

Programmeren voor multimedia (Software)

- [2014 juni examen](#)
- [2015 juni examen](#)
- [2016 juni examen](#)
- [2020 juni examen](#)
- [algemeen](#)
- [2022 augustus examen](#)

2014 juni examen

Deel 1

Je krijgt de vraag om enkele classes te implementeren met hun header files. Deze classes hebben we al gezien tijdens het jaar (Point, Vector,...) Enkele operator overloading, assignment operator en hard copy dingen

Deel 2

1) Je krijgt een code en moet de eindwaardes van drie variabelen berekenen (pointers en shizzlewizzle)

2) Je krijgt enkele papieren code, haal alle fouten hieruit en verbeter deze (max 8 fouten)

2015 juni examen

Deel 1

Deel 1.1

Code:

```
int main() {
```

```
    IObject* s1 = new Square(Point(1,4),2.2);
    cout << s1->getPosition() << endl;
    cout << s1->getSize() << endl;
    Square s2 = new Square(Point(4,5),5);
    Square* s3 = new RotatedSquare(Point(5,5),8,60);
    cout << s2;
    cout << &s3;
```

```
}
```

uitvoer:

```
(1,4)
2.2
Position: (4,5) size: 5
Position: (5,5) size: 8 angle: 60
```

opdracht:

Schrijf de klassen IObject, Square en RotatedSquare.
Schrijf code zoals je zou verwachten van een professionele programmeur.
De klasse Point bestaat al. Je mag er vanuit gaan dat deze de << operator al ondersteunt.
Codeduplicatie is niet toegestaan.
IObject is een interfaceklasse.

Deel 1.2

Geef telkens één regel code voor elke vraag.

1) Alloceer met dynamisch geheugen de array d die 12 doubles bevat. 2) Gebruik een pointerimplementatie om de laatste waarde van deze array op 101 te zetten. 3) Verwijder nu deze dynamische array. 4) Maak een array b aan die 18 integers bevat. Dit zonder dynamische allocatie.

Deel 1.3

1. Tijdens het framework bij de drawmethode van boid zagen we dat de boid niet altijd juist geïntendeerd stond naargelang zijn richting. Leg uit hoe dit kwam en hoe dit kan opgelost worden.
2. Leg uit tijdens het laatste labo wat er gebeurde wanneer enkele boids van de flock over de westelijke grens van het scherm kwamen. Leg ook uit hoe dit komt.

Deel 2

Deel 1.1

Gegeven zijn de klassieke oefeningen tijdens de eerste 7 labo's. Er worden variabelen aangemaakt en (dubbele) pointers worden daarnaar toegewezen. Dan worden de pointers gemixt en waarden veranderd. Geef op het einde de uitvoer van elke variabele.

Deel 1.2

Gegeven is een stuk code met maximaal (maar niet noodzakelijk) 10 fouten in. Nummer elke regel waar een fout in zit en schrijf op een ander blad wat er wel op deze regel moet staan.

2016 juni examen

Vraag 1 en 2

Je kreeg code en moest zeggen wat de output was.

Vraag 3

Je krijgt 3 struct klassen en moest een functie schrijven in C++ zoals in java de "instanceof" methode bestaat.

Dit moest met "dynamic_casting".

Vraag 4

Teken de Huffman-boom

Vraag 5

Zet de Huffman-boom om naar bits.

Vraag 6

je kreeg een zin bestaande uit "abc" letters en moest het omzetten naar bits via het LZ77-compressiealgoritme.

Vraag 7 en 8

Je kreeg 2 keer een volledige "tekst" en je moest zeggen welk compressie-algoritme het beste gebruikt zou worden hiervoor. (LZ77, RLE of Huffman). Ook uitleg geven waarom.

Vraag 9

Je kreeg Java-code en moest dit omzetten naar C++ code.

Van wat ik me nog herinner van de opgave was:

```
interface Observer{  
    void update();  
}
```

```
public class PrintingObserver implements Observer{  
    @Override  
    public void update(){  
        system.out.println("Update detected.");  
    }  
}
```

```
public class Observable{  
    private ArrayList<Observer> observers = new ArrayList<Observer>();  
    public void addObserver(Observer observer){  
        observers.add(observer);  
    }  
    public void updateObservers(){  
        for(Observer observer : observers){  
            observer.update();  
        }  
    }  
}
```

```
public class App(){  
    public static void main(String[] args){  
        Observable o1 = new Observable();  
        Observable o2 = new Observable();  
        Observer o = new PrintingObserver();  
        o1.addObserver(o);  
        o2.addObserver(o);  
        o1.updateObservers();  
    }  
}
```

2020 juni examen

Het examen was een kleine test om te testen of je de taal C++ wel kon. Het was open-internet examen met een tijdslimiet van 30 min. Er waren geen vragen over het PE project 'Midi' zelf.

Voorbeelden

- What is the output of the code below?

```
#include <iostream>
#define P(x) std::cout << x;
struct Foo
{
    Foo() { P('a'); }
    Foo(const Foo&) { P('b'); }
    ~Foo() { P('c'); }
};
struct Bar : Foo
{
    Bar() { P('d'); }
    Bar(const Bar&) { P('e'); }
    ~Bar() { P('f'); }
};
void qux(Foo& foo) { }
int main()
{
    Bar* bar = new Bar;
    P('[');
    qux(*bar);
    P(']');
}
```

- What is the type of x?

```
T** a;
auto x = a[0];
```

- What is the value of y? Write your answer as 0x???????, i.e., use exactly 8 hexadecimal digits (bit manipulation)

```
uint32_t x = 0x12345678;  
uint32_t y = x & 0x12345678;
```

algemeen

Er wordt normaliter geen theorie gegeven. Alles is zelfstudie.

- Doorheen het semester zijn er een reeks kleine opdrachten waar 25% van de punten staan.
- Op het einde van het semester dient er een project ingeleverd te worden waar 25% van de punten op staan.
- Tijdens de examenperiode is er een schriftelijk openboek examen dat meetelt voor 50% van de punten.

2022 augustus examen

Het Examen

Het examen was een quiz om te testen of je de taal C++ wel kon. Er was geen internet toegang maar je mocht gebruiken wat er op je eigen laptop stond met een tijdslimiet van 30 min.

Er waren geen vragen over het PE project zelf.

De vragen waren zo goed als dezelfde als de oefen quiz die je kon doen op x.toledo.

Veel succes!!