**TM 02**

1. *Copy constructor* suatu klas dieksekusi jika obyek klas di-*copy* dengan mekanisme di bawah ini kecuali

a. direct assignment b. initialization

c. passing by value c. return by value

1. Fungsi anggota yang otomatis di-*create* oleh kompilator meskipun ia tidak secara eksplisit ditulis oleh pemrogram dalam definisi klas adalah opsi-opsi di bawah ini kecuali

a. default constructor b. copy constructor

c. destructor d. overloaded constructor

1. Level akses *utility function* atau *helper function* suatu klas adalah

a. public b. protected c. private d. inline

1. Untuk class X, sintaks *copy constructor* adalah

a. X(& X) b. X(const X&)

c. X(const &X) d. X(X &)

1. *C++ class* mempunyai
   1. satu konstruktor dan satu destructor
   2. lebih dari satu konstruktor dan satu destructor
   3. satu konstruktor dan lebih dari satu destructor
   4. lebih dari satu konstruktor dan lebih dari satu destructor
2. Apa keluaran dari penggalan program berikut ini:

class Rational {

public:

Rational() { cout << “OBYEK LAHIR.\n”; }

~Rational() { cout << “OBYEK MATI. \n”; }

private:

int num, den;

};

main() {

{ Rational x;

cout << “SEKARANG x HIDUP.\n”;

}

cout << “ANTAR BLOK.\n”;

{ Rational y;

cout << “SEKARANG y HIDUP.\n”;

}

}

1. Uraikan apa yang kamu ketahui tentang constructor, user defined constructor, default constructor, parameterized constructor, constructor initialization list, copy constructor dan destructor.
2. Uraikan apa yang kamu ketahui tentang *default arguments* di C++. Sebutkan apa saja yang dapat digunakan sebagai *devalut value*. Mengapa nilai *constant* tidak dapat digunakan sebagai *default value* untuk *reference variable*?
3. Apa yang dimaksud dengan *function overloading*? Kriteria apa yang digunakan untuk menyatakan bahwa dua fungsi dengan nama yang sama adalah berbeda satu sama lain.
4. Banyak penulis buku menyatakan bahwa untuk *class* X paling sedikit mempunyai satu *constructor* yaitu *default constructor* dengan prototype X(). Tetapi John Hubbard dalam bukunya *Programming with C++* menyatakan bahwa setiap *class* X paling sedikit mempunyai dua *constructor* yaitu *default* X() dan *copy constructor* yang prototipe sintaksnya adalah *X (const X & )*, fungsi anggota dengan satu parameter dan *pass by constant reference*.

Tugas Anda:

a. Kapan *copy constructor* dieksekusi?

b. Untuk penggalan kode berikut berapa kali *copy constructor* dieksekusi?

class Rational {

public : Rational(int n, int d) : num(n), den (d) { }

Rational(const Rational & r) : num(r.num), den(r.den)

{ cout << “COPY CONSTRUCTOR CALLED\n”;}

private: int num, den;

};

Rational f(Rational r) {

Rational s = r;

return s;

}

main() {

Rational(22,7);

Rational y(x);

f(y);

}

**I AM PROUD AS A BINUSIAN**

1. a. direct assignment
2. a. default constructor
3. c. private
4. b. X(const X&)
5. b. lebih dari satu konstruktor dan satu destructor
6. OBYEK LAHIR

SEKARANG x HIDUP

OBYEK MATI

ANTAR BLOK

OBYEK LAHIR

SEKARANG y HIDUP

OBYEK MATI

1. Constructor → Constructor adalah blok penyataan khusus yang dibuat ketika sebuah objek dibuat

User Defined Constructor → Sebuah constructor yang dibuat oleh user itu sendiri

Default Constructor → Sebuah constructor yang dibuat secara otomatis oleh compiler yang memiliki nama yang sama dengan classnya

Parameterized Constructor → Constructor yang memiliki parameter untuk memberikan nilai awal pada objek yang ingin dibuat

Constructor Initialization List → Cara lain menginisialisasi nilai-nilai parameter awal dari sebuah objek suatu class. Diletakkan di definisi fungsi constructor, sebelum body constructor tersebut.

Copy Constructor → Merupakan construct dengan satu parameter dengan tipe yang sama yang ditetapkan untuk setiap anggota variabel class nonstatik objek yang datanya disalin dari objek sebelumnya.

Destructor → Fungsi yang menghapus atau membuang objek. Nama fungsinya sama dengan nama class, tetapi ditambahkan (~).

1. Default Argument adalah bagian dari deklarasi fungsi yang menginstruksikan compiler nilai apa yang harus dikirim untuk sebuah argumen jika programmer sengaja tidak memberikan argument ketika memanggil fungsi

Default Value → Semua tipe data kecuali constant

Constant tidak dapat digunakan sebagai default value untuk reference variable karena nilai constant tidak dapat dirubah

1. Function Overloading → Function overloading adalah suatu keadaan dimana beberapa fungsi memiliki nama yang sama, tetapi argumennya berbeda

Kriteria :

* Nama fungsinya sama
* Urutan parameter
* Jumlah parameter

1. a. Copy Constructor akan dieksekusi ketika fungsi dari copy construct tersebut dipanggil

b. 4 kali

**Nama : Henricko Randy Setiawan**

**NIM : 1701293931**

**Kelas : 02PPT**