|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Iqbal Tri Agustian**  **G1F024043** | **Konversi Tipe Data** | **30 Agustus 2024** |
| **[Nomor Soal 1] Identifikasi Masalah:** | | |
| public class TipeData{   public static void main(String[] args) {     System.**out**.println("byte terkecil= " + Byte.**MIN\_VALUE**);     System.**out**.println("byte terbesar= " + Byte.**MAX\_VALUE**);     System.**out**.println("\n short terkecil= " + Short.**MIN\_VALUE**);     System.**out**.println("short terbesar= " + Short.**MAX\_VALUE**);     System.**out**.println("\n int terkecil= " + Integer.**MIN\_VALUE**);     System.**out**.println("int terbesar= " + Integer.**MAX\_VALUE**);     System.**out**.println("\n long terkecil= " + Long.**MIN\_VALUE**);     System.**out**.println("long terbesar= " + Long.**MAX\_VALUE**);     System.**out**.println("\n float terkecil= " + Float.**MIN\_VALUE**);     System.**out**.println("float terbesar= " + Float.**MAX\_VALUE**);     System.**out**.println("\n double terkecil= " + Double.**MIN\_VALUE**);     System.**out**.println("double terbesar= " + Double.**MAX\_VALUE**);      int nilai1 = 200;     byte nilbyte = (byte) 200;     System.**out**.println("\n Nilai integer = " + nilai1);     System.**out**.println("Nilai byte= " + nilbyte);         int nilai2 = -40000;     short nilshort = (short) nilai2;     System.**out**.println("\n Nilai integer = " + nilai2);     System.**out**.println("Nilai short= " + nilshort); }    }  **Luaran:**  byte terkecil= -128 byte terbesar= 127  short terkecil= -32768 short terbesar= 32767  int terkecil= -2147483648 int terbesar= 2147483647  long terkecil= -9223372036854775808 long terbesar= 9223372036854775807  float terkecil= 1.4E-45 float terbesar= 3.4028235E38  double terkecil= 4.9E-324 double terbesar= 1.7976931348623157E308  Nilai integer = 200 Nilai byte= -56  Nilai integer = -40000 Nilai short= 25536  1.1. Analisa pengaruh dari ukuran tipe data terhadap data yang akan diolah!  1.2. Ubah nilai integer menjadi short dan byte. Simpulkan perubahan yang terjadi   1. Rincikan sumber informasi yang relevan (buku / webpage)  * <https://youtu.be/at27jZTFWe8> | | |
| **[Nomor Soal 1] Analisis dan Argumentasi** | | |
| 1). 1.1. Analisa pengaruh dari ukuran tipe data terhadap data yang akan diolah!   * Setiap tipe data memiliki rentang nilai yang berbeda. Misalnya:   + byte: -128 hingga 127   + short: -32,768 hingga 32,767   + int: -2,147,483,648 hingga 2,147,483,647   + long: -9,223,372,036,854,775,808 hingga 9,223,372,036,854,775,807 * Jika nilai yang ingin disimpan melebihi batas ini, akan terjadi overflow, yang dapat menyebabkan hasil yang tidak diinginkan.   2). 1.2. Ubah nilai interger menjadi short dan byte. Simpulkan perubahan yang terjadi   * Pada saat mengubah nilai data integer menjadi nilai data short dan byte, maka pada akan terjadi overflow. * Overflow adalah ketika nilai yang dimasukkan melebihi batas maksimum tipe data. Misalnya   + konversi dari int 200 ke byte menghasilkan -56 karena melampaui batas maksimum byte.   + Konversi int -40000 ke short menghasilkan 25536 karena melampaui batas maksimum short. | | |
| **[Nomor Soal] Penyusunan Algoritma dan Kode Program** | | |
| 1. Algoritma  * Membuat nama file dan nama kelas yang sama * Membuat method main yang bersifat public * Memasukan tipe data int dengan nilai data 200 yang akan dikonversikan kedalam tipe data byte * Membuat tipe data byte dengan nilai tipe data int yaitu 200 * Memasukan tipe data int dengan nilai data -40000 yang akan dikonversikan kedalam tipe data short * Membuat tipe data byte dengan nilai tipe data int yaitu -40000 * Membuat perintah untuk menampilkan luaran dari nilai int * Membuat perintah untuk menampilkan luaran dari nilai int yang sudah dikonversi menajadi byte * Membuat perintah untuk menampilkan luaran dari nilai int * Membuat perintah untuk menampilkan luaran dari nilai int yang sudah dikonversi menajadi short  1. Tuliskan kode program dan luaran   Codingan 1.1    Nama : Iqbal Tri Agustian  Npm : G1F024043  Luaran codingan 1.1    Nama: Iqbal Tri Agustian  Npm: G1F024043     * Codingan ini menampilkan luaran dari berbagai macam tipe data dengan nilai terkecil juga nilai terbesar dari masing masing tipe data * Codingan ini juga menampilkan luaran hasil konversi tipe data int ke tipe data lainnya   Codingan 1.2    Nama: Iqbal Tri Agustian  Npm: G1F024043  Luaran codingan 1.2    Nama: Iqbal Tri Agustian  Npm: G1F024043   * + Codingan ini menampilkan lauaran dari hasil konversi tipe data int menjadi tipe data byte dan juga tipe data short   + Luaran menunjukan hasil perubadan dari nilai data int yang berarti bahawa penggunaan tipe data yang tidak sesuai dapat mempengaruhi akurasi perhitungan. Misalnya, jika nilai yang diharapkan lebih besar dari batas tipe data, hasilnya bisa jauh dari ekspektasi. | | |
|  | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  * Ukuran tipe data sangat penting dalam pemrograman, karena mempengaruhi kemampuan untuk menyimpan dan memproses nilai dengan benar. Memahami batasan dan karakteristik masing-masing tipe data membantu menghindari kesalahan dan memastikan program berjalan dengan efisien dan akurat.  1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  * Pada pemrograman ini saya menambahkan tipe data byte pada nilai datanya karena dengan cara itulah codingan bisa berjalan tanpa terjadinya eror, begitu juga dengan tipe data short | | |

|  |  |  |
| --- | --- | --- |
| **Nama & NPM** | **Topik:** | **Tanggal:** |
| **Iqbal Tri Agustian**  **G1F024043** | **Format Specifer** | **30 Agustus 2024** |
| **[Nomor Soa Pengayaan 3] Identifikasi Masalah:** | | |
| 1. Uraikan permasalahan dan variabel   public class TipeData{   public static void main(String[] args) {     double num = 1974.9215;     System.**out**.println( "number=1974.9215" );     System.**out**.println( "Format |Position" );     System.**out**.println( "String |01234567890123456789" );     System.**out**.println( "----------+--------------------" );     printFloatWithFormat( "%f:", num );     printFloatWithFormat( "%+f:",num);     printFloatWithFormat( "%,f:",num);     printFloatWithFormat( "%+,f:",num);     printFloatWithFormat( "%20f:",num);     printFloatWithFormat( "%+20f:",num);     printFloatWithFormat( "%,20f:", num );     printFloatWithFormat( "%+,20f:", num );     printFloatWithFormat( "%-20f:", num );     printFloatWithFormat( "%+-20f:", num );     printFloatWithFormat( "%,-20f:", num );     printFloatWithFormat( "%+-,20f:",num);     printFloatWithFormat( "%020f:",num);     printFloatWithFormat( "%+020f:", num );     printFloatWithFormat( "%,020f:", num );     printFloatWithFormat( "%+,020f:",num);     System.**out**.println( "----------+--------------------" ); }  public static void printFloatWithFormat( String fmt, double v) {    System.**out**.printf( "%-10s|",fmt);    System.**out**.printf( fmt, v );    System.**out**.println(); }   }  Luaran:  number=1974.9215 Format |Position String |01234567890123456789 ----------+-------------------- %f:       |1974.921500: %+f:      |+1974.921500: %,f:      |1,974.921500: %+,f:     |+1,974.921500: %20f:     |         1974.921500: %+20f:    |        +1974.921500: %,20f:    |        1,974.921500: %+,20f:   |       +1,974.921500: %-20f:    |1974.921500         : %+-20f:   |+1974.921500        : %,-20f:   |1,974.921500        : %+-,20f:  |+1,974.921500       : %020f:    |0000000001974.921500: %+020f:   |+000000001974.921500: %,020f:   |000000001,974.921500: %+,020f:  |+00000001,974.921500: ----------+--------------------   1. 3.1 Analisa pengaruh dari specifier terhadap data yang akan ditampung! 2. 3.2 Konstruksikan kode specifier untuk tipe data boolean, integer, string dengan masing-masing 3 contoh format specifier!   2). Rincikan sumber informasi yang relevan (buku / webpage   * <https://docs.oracle.com/javase/8/docs/api/java/util/Formatter.html> * <https://www.w3schools.com/java/java_output_formatting.asp> * <https://www.youtube.com/results?search_query=java+output+formatting> | | |
| **[Nomor Soal Pengayaan 3] Analisis dan Argumentasi** | | |
| 1. 3.1 Analisa pengaruh dari specifier terhadap data yang akan ditampung!  * **%f:**   + Menampilkan angka dalam format desimal standar.   + Contoh luaran: 1974.921500 * **%+f:**   + Menampilkan angka dengan tanda positif (+) jika nilainya positif.   + Contoh luaran: +1974.921500 * **%,f:**   + Menampilkan angka dengan pemisah ribuan (comma).   + Contoh luaran: 1,974.921500 * **%+,f:**   + Menampilkan angka dengan tanda positif (+) dan pemisah ribuan.   + Contoh luaran: +1,974.921500 * **%20f:**   + Menampilkan angka dengan total lebar 20 karakter, diratakan ke kanan.   + Contoh luaran: 1974.921500 (ada spasi di depan) * **%+20f:**   + Menampilkan angka dengan tanda positif (+) dan total lebar 20 karakter.   + Contoh luaran: +1974.921500 * **%,20f:**   + Menampilkan angka dengan pemisah ribuan dan total lebar 20 karakter.   + Contoh luaran: 1,974.921500 * **%+,20f:**   + Menampilkan angka dengan tanda positif (+), pemisah ribuan, dan total lebar 20 karakter.   + Contoh luaran: +1,974.921500 * **%-20f:**   + Menampilkan angka dengan total lebar 20 karakter, diratakan ke kiri.   + Contoh luaran: 1974.921500 (ada spasi di belakang) * **%+-20f:**   + Menampilkan angka dengan tanda positif (+) dan total lebar 20 karakter, diratakan ke kiri.   + Contoh luaran: +1974.921500 * **%,-20f:**   + Menampilkan angka dengan pemisah ribuan, total lebar 20 karakter, diratakan ke kiri.   + Contoh luaran: 1,974.921500 * **%+-,20f:**   + Menampilkan angka dengan tanda positif (+), pemisah ribuan, total lebar 20 karakter, diratakan ke kiri.   + Contoh luaran: +1,974.921500 * **%020f:**   + Menampilkan angka dengan padding nol hingga total lebar 20 karakter, diratakan ke kanan.   + Contoh luaran: 00000000001974.921500 * **%+020f:**   + Menampilkan angka dengan tanda positif (+), padding nol hingga total lebar 20 karakter.   + Contoh luaran: +0000000001974.921500 * **%,020f:**   + Menampilkan angka dengan pemisah ribuan dan padding nol hingga total lebar 20 karakter.   + Contoh luaran: 000000001,974.921500 * **%+,020f:**   + Menampilkan angka dengan tanda positif (+), pemisah ribuan, dan padding nol hingga total lebar 20 karakter.   + Contoh luaran: +000000001,974.921500      1. 3.2 Konstruksikan kode specifier untuk tipe data boolean, integer, string dengan masing-masing 3 contoh format specifier!  Format Specifier yang Digunakan  * Boolean:   + %b : Menampilkan nilai boolean dalam format default (true/false).   + %B : Menampilkan nilai boolean dalam huruf kapital (TRUE/FALSE).   + %s : Menampilkan nilai boolean dengan kondisi ternary (boolValue ? "True" : "False"), yang menghasilkan "True" jika boolValue adalah true, dan "False" jika false. * Integer:   + %d : Menampilkan nilai integer dalam format desimal.   + %x : Menampilkan nilai integer dalam format heksadesimal (huruf kecil).   + %05d : Menampilkan nilai integer dengan padding nol di depan hingga mencapai 5 digit. * String:   + %s : Menampilkan string dalam format default.   + %.5s : Menampilkan hanya 5 karakter pertama dari string.   + %-20s : Menampilkan string yang diratakan ke kiri dengan lebar 20 karakter (spasi ditambahkan di sebelah kanan jika kurang dari 20 karakter). | | |
| **[Nomor Soal Pengayaan 3 ] Penyusunan Algoritma dan Kode Program** | | |
| 1. Rancang desain solusi atau algoritma  * Mebuat nama file dan nama kelas yang sama * Membuat method main bersifat public * Membuat tipe data boolean dengan nilai data true * Membuat tipe data int dengan nilai data 12345 * Membuat tipe data string dengan nilai data “Hello, World!” * Membuat perintah untuk menampilkan luaran dari tipe data boolean yangsudah diberi format specifier * Membuat perintah untuk menampilkan luaran dari tipe data int yangsudah diberi format specifier * Membuat perintah untuk menampilkan luaran dari tipe data stringyangsudah diberi format specifier  1. Tuliskan kode program dan luaran   Codingan 3.2    Nama: Iqbal Tri Agustian  Npm: G1F024043  Luaran 3.2    Nama: Iqbal Tri Agustian  Npm: G1F024043   * **Codingan ini menampilkan luaran dari format specifier dari tipe data boolean, int, dan juga string** | | |
| **[Nomor Soal] Kesimpulan** | | |
| 1. Analisa 2. Susunlah kesimpulan berdasarkan permasalahan, algoritma, dan kode program!  * Program ini bertujuan untuk mendemonstrasikan penggunaan spesifier format dalam menampilkan berbagai tipe data di Java, termasuk boolean, integer, dan string. Hal ini penting untuk memahami cara format output dalam pemrograman, yang dapat mempengaruhi cara informasi disajikan kepada pengguna.  1. Apakah dasar alasan pengambilan keputusan Anda untuk kasus ini?  * Menggunakan format yang tepat membantu meningkatkan keterbacaan output. Dengan menampilkan boolean dalam huruf kapital atau mengatur padding untuk integer, pengguna dapat dengan mudah memahami informasi yang disajikan. | | |