



Tentamen

C++ programmeermethoden

Bachelor Kunstmatige Intelligentie

1e Deeltentamen

Datum: 25 maart 2018

Tijd: 18.00-20.00

Aantal pagina's: 8 (inclusief voorblad)

Aantal vragen: 5

Maximaal aantal te behalen punten: 10

VOORDAT U BEGINT

- **Wacht** tot u de instructie krijgt het tentamen te openen.
 - Controleer of uw versie van het tentamen compleet is.
 - Schrijf uw **naam en studentnummer en indien van toepassing versienummer op elk vel papier** dat u inlevert en **nummer de pagina's**.
 - U dient uw **mobiele telefoon** uit te schakelen en te bewaren in uw jas of tas.
Uw **jas en tas** moeten onder uw tafel liggen.
 - **Toegestane hulpmiddelen:** boek & notities & laptop (alleen voor lezen van ebook!).
-



HUISHOUELIJKE MEDEDELINGEN

- De eerste 30 minuten en de laatste 15 minuten mag u de zaal niet verlaten, ook niet voor het bezoeken van het toilet.
- Op verzoek van de examiner (of diens vertegenwoordiger) moet u zich kunnen legitimeren met een bewijs van inschrijving of een geldig legitimatiebewijs.
- Tijdens het tentamen is toiletbezoek niet toegestaan, tenzij de surveillant hier toestemming voor geeft.
- 15 minuten voor het eind wordt u gewaarschuwd dat het inlevertijdstip nadert.
- Vul indien van toepassing na afloop van het tentamen alstublieft het evaluatieformulier in.

Succes!



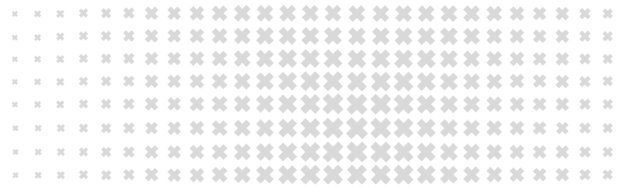
C++ programmeermethoden Deeltoets 1

Maandag 25 Maart, 18:00-20:00, World Fashion Center Oosthal

Bas Terwijn

Schakel de Wifi en eventuele andere communicatie-mogelijkheden (bluetooth, telefonie, mobiel internet, etc.) van uw mobiel, laptop, tablet, e-reader, etc. uit.

Gebruik uw laptop, tablet of e-reader alleen voor het lezen van uw ebook. Andere windows (bv. een command prompt of IDE) kunnen worden aangezien als fraude zoals beschreven in de 'UvA Fraude- en plagiaatsregeling'. Fraude kan leiden tot uitsluiting van deelname aan dit vak en in het uiterste geval tot beëindiging van de inschrijving bij opleidingen van de UvA. Kom dus niet in de verleiding en houd ook de schijn tegen.



Vraag 1 (2 punten)

Geef voor de condities in onderstaande if-statements aan of ze gegarandeerd waar of niet gegarandeerd waar zijn.

```
#include <iostream>

int main()
{
    int dividend = 7184684, divisor = 823633;
    int quotient = dividend / divisor;
    int remainder = dividend % divisor;
    if ( dividend == quotient * divisor + remainder )
        std::cout<<"A: guaranteed true\n";
    else
        std::cout<<"A: not guaranteed true\n";

    double d1 = 5.0 * (1.0 / 3.0);
    double d2 = (5.0 * 1.0) / 3.0;
    if ( d1 == d2 )
        std::cout<<"B: guaranteed true\n";
    else
        std::cout<<"B: not guaranteed true\n";

    d1 = 3 / static_cast<double>(9.0);
    d2 = static_cast<double>(3) / 9;
    if ( d1 == d2 )
        std::cout<<"C: guaranteed true\n";
    else
        std::cout<<"C: not guaranteed true\n";

    const double& dref = d1;
    d1 = 1 / 3.0 + dref;
    if ( d1 == dref )
        std::cout<<"D: guaranteed true\n";
    else
        std::cout<<"D: not guaranteed true\n";

    if ( 200 > 20 > 2 )
        std::cout<<"E: guaranteed true\n";
    else
        std::cout<<"E: not guaranteed true\n";
    return 0;
}
```



Vraag 2 (1 punt)

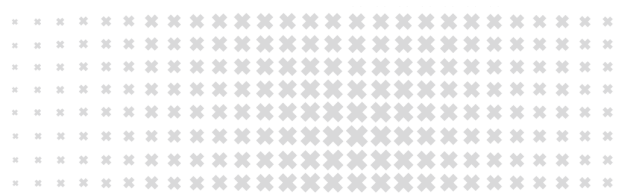
Wat is de uitvoer van onderstaand programma?

```
#include <iostream>

int    func(int a,int b)           { std::cout<<"1\n"; }
double func(const double& a,double b) { std::cout<<"2\n"; }
int    func(int a,const double& b)   { std::cout<<"3\n"; }
void   func(double a,const int b)     { std::cout<<"4\n"; }

int main()
{
    int  i1=1, i2=2;
    double d1=3, d2=4;
    std::cout<<"A: "; func( i1 , static_cast<double>(i2) );
    std::cout<<"B: "; func( d1 , i2 );
    std::cout<<"C: "; func( static_cast<char>(i1) ,
                           static_cast<char>(i2) );

    {
        std::cout<<"D: "; func( d1 , static_cast<int>(d2) );
        std::cout<<"E: "; func( static_cast<short>(d1) , d2 );
    }
    return 0;
}
```



Vraag 3 (2 punten)

Wat is de uitvoer van onderstaand programma?

```
#include <iostream>
using namespace std;

int myVar = 0;

int myFunction1( int myVar )
{
    {
        cout<<"A: "<< myVar <<endl;
        int myVar = 1;
        cout<<"B: "<< myVar <<endl;
    }

    cout<<"C: "<< myVar <<endl;

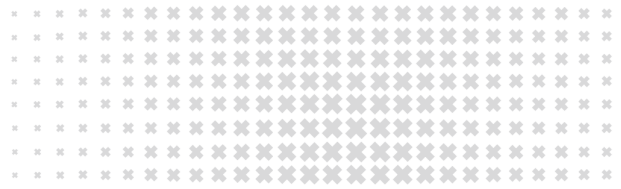
    for (int myVar = 3; myVar < 4; myVar++)
        cout<<"D: "<< myVar <<endl;

    {
        myVar = 4;
    }

    cout<<"E: "<< myVar <<endl;
    return myVar + 1;
}

void myFunction2( int& myVar )
{
    cout<<"F: "<< myVar <<endl;
    myVar = 5;
}

int main()
{
    myFunction1( 2 );
    cout<<"G: "<< myVar <<endl;
    myFunction2( myVar );
    cout<<"H: "<< myVar <<endl;
    return 0;
}
```



Vraag 4 (3 punten)

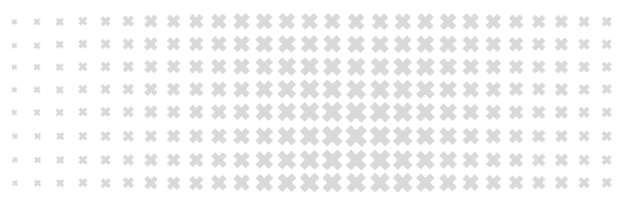
A. Schrijf een class waarmee de gebruiker door het aanroepen van member functies:

- een “double” waarde aan het einde kan toevoegen aan een intieel lege lijst, met maximale grootte van 10000
- de “double” waarde aan het einde van de lijst kan verwijderen
- op een efficiënte manier steeds de gemiddelde waarde van alle “doubles” in de lijst kan opvragen

Het moet voor de gebruiker duidelijk zijn wanneer het toevoegen/verwijderen is mislukt (wanneer de lijst al vol/leeg is). Geef de waarde 0 voor de gemiddelde waarde van een lege lijst. Gebruik bij de implementatie van deze class ten minste één keer de “++” operator, “--” operator en de conditional operator.

B. Gebruik keyword “const” op zoveel mogelijk plekken in uw class en leg uit wat in het algemeen het belang is van het gebruik van “const”.

C. Leg aan de hand van uw class uit wat “encapsulation” betekent, en leg uit waarom het in het algemeen zo belangrijk is.



Vraag 5 (2 punten)

Schrijf de implementatie voor de 4 conversie-functies in het onderstaande programma met gebruik van de “stringstream” class. Als een string niet naar een “int” of “double” geconverteerd kan worden, geef dan een “0” waarde terug.

```
#include <iostream>
#include <sstream>

// conversion functions
std::string int_to_string(int a);
int         string_to_int(std::string a);
std::string double_to_string(double a);
double     string_to_double(std::string a);

int main()
{
    std::cout<< int_to_string( 1234 ) <<' \n';
    std::cout<< string_to_int( "4321" ) <<' \n';
    std::cout<< double_to_string( 12.34 ) <<' \n';
    std::cout<< string_to_double( "43.21" ) <<' \n';
    std::cout<< string_to_double( "ABCDE" ) <<' \n';
    return 0;
}
```

The End!