diikuti dengan baris kedua yang menunjukkan "Ini nilai i : 5". Ini menunjukkan bahwa program berhasil mengeksekusi semua instruksi yang ditulis dengan benar, tanpa kesalahan. Untuk outputnya program mencetak argumen yang diterima hello dan mengeluarkan pesan yang menunjukkan nilai variabel i yang bernilai 5

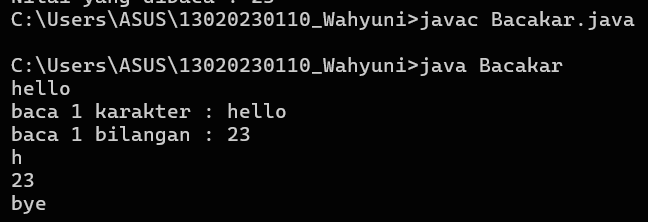
1. 

Gambar tersebut menunjukkan output dari program java yang dijalankan di CLI pada sistem operasi windows. Dalam tampilan ini, terlihat langkah langkah dalam pembuatan serta eksekusi program java bernama ASIGNi. Proses dimulai dengan perintah untuk mengimplikasi file java denganperintah javac ASUGNi.java diikuti dengan perintah untuk menjalankan program dengan java ASIGNi

Dari output yang dihasilkan, program mencetak beberapa nilai yang terkait dengan variabel yang dideklarasikan dalam program. Variabel pertama adalah karakter, yang menunjukkan nilai 'A' dan 'Z' sebanyak dua kali. Selanjutnya, terdapat informasi tentang beberapa bilangan: satu integer dengan nilai 1, dan dua tipe data long dan integer yang diharapkan menunjukkan nilai yang lebih besar. Terakhir, dua variabel real dideklarasikan, yaitu x, dengan nilai lebih presisi (50.20000076293945), dan y, yang dinyatakan sebagai 50.2. Output ini memberikan gambaran tentang bagaimana variabel bekerja dalam Java, serta perbedaan antara tipe data integer dan real.

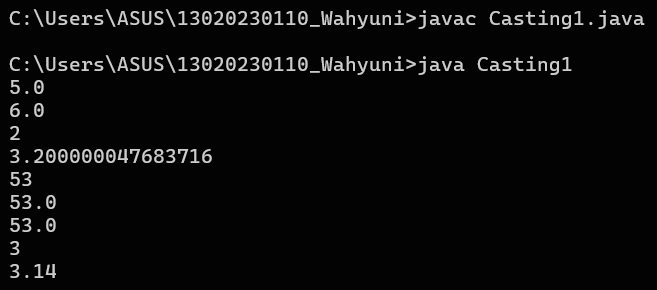


Dalam cuplikan gambar di atas, terlihat sebuah proses menjalankan program Java yang bernama "BacaData.java". Pada langkah pertama, perintah javac DataBaca.java digunakan untuk mengompilasi kode sumber program tersebut. Setelah proses kompilasi yang sukses, perintah java DataBaca dijalankan untuk mengeksekusi program. Program ini meminta input dari pengguna dengan menampilkan pesan "Contoh membaca dan menulis, ketik nilai integer:". Pengguna kemudian memasukkan angka 23 sebagai input. Setelah itu, program membaca nilai yang dimasukkan dan menampilkannya dengan mencetak "Nilai yang dibaca: 23". Proses ini menunjukkan bagaimana program Java dapat membaca input dari pengguna dan mengolahnya untuk ditampilkan kembali.

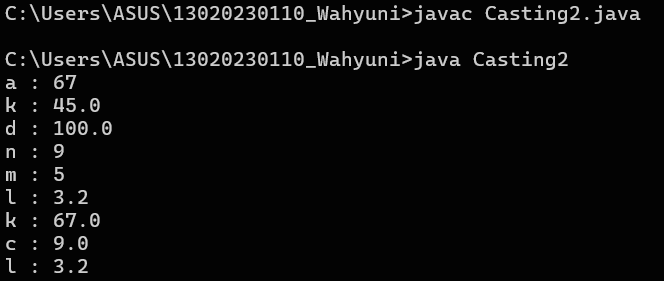


Program Java yang diberi nama Bacakar adalah contoh sederhana yang menunjukkan cara membaca input dari pengguna melalui konsol. Di dalam metode main, program dimulai dengan mendeklarasikan dua variabel: cc untuk menyimpan karakter yang akan dibaca dan bil untuk menyimpan bilangan bulat. Program kemudian menggunakan BuffereReader untuk membaca input dari pengguna. Setelah mencetak pesan "hello", program meminta pengguna untuk memasukkan satu karakter dengan menampilkan prompt "baca 1 karakter : ". Karakter pertama dari input yang diberikan oleh pengguna disimpan dalam variabel cc. Selanjutnya, program meminta pengguna untuk memasukkan sebuah bilangan bulat dengan prompt "baca 1 bilangan : ". Input ini dibaca sebagai string dan kemudian diubah menjadi integer menggunakan Integer.parseInt(), sebelum disimpan dalam variabel bil. Setelah kedua input diterima, program mencetak karakter dan bilangan bulat yang telah dimasukkan, diikuti dengan pesan "bye". Sebagai contoh, jika pengguna memasukkan karakter 'a' dan bilangan 10, output yang dihasilkan akan menampilkan "hello", diikuti oleh 'a', 10, dan diakhiri dengan "bye". Program ini memberikan gambaran dasar tentang bagaimana cara berinteraksi dengan pengguna melalui input dan output di konsol dalam bahasa pemrograman Java.





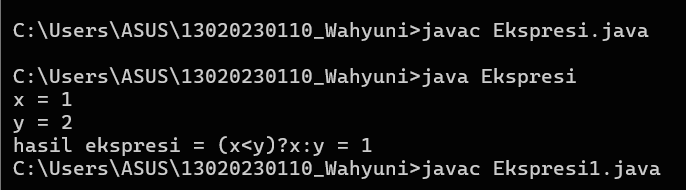
Program Java yang berjudul Casting1 mendemonstrasikan penggunaan casting antara berbagai tipe data primitif. Di dalam metode main, beberapa variabel dideklarasikan, termasuk dua integer (a dan b), dua float (g dan e), satu karakter g, dan satu double (k). Program kemudian mencetak hasil dari berbagai operasi casting yang dilakukan pada variabel-variabel tersebut. Misalnya, ketika integer a dikonversi menjadi float, hasilnya adalah 5.0, sedangkan konversi dari float d menjadi integer menghasilkan 2, di mana nilai desimal dibuang. Program juga menunjukkan bagaimana karakter g, yang berisi '5', dapat dikonversi menjadi nilai ASCII-nya, yaitu 53, baik dalam bentuk integer, float, maupun double. Selain itu, konversi dari double k menjadi integer menghasilkan 3, dengan nilai desimal yang dibuang, sedangkan konversi ke float tetap mempertahankan nilai desimal, menghasilkan 3.14. Dengan demikian, program ini memberikan gambaran yang jelas tentang bagaimana casting bekerja dalam Java dan bagaimana nilai-nilai dapat berubah ketika dikonversi dari satu tipe data ke tipe data lainnya, tanpa memerlukan input dari pengguna.



Program Java yang berjudul Casting 2 menunjukkan penggunaan casting dan konversi antara berbagai tipe data menggunakan kelas wrapper dan metode konversi. Di dalam metode main, beberapa variabel dideklarasikan, termasuk integer (a dan b), float (d dan e), karakter (g), double (k), dan string (n, m, dan l). Program pertama-tama mengonversi string n, m, dan i menjadi tipe data numerik yang sesuai menggunakan metode Integer.PrseInt() Double.parseDouble() dan Float.parseFloat() Hasil konversi ini kemudian dicetak, menunjukkan nilai integer a, double k, dan float d. Selanjutnya, program melakukan konversi dari tipe data numerik kembali ke string menggunakan String.valueOf() yang mengubah integer b, karakter g, dan float e menjadi string, dan hasilnya juga dicetak. Program kemudian mengonversi integer a menjadi double menggunakan Double.valueOf()  dan mencetak hasilnya. Selain itu, integer b juga dikonversi menjadi double dengan cara yang sama, dan hasilnya disimpan dalam variabel c. Program ini memberikan gambaran yang jelas tentang bagaimana konversi antara tipe data dapat dilakukan menggunakan metode yang disediakan oleh kelas wrapper di Java, serta bagaimana nilai-nilai dapat diubah dari satu tipe ke tipe lainnya dengan mudah untuk output

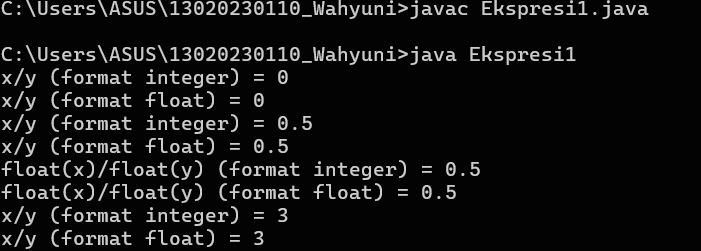
Gambar di atas menunjukkan hasil eksekusi program Java yang bernama Casting2.java. Dalam tampilan tersebut, terdapat proses kompilasi dan eksekusi yang menghasilkan beberapa variabel dengan nilai yang dicetak di konsol. Variabel-variabel yang ditampilkan antara lain a,k,d,n,m,l,dan c dimana variabel a memiliki nilai 67, sedangkan k bernilai 45.0. Terdapat juga variabel d yang menampung nilai 100.0, serta n yang berisi angka 9. Demikian pula, variabel m  memiliki nilai 5, l  sejumlah 3.2, dan c bernilai 69.0. Variabel-variabel ini menunjukkan berbagai tipe data, yaitu integer dan floating-point, yang diolah dalam program menggunakan casting, sehingga kita dapat melihat berbagai representasi nilai yang berbeda. Hasil ini menunjukkan bagaimana Java menangani tipe data dan melakukan konversi di antara mereka sesuai dengan kebutuhan dalam program.





Program Java yang dituliskan dalam kelas Ekspresi menggunakan operator kondisional untuk menentukan nilai yang akan dicetak berdasarkan perbandingan dua variabel,x dan y. Dalam bagian algoritma , variabel x diinisialisasi dengan nilai 1, dan y dengan nilai 2. Ketika program dijalankan, output pertama yang dihasilkan adalah nilai dari x dan y yang masing-masing dicetak dengan pesan x=1 dan y = 2.  Selanjutnya, program memeriksa kondisi apakah x lebih kecil dari y menggunakan ekspresi kondisional (x<y)?x:y. Karena kondisi tersebut bernilai benar (true), maka hasil dari ekspresi tersebut adalah x dan outputnya adalah Hasil ekspresi = (x<y)?x:y=1

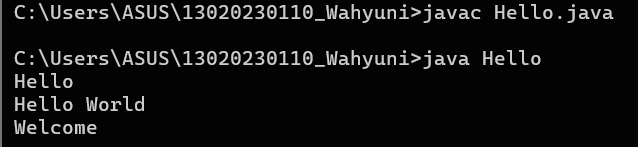




Program Java yang diberikan melakukan pembagian antara dua angka integer dan float, serta menunjukkan perbedaan hasilnya berdasarkan tipe data yang digunakan. Pada awalnya, variabel x dan y diisi dengan nilai 1 dan 2, dan hasil pembagian x/y ditampilkan dalam format integer dan float. Namun, hasil pembagian dalam format integer akan menjadi 0 karena pembagian antara dua integer dengan hasil pecahan akan dibulatkan ke bawah, sedangkan untuk format float, hasilnya akan tetap 0.5 karena salah satu operand (variabel fy dan fy) di-casting menjadi float. Selanjutnya, nilai x dan y dirubah menjadi 10 dan 3, masing-masing, dan hasil pembagian kembali ditampilkan.

Output dari program ini menunjukkan bahwa pemrograman Java membedakan hasil berdasarkan tipe data, di mana pembagian antara dua integer hanya memperlihatkan bagian utuhnya, sedangkan pembagian dengan casting menjadi float memberikan hasil yang lebih akurat. Hasil output mungkin terlihat sebagai berikut: "x/y (format integer) = 0, x/y (format float) = 0.5, float(x)/float(y) (format integer) = 0, float(x)/float(y) (format float) = 0.5, x/y (format integer) = 3, x/y (format float) = 3.333...".

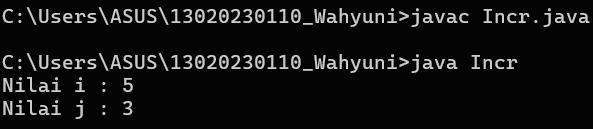




Program Java yang diberikan adalah contoh sederhana yang berfungsi untuk mencetak beberapa teks ke layar. Pada metode main  terdapat beberapa instruksi untuk mencetak string menggunakan System.out.print dan System.out.prinln.

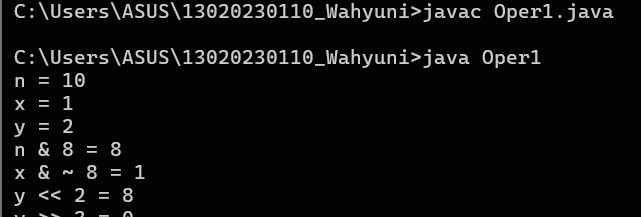
Pertama, program akan mencetak "Hello" tanpa berpindah ke baris baru, diikuti dengan \n yang berfungsi menjadikan baris berikutnya dimulai dari baris baru dan mencetak "Hello " lagi. Selanjutnya, System.out.prinln (World) akan mencetak "World" di baris baru, dan terakhir , System.out.prinln(welcome) mencetak "Welcome" juga di baris baru. Akhir dari eksekusi metode  main ini menghasilkan keluaran yang menampilkan gambar seperti diatas





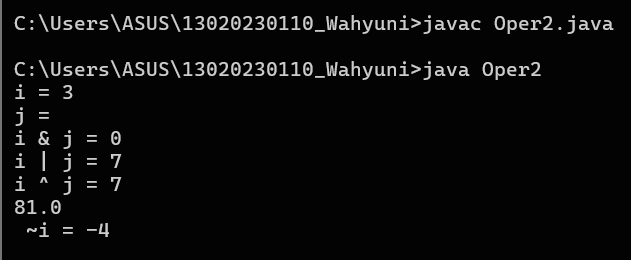
Di dalam kode, kita memiliki dua variabel bertipe integer, yaitu i dan j . Pada awalnya, i diatur ke nilai 3, dan kemudian j diatur dengan nilai 1++. Ini berarti bahwa nilai j akan menjadi nilai awal i sebelum ditingkatkan, yaitu 3. Setelah pernyataan ini, nilai i akan bertambah 1, sehingga menjadi 4. Untuk output nya Saat kita mencetak nilai i dan j, kita menggunakan ekspresi i++ . Ini menyebabkan nilai i  yang sebelumnya 4, ditingkatkan lagi menjadi 5 sebelum ditampilkan. Dengan demikian output akan mencetak nilai i= 5 dan nilai j tetap 3 sesuai dengan nilai sebelum peningkatan.





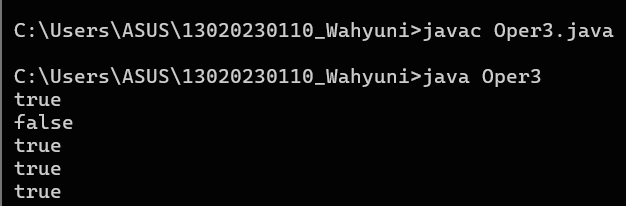
Dari tampilan yang diberikan, kita melihat output hasil kompilasi dan eksekusi sebuah program Java yang membahas operasi bitwise. Di sini, variabel n di-set dengan nilai 10, x dengan nilai 1, dan y dengan nilai 2. Operasi n & 8enghitung hasil AND bitwise antara 10 (nilai dari n) dan 8, menghasilkan 8. Selain itu, dengan operasi x & 8, kita mendapatkan hasil 1, di mana 8 adalah komplement bitwise dari 8. Terakhir, pergeseran bit kiri x>> 2menghasilkan 8, yang menunjukkan bahwa variabel x digeser dua tempat ke kiri. Secara keseluruhan, program ini menunjukkan bagaimana operasi bitwise dapat digunakan untuk manipulasi angka dalam bahasa pemrograman Java.





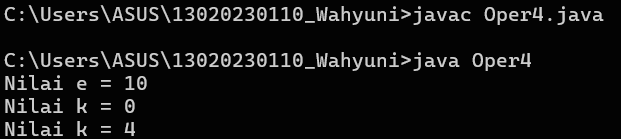
Program Java di atas menunjukkan penggunaan berbagai operator bitwise dan relasional pada tipe data karakter( char ). Dalam kode ini, dua variabel karakter, i dan j , diinisialisasi dengan nilai 3 dan 4, yang dalam representasi biner adalah 00000011 dan 00000100 Program mencetak nilai dari i dan j , di mana i dikonversi menjadi integer untuk menampilkan nilainya sebagai 3, sedangkan j tetap ditampilkan sebagai 4. Selanjutnya, program melakukan beberapa operasi bitwise: operasi AND (&) antara i dan j menghasilkan 0, karena tidak ada bit yang sama-sama 1; operasi OR (l) menghasilkan 7, yang merupakan hasil penggabungan semua bit yang bernilai 1; dan operasi XOR (^) juga menghasilkan 7, karena XOR menghasilkan 1 hanya jika bit yang dibandingkan berbeda. Penting untuk dicatat bahwa operator ^ di Java tidak digunakan untuk pemangkatan, melainkan untuk operasi XOR. Untuk menghitung pemangkatan, program menggunakan metode Math.pow(i,j), yang menghitung 3 pangkat 4 dan menghasilkan 81. Terakhir, operator bitwise NOT diterapkan pada i, yang menghasilkan -4, karena operator ini membalik semua bit dari nilai biner i . Secara keseluruhan, program ini memberikan pemahaman yang baik tentang bagaimana operator bitwise dan relasional berfungsi dalam Java, serta perbedaan antara operasi bitwise dan aritmetika.





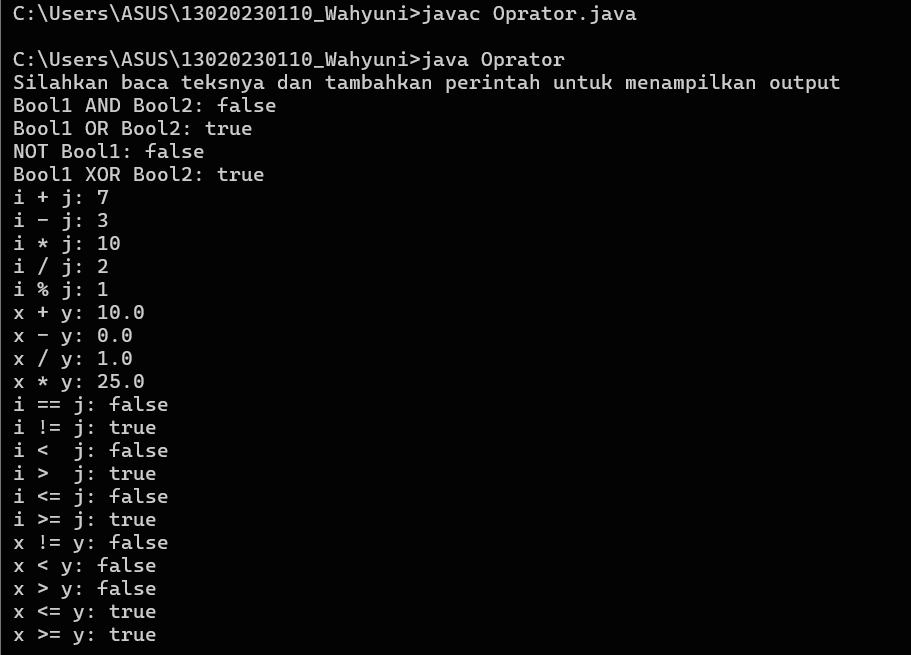
Program Java di atas mendemonstrasikan penggunaan operator logika dalam beberapa kondisi if. Dalam kode ini, terdapat beberapa pernyataan yang menggunakan operator logika AND (&& dan &) serta OR (|| dan |). Pertama, pernyataan if (true && true) dieksekusi, dan karena kedua operand bernilai true, maka kondisi ini terpenuhi dan mencetak true ke konsol. Selanjutnya, pernyataan if (true & true) juga dieksekusi, tetapi di dalam blok tersebut terdapat System.out.println(true & false), yang akan menghasilkan false karena operator AND (&) hanya menghasilkan true jika kedua operand bernilai true. Namun, karena kondisi if sebelumnya adalah true, maka blok tersebut dieksekusi dan mencetak false. Kemudian, pernyataan if (true) dieksekusi, yang langsung mencetak true ke konsol. Selanjutnya, pernyataan if (true || true) dieksekusi, yang juga mencetak true karena operator OR (||) menghasilkan true jika salah satu operand bernilai true. Terakhir, pernyataan if (true | false) dieksekusi, dan di dalam blok tersebut terdapat System.out.println(true | false), yang juga menghasilkan true karena operator OR (|) menghasilkan true jika salah satu operand bernilai true.





Program Java ini menggunakan operator terner (ternary operator) untuk menentukan nilai berdasarkan kondisi tertentu. Di dalam program, dua variabel integer i dan j diinisialisasi dengan nilai 0. Dua karakter c dan d diinisialisasi dengan nilai 8 dan 10. Kemudian, program menggunakan operator terner untuk membandingkan nilai c dan d. Karena 8 tidak lebih besar dari 10, nilai d (10) yang lebih besar akan disimpan dalam variabel e. Selanjutnya, program membandingkan i dan j, yang keduanya adalah 0, sehingga nilai j (0) disimpan dalam variabel k. Program kemudian mencetak nilai e dan k, yang masing-masing adalah 10 dan 0. Setelah itu, nilai i diubah menjadi 2 dan j menjadi 3. Program kemudian menggunakan operator terner lagi untuk membandingkan i++ dan j++. Namun, karena i++ dan j++ adalah post-increment, nilai yang dibandingkan adalah 2 dan 3. Karena 2 tidak lebih besar dari 3, nilai j (4, setelah increment) akan disimpan dalam k.





Program Java ini menunjukkan berbagai operasi yang dapat dilakukan pada variabel bertipe dasar, termasuk boolean, integer, dan float. Pertama, program mendeklarasikan beberapa variabel boolean (Bool1 dan Bool2) dan melakukan operasi logika menggunakan operator AND (&&), OR (||), NOT (!), dan XOR (^). Hasil dari setiap operasi dicetak ke layar, sehingga pengguna dapat melihat hasil dari kombinasi nilai boolean tersebut. Setelah itu, program melanjutkan dengan operasi numerik pada dua variabel integer i dan j, yang diinisialisasi dengan nilai 5 dan 2. Program melakukan operasi penjumlahan, pengurangan, perkalian, pembagian, dan modulo, mencetak hasil dari setiap operasi. Selanjutnya, program melakukan operasi yang sama pada dua variabel float x dan y, yang keduanya diinisialisasi dengan nilai 5. Hasil dari operasi penjumlahan, pengurangan, pembagian, dan perkalian juga dicetak. Terakhir, program melakukan serangkaian operasi relasional pada variabel i dan j, serta x dan y, untuk membandingkan nilai-nilai tersebut. Hasil dari setiap perbandingan (seperti sama dengan, tidak sama dengan, lebih kecil dari, lebih besar dari, dan seterusnya) dicetak ke layar. Output dari program ini akan mencakup hasil dari semua operasi yang dilakukan, yang memberikan gambaran lengkap tentang bagaimana operator logika, aritmetika, dan relasional bekerja dalam Java.