Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

Tentamen

C++ programmeermethoden

Bachelor Kunstmatige Intelligentie

1e Deeltentamen
Datum: 30 maart 2022
Tijd: 9.00-11.00

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 en slides!).

HUISHOUDELIJKE 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 examinator (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.

Succes!

C++ programmeermethoden Deeltoets 1

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 inschijving bij opleidingen van de UvA. Kom dus niet in de verleiding en houd ook de schijn tegen.

Vraag 1: Arithmetic Operators and Expressions (2 punten)

Wat is de uitvoer van het onderstaande programma?

```
#include <iostream>

int main()
{
   int n1 = 73; // a prime number
   int n2 = 67; // a prime number
   int n3 = 59; // a prime number

   if ( n1 == (n1 / n2) * n2 )
      std::cout<<"A: True\n";
   else
      std::cout<<"A: Flase\n";

   if ( n1 == (n1 / n2) * n2 + n1 % n2 )
      std::cout<<"B: True\n";
   else
      std::cout<<"B: False\n";

   if ( n1 == (n1 / n2) * n2 + n2 % n1 )
      std::cout<<"C: True\n";
   else
      std::cout<<"C: False\n";

   if (n1 / n2 <= n1 / static_cast<double>(n2) )
      std::cout<<"D: True\n";
   else
      std::cout<<"D: False\n";

   if ( n1 * n2 / n3 == n1 * (n2 / n3) )
      std::cout<<"E: True\n";
   else
      std::cout<<"E: False\n";
}
```

Vraag 2: Variable Scope (2 punt)

Wat is de uitvoer van het onderstaande programma?

```
#include <iostream>

int var = 1;

void some_function()
{
   std::cout<< "B: "<< var <<'\n';
   int var = 11;
   std::cout<< "C: "<< var <<'\n';
}

int main()
{
   std::cout<< "A: "<< var <<'\n';
   int var = 111;
   some_function();
   std::cout<< "D: "<< var <<'\n';
   {
      int var = 1111;
   }
   std::cout<< "E: "<< var <<'\n';
}
```

Vraag 3: Algorithm (2 punten)

Wat is de uitvoer van het onderstaande programma?

```
#include <iostream>

void print(const int a[], const int size)
{
   for (int i=0; i<size; i++)
      std::cout<< a[i] <<' ';
   std::cout<<'\n';
}

void do_something(int a[], int size)
{
   bool done = false;
   while (!done)
   {
      done = true;
      for (int i = 0; i < size - 1; ++i)
         if (a[i] < a[i+1])
         {
            int temp = a[i];
            a[i] = a[i+1];
            a[i+1] = temp;
            done = false;
         }
      --size;
   }
}

int main()
{
   int size=9;
   int array[]={4, 7, 2, 5, 3, 1, 8, 6, 9};
   do_something(array, size);
   print(array, size);
   return 0;
}
```

Vraag 4: Object-Oriented Programming (2 punten)

De square_distance van punt $point = (x, y, z, ...)$ tot de oorsprong kan berekend worden met:

$$square\_distance = x^2 + y^2 + z^2 + ...$$

Het onderstaande programma heeft met functie "add_square_distance()" al een werkende Procedurele implementatie hiervan.

A. Voeg eigen code aan class Square_Distance toe zodat ook de Object-Oriented implementatie goed werkt.

B. Beschrijf in je eigen woorden wat "encapsulation" betekent.

C. Geef een voordeel van het gebruik van "encapsulation" in het onderstaande programma.

```
#include <iostream>

void add_square_distance(double coord, double& square_distance)
{
   square_distance += coord * coord;
}

class Square_Distance
{
/*** your code goes here ***/
};

int main()
{
   int n=3;
   double point[] = {4, 3, 5 /* ... */ };
   { // --------- Procedural implementation
      double square_distance = 0;
      for (int i=0; i<n; i++)
         add_square_distance( point[i], square_distance );
      std::cout<<"square_distance:"<< square_distance <<'\n'; // 50
   }
   { // --------- Object-Oriented implementation
      Square_Distance square_distance;
      for (int i=0; i<n; i++)
         square_distance.add( point[i] );
      std::cout<<"square_distance:"<< square_distance.get() <<'\n';
   }
}
```

Vraag 5 (2 punten)

A. Beschrijf in je eigen woorden waarom het handig is om op zo veel mogelijk plekken keyword "const" te gebruiken.

B. Beschrijf in je eigen woorden wat "automatic type conversion" is en geef een voorbeeld waarin "automatic type conversion" tot problemen kan leiden.

C. Beschrijf in je eigen woorden waarom het handig is om op zo veel mogelijk plekken type "std::ostream" te gebruiken in plaats van type "std::ofstream".

D. Beschrijf een toepassing (anders dan in het boek) van het gebruik van een "Multidimensional Array". Laat de toepassing ook zien in een kort voorbeeld-programma.

The End!