1. **ESSAY (100%)**
2. **[20%]** Pada tipe data abstrak:
3. Apa **persyaratan** desain bahasa untuk bahasa pemrograman yang mendukung tipe data abstrak?
4. Apa **masalah** dari desain dalam bahasa pemrograman dengan tipe data abstrak?
5. Apa **tujuan** dari destructor dalam C++?
6. Apa itu namespace dalam C++? Apa sajakah tujuannya?

**Jawaban:**

1. **Persyaratan Desain Bahasa untuk mendukung tipe data abstrak:**
2. Unit sintaksis untuk Enkapsulasi tipe dari inisialisasi.
3. Metode membuat jenis nama dan header subprogram terlihat oleh klien, sambil menyembunyikan definisi sebenarnya.
4. Beberapa operasi primitif harus dibangun ke dalam Language Processor.
5. **Masalah desain bahasa untuk tipe data abstrak adalah:**
6. Masalah desain pertama adalah bentuk Container untuk Interfaces ke Types.
7. Masalah desain kedua adalah apakah tipe data abstrak dapat diparameterisasi.
8. Masalah desain ketiga adalah bagaimana kontrol akses disediakan dan bagaimana kontrol tersebut ditentukan. Di sini, kontrol akses berarti Visibilitas Objek.
9. **Tujuan dari Destructor di C++:**

* Melakukan operasi pembersihan jika ada objek yang memiliki koneksi yang berhubungan dengan Database yang dapat kita tutup sebelum objek akan dihancurkan.

1. **Maksud NameSpace Di C++:**

* Namespace dapat diibaratkan seperti wadah atau grup yang digunakan untuk mengelompokkan fungsi atau data lainnya sehingga menjadi dapat dibedakan.

**Tujuan NameSpace Di C++:**

* Untuk menghindari Konflik penggunaan nama class, fungsi, variable. Jadi isi dari suatu namespace bisa berupa class, fungsi / method, dan variable.

1. **[20%]** Pada pemrograman berorientasi objek:
2. Apa perbedaan **method overriding** dan **method overloading**? Jelaskan contoh skenario penggunaan method overriding dan method overloading dalam class.
3. Terdapat 3 class pada sebuah program. Class tersebut adalah class **customer**, class **kamar**, dan class **transaksi**. Class customer berisi data terkait customer dan class kamar berisi data terkait kamar. Sedangkan pada class transaksi harus terdapat data customer dan kamar yang dipesan. Berdasarkan kasus tersebut, buatlah **multiple inheritance** dari class tersebut yang terdiri dari base class dan derived class.

**Jawaban:**

1. **Perbedaan Method Overriding dan Method Overloading:**

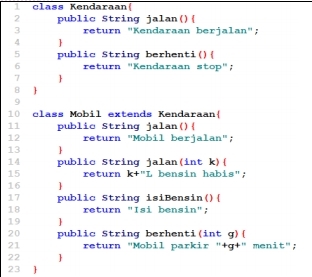
**Method Overriding**

* Memiliki **nama method yang sama**, tapi memiliki **parameter dan tipe data yang berbeda** di **dalam satu class yang sama**.

**Method Overloading**

* Memiliki nama **method yang sama**, di **class yang berbeda**, tapi **menghasilkan Output yang berbeda**.

**Contoh Skenario:**



* Fungsi **jalan()** pada **line 11** dan **line 14** Merupakan **Method Overloading**
* Fungsi **jalan()** pada **line 11** dan **line 2** Merupakan **Method Overriding**

1. **Multiple Inheritance dengan Based Class dan Derived Class (Dengan C++):**

#include <iostream>

**using** **namespace** std**;**

class customer **{**

public**:**

string customerNIM **=** "2201741971"**;**

string namaCustomer**=**"EDWARD"**;**

**};**

class kamar **{**

public**:**

string kodeKamar **=** "400"**;**

string LuasatauSempit **=** "Luas"**;**

**};**

class Transaksi **:** public customer**,** public kamar **{**

public**:**

string nomorTransaksi **=** "081289434641"**;**

**};**

int main**()** **{**

Transaksi obj**;**

cout**<<**"NIM Customer = "**<<** obj**.**customerNIM **<<**endl**;**

cout**<<**"Nama Customer = "**<<** obj**.**namaCustomer **<<**endl**;**

cout**<<**"Kode Kamar = "**<<** obj**.**kodeKamar **<<**endl**;**

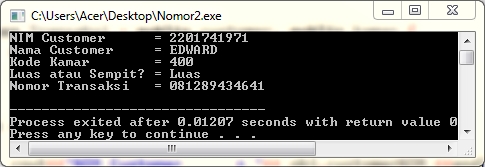
cout**<<**"Luas atau Sempit? = "**<<** obj**.**LuasatauSempit **<<**endl**;**

cout**<<**"Nomor Transaksi = "**<<** obj**.**nomorTransaksi **<<**endl**;**

**return** 0**;**

**}**

**OUTPUT:**



1. **[15%]** Pada bahasa pemrograman yang mendukung concurrency dengan multithreading, dimungkinkan terdapat lebih dari satu thread of control yang berjalan, pada satu program. Pertanyaan: apa yang akan terjadi apabila terdapat dua thread yang ingin mengubah nilai sebuah variabel **yang dapat diakses bersama**, dengan **x = x + 1**, dimana **x** (sebuah integer, nilai awal: 0) adalah nama variabel tersebut? Diasumsikan bahwa masing-masing thread melakukan **100.000 kali** operasi tersebut berturut-turut. Tulislah sebuah program sederhana dan tampilkan outputnya untuk mendukung jawaban Anda.

**Jawaban:**

**Yang akan Terjadi Adalah:**

* **Jika dua Thread mengakses variabel secara bersamaan, dan meningkatkan Counter (x) seperti yang disebutkan dalam Soal**, **ada kemungkinan bahwa nilai akhir Counter (x) akan kurang dari jumlah total peningkatan dari Thread yang digabungkan**.
* Jadi, misalnya, dalam Soal dua Thread menambah Counter (x) sebesar 1, dan setiap Thread melakukan ini 100.000 kali. Jadi, nilai akhirnya harus 200.000. **Tetapi, karena mereka mengakses Counter secara bersamaan, nilai akhirnya Bisa Kurang Dari 200.000**.

**CODE (Dengan Bahasa Pemrograman JAVA):**

public class Counter **{**

// Variabel Penghitung yang diakses oleh dua Thread.

private static int x **=** 0**;**

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

// Jumlah iterasi untuk setiap Thread tempat mereka menambah counter.

int jumlahIterasi **=** 100000**;**

// Inisialisasi Thread Pertama.

// Thread menambah counter di setiap iterasi.

Thread threadOne **=** **new** Thread**(()** **->** **{**

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

x **=** x **+** 1**;**

**}**

**});**

// Inisialisasi Thread Kedua.

// Thread menambah counter di setiap iterasi.

Thread threadTwo **=** **new** Thread**(()** **->** **{**

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

x **=** x **+** 1**;**

**}**

**});**

// Memulai threads.

threadOne**.**start**();**

threadTwo**.**start**();**

// Menunggu Threads Hingga Selesai.

threadOne**.**join**();**

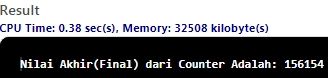
threadTwo**.**join**();**

// Cetak hasil Final untuk Counter.

System**.**out**.**println**(**"Nilai Akhir(Final) dari Counter Adalah: " **+** x**);**

**}**

**}**



**Nilai akhir setelah Thread selesai Bisa Kurang Dari** **200.000**, dan **akan bervariasi dari satu eksekusi ke yang lain**.

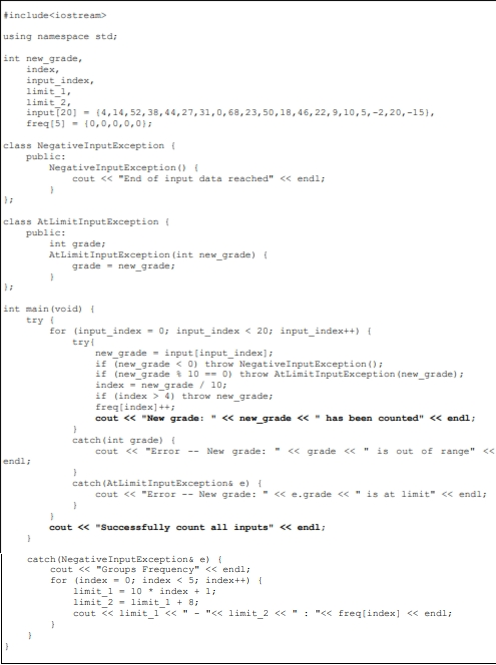
**Ini karena kedua Thread dapat membaca nilai pada saat yang sama**, dan menambahkannya bersamaan, dan **nilai akhir hanya bertambah satu kali bukan 2 kali**.

**Sebagai contoh, anggap nilai penghitungnya 5, dan kedua Thread membaca nilai dan menambahnya dengan 1, dan mengaturnya menjadi 6**.

**Dengan demikian, nilai akhir menjadi 6, di mana seolah-olah mereka membaca penghitung secara terpisah satu demi satu, nilainya seharusnya 7**.

1. **[15%]** Pada penanganan eksepsi:
2. Tuliskan program penanganan eksepsi sederhana untuk menangani pembagian nol:
3. Dengan menggunakan keyword **try-catch**
4. Dengan menggunakan keyword **finally**
5. Kode berikut adalah bagian dari sebuah program C++.
6. Tujuan dari program tersebut dibuat?
7. Berapa kali object dari class NegativeInputException dibuat?
8. Berapa kali object dari class AtLimitInputException dibuat?
9. Berapa kali kode di baris pertama yang ditebalkan dalam kode dijalankan?

**Kode: cout << "New grade: " << new\_grade << " has been counted" << endl;**



**Jawaban:**

1. **Program Penanganan eksepsi sederhana untuk menangani pembagian nol:**
2. **Menggunakan Keyword try-catch (Dengan C++):**

#include <iostream>

**using** **namespace** std**;**

double division**(**int a**,** int b**)**

**{**

**if(** b **==** 0 **)**

**{**

**throw** "Eksepsi: Penyebut Bernilai Nol atau Bukan Bilangan"**;**

**}**

**return** **(**a**/(**float**)**b**);**

**}**

int main **()**

**{**

int pembilang**,** penyebut **;**

double hasil **=** 0**;**

cout**<<**"Masukkan Pembilang: "**;**

cin**>>**pembilang**;**

cout**<<**"Masukkan Penyebut: "**;**

cin**>>**penyebut**;**

**try** **{**

hasil **=** division**(**pembilang**,** penyebut**);**

cout **<<** "Hasil Pembagian : "**<<** hasil **<<** endl**;**

**}catch** **(**const char**\*** msg**)** **{**

cerr **<<** msg **<<** endl**;**

**}**

**return** 0**;**

**}**

1. **Menggunakan Keyword Finally (Dengan Bahasa Pemrograman JAVA)**

**import** java**.**util**.\*;**

public class Main **{**

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

public double pembagianDenganNol**(**int x**,** int y**)** **{**

**if(** y **==** 0 **)** **{**

System**.**out**.**println**(**"Eksepsi: Penyebut Bernilai Nol atau Bukan Bilangan"**);**

**}**

**return** **(**x**/(**double**)**y**);**

**}**

public Main**()** **{**

double hasil **=** 0**;**

System**.**out**.**print**(**"Masukkan Pembilang: "**);**

int pembilang **=** scan**.**nextInt**();**

System**.**out**.**print**(**"Masukkan Penyebut: "**);**

int penyebut **=** scan**.**nextInt**();**

**try** **{**

hasil **=** pembagianDenganNol**(**pembilang**,**penyebut**);**

**if(**penyebut**!=**0**){**

System**.**out**.**println**(**"Hasil Pembagian : "**+**hasil**);**

**}**

**}catch** **(**ArithmeticException ae**)** **{**

System**.**out**.**println**(**"Eksepsi: Penyebut Bernilai Nol atau Bukan Bilangan!"**);**

**}finally** **{**

System**.**out**.**println**(**"Program Selesai Dijalankan"**);**

**}**

**}**

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

**new** Main**();**

**}**

**}**

1. **Bagian dari sebuah program C++:**
   * 1. **Tujuan dari program tersebut dibuat?**
     + Untuk **menemukan frekuensi rentang dalam array yang diberikan**.
     + Program ini **mengabaikan angka yang merupakan kelipatan dari 10** (AtLimitInputException) **dan angka yang di atas 50** (di luar jangkauan).
     + **Jika nomor dalam array yang diberikan negatif**, maka akan **menganggapnya sebagai akhir array (NegativeInputException)**.
     + Setiap kali menghitung angka itu akan mencetak nomor itu. **Jika jumlahnya lebih dari 50**, itu akan **mencetak "is out of range"**.
     + **Jika nomornya kelipatan 10**, maka itu akan **mencetak AtLimitInputException**.
     + **Jika angka negatif**, itu akan **mencetak NegativeInputException dan keluar dari loop**.
     + Akhirnya akan mencetak frekuensi.
     1. **Berapa kali object dari class NegativeInputException dibuat?**
     + Objek NegativeInputException **hanya SEKALI dibuat ketika mencapai angka -2 (indeks 17)**.
     1. **Berapa kali object dari class AtLimitInputException dibuat?**
     + Objek AtLimitInputException **hanya TIGA KALI dibuat ketika angkanya 0, 50, 10**.
     1. **Berapa kali kode di baris pertama yang ditebalkan dalam kode dijalankan?**

**Kode: cout << "New grade: " << new\_grade << " has been counted" << endl;**

* + - **Program ini akan mengeksekusi baris kode yang ditebalkan sebanyak 12 KALI** (ketika nilai new\_grade adalah 4, 14, 38, 44, 27, 31, 23, 18, 46, 22, 9, 5).

1. **[15%]** Pada bahasa pemrograman functional:
   * + 1. Berikut adalah pernyataan dalam bahasa pemrograman Scheme.
          1. Dari fungsi untuk memanipulasi list berikut ini, apakah hasil dari:

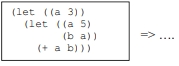
(car ‘(1 2 3 4 5)) => ….

(cdr ‘(1 2 3 4 5)) => ….

(cons 1 ‘(2 3)) => ….

Isi titik-titik kosong di atas kemudian jelaskan perbedaan car, cdr, dan cons.

* + - * 1. Dari scope bercabang berikut, nilai apakah yang akan dihasilkan?



* + - 1. Tuliskan hasilnya

(lambda (a b x y) + ( \* b x y) (+ a x y))

Jika a = 2, b =3, x = 4, y = 5

* + - 1. Apa hasil dari:
         1. (eqv? 2.8 (- 3 0.2))
         2. (caddar ‘( (A (B C) D E (F G) H) I J) )
         3. (+ (\* 5 6) (\* 8 3) 10)

**Jawaban:**

1. **Hasil dari:**
2. (car ‘(1 2 3 4 5)) => **1**

(cdr ‘(1 2 3 4 5)) => **2 3 4 5**

(cons 1 ‘(2 3)) => **1 2 3**

**Penjelasan Perbedaan:**

**CAR mengembalikan elemen pertama dari List.**

**CDR mengembalikan elemen kedua dari List sampai terakhir.**

**CONS menggabungkan dua nilai dari List yang ada.**

1. (let ((a 3)) (let ((a 5) (b a)) (+ a b))) => **8**
   * Pertama, Nilai a adalah 3
   * Kemudian, Ada Fungsi tersendiri dimana a = 5
   * Memasukkan nilai a ke variable b , jadi b bernilai 5
   * Lalu, Tambahkan nilai a dan b, dimana itu di luar Fungsi independen
   * Maka, a = 3, dan b=5, sehingga a+b = 3 + 5 = 8
2. **Hasil dari (lambda (2 3 4 5) + ( \* 3 4 5) (+ 2 4 5)) adalah 11, Karena:**

* **Ekpresi Dasar nya adalah:** **(lambda (parameter) body)**
* Jadi, (2 3 4 5) adalah Parameternya.
* (\* 3 4 5) tidak dijalankan, karena hanya blok Terakhir yang dijalankan.
* Maka, Hanya (+ 2 4 5) yang berjalan dan aktif.
* **Dimana 2 + 4 + 5 = 11, Maka Hasilnya adalah 11**.

1. **Hasil dari:**
2. **(eqv? 2.8 (- 3 0.2)) => FALSE**

* Fungsi eqv memiliki makna EQUIVALENT atau SETARA.
* Jika sama, akan Mengeluarkan TRUE, dan Jika beda mengeluarkan FALSE.
* **Jadi, Ketika (-2.8) dan (-30.2) akan menghasilkan FALSE karena beda.**

1. **(caddar ‘( (A (B C) D E (F G) H) I J) ) => ERROR**

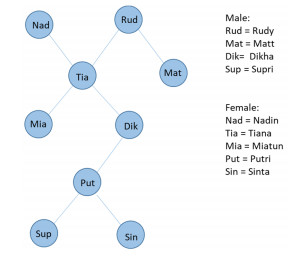
* Formula dari Caddar (x) = (car (cdr (cdr (car '(x))
* Pertama, ketika car dieksekusi, akan menghasilkan ((A (B C) D E (F G) H)).
* Kedua, ketika cdr dieksekusi, akan menghasilkan H.
* Ketiga, ketika cdr dieksekusi lagi, akan **ERROR**, karena tidak ada nilainya.

1. **(+ (\* 5 6) (\* 8 3) 10) => 64**

* Pertama, (\* 5 6), hasilnya 5 \* 6 = 30.
* Kedua, (\* 8 3), hasilnya 8 \* 3 = 24.

**Ketiga, (+ 30 24 10), hasilnya 30 + 24 + 10 = 64**.

1. **[15%]** Pada bahasa pemrograman logic. Perhatikanlah gambar berikut dan buatlah Fact dari object Female, Male, Parent pada Prolog programming.



**Jawaban:**

1. **CHILD b) MOTHER c) FATHER d) SON**

child(Matt) mother(Putri,Sinta) father(Dikha,Putri) son(Matt,Rudy)

child(Miatun) mother(Putri,Supri) father(Rudy,Tiana) son(Dikha,Tiana)

child(Sinta) mother(Tiana,Dikha) father(Rudy,Matt) son(Supri,Putri)

child(Supri) mother(Tiana,Miatun)

mother(Nadin,Tiana)

**e) DAUGHTER f) SISTER g) BROTHER**

daughter(Tiana,Rudy) sister(Sinta,Supri) brother(Supri,Sinta)

daughter(Tiana,Nadin) sister(Miatun,Dikha) brother(Dikha,Miatun)

daughter(Miatun,Tiana) sister(Tiana,Matt) brother(Matt,Tiana)

daughter(Putri,Dikha)

daughter(Sinta,Putri)

**h) UNCLE i) AUNT j) NIECE k) NEPHEW**

uncle(Matt,Dikha) aunt(Miatun,Putri) niece(Miatun,Matt) nephew(Dikha,Matt)

uncle(Matt,Miatun) niece(Putri,Miatun)

**l) GRANDFATHER m) GRANDMOTHER n) GRANDSON**

grandfather(Dikha,Sinta) grandmother(Tiana,Putri) grandson(Supri,Dikha)

grandfather(Dikha,Supri) grandmother(Nadin,Dikha) grandson(Dikha,Rudy)

grandfather(Rudy,Dikha) grandmother(Nadin,Miatun) grandson(Dikha,Nadin)

grandfather(Rudy,Miatun)

**o) GRANDDAUGHTER**

granddaughter(Sinta,Dikha)

granddaughter(Putri,Tiana)

granddaughter(Miatun,Rudy)

granddaughter(Miatun,Nadin)