

# Academie voor Technology, Innovation & Society Delft

KLAS(SEN) : EQ	BLAD : 1 van 3 BLADEN
OND. DEEL : Object Georiënteerd Programmeren in C++	DOCENT : Harry Broeders
AFK. O.E. : <b>OGOPRG</b>	DATUM : -
KWART. : 3	TYPE : <b>proeftentamen</b>
	TIJD : -

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

Bij elke opgave staat tussen haakjes het maximale aantal te behalen punten vermeld.

Eindcijfer = (aantal behaalde punten + 10) / 10.

**Let op!** Dit **proeftentamen** gaat alleen over de stof die in de **eerste 11 lessen** van OGOPRG behandeld is! Het **echte tentamen** gaat **ook over de stof** die in **les 12 t/m 21** behandeld zal worden!

**1.** Gegeven zijn de volgende C++ classes:

```
class TemperatuurSensor {
public:
    int geefTemperatuur() const; // geef temperatuur in graden Celsius
    // verdere code niet gegeven.
};

class TemperatuurActuator {
public:
    virtual void zetAan() = 0;
    virtual void zetUit() = 0;
    // verdere code niet gegeven.
};

class VerwarmingsElement: public TemperatuurActuator {
    // code niet gegeven.
};

class KoelElement: public TemperatuurActuator {
    // code niet gegeven.
};
```

Deze classes worden gebruikt in de het volgende programma:

```
void schakelUit(TemperatuurActuator& t) {
    // schakelt de als argument meegegeven TemperatuurActuator uit.
    // code niet gegeven.
}

void regel(const TemperatuurSensor& s, VerwarmingsElement& v,
           KoelElement& k) {
    // code niet gegeven.
}

int main() {
    TemperatuurSensor sensor;
    VerwarmingsElement verwarming;
    KoelElement koeling;
    schakelUit(verwarming);
    schakelUit(koeling);
    while (true) {
        regel(sensor, verwarming, koeling);
        sleep(5); // wacht 5 seconden.
    }
    return 0;
}
```

**Zie volgende blad** ⇨

# Academie voor Technology, Innovation & Society Delft

---

KLAS(SEN) : EQ	BLAD : 2 van 3 BLADEN
OND. DEEL : Object Georiënteerd Programmeren in C++	DOCENT : Harry Broeders
AFK. O.E. : <b>OGOPRG</b>	DATUM : -
KWART. : 3	TYPE : <b>proeftentamen</b>
	TIJD : -

---

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

---

Dit programma wordt gebruikt om de temperatuur in een bierkelder te regelen. De temperatuur in de bierkelder moet tussen de 5 en 10 graden Celsius zijn.

- A. (5) Geef de implementatie van de functie `schakelUit`.
- B. (10) De functie `schakelUit` heeft een `TemperatuurActuator&` als parameter. De functie wordt echter aangeroepen met een `VerwarmingElement` (of `KoelElement`) als argument. Levert dit een compile error op? **Verklaar je antwoord! Hoe wordt dit verschijnsel genoemd?**
- C. (5) Als in de functie `schakelUit` de `TemperatuurActuator&` parameter wordt vervangen door een `TemperatuurActuator` parameter dan geeft de compiler een foutmelding op de regel waarop deze parameter is gedefinieerd. **Verklaar dit.**
- D. (10) Geef de implementatie van de functie `regel` die er voor moet zorgen dat de temperatuur in een bierkelder tussen de 5 en 10 graden Celsius blijft. De verwarming moet worden aangeschakeld als de temperatuur  $< 5$  graden is en de koeling moet worden aangeschakeld als de temperatuur  $> 10$  graden is. Natuurlijk moet de verwarming worden uitgeschakeld als de temperatuur  $\geq 5$  graden is en moet de koeling worden uitgeschakeld als de temperatuur  $\leq 10$  graden is.
- E. (5) In de functie `regel` is de eerste parameter een `const` reference en de overige twee parameters zijn “gewone” references. Verklaar waarom de laatste twee parameter **niet** `const` zijn.
- F. (5) De code van de class `VerwarmingsElement` is niet gegeven. Welke memberfuncties **moeten** in deze class gedefinieerd zijn, om het programma te kunnen compileren?

2. Om de class `TemperatuurSensor` (uit opgave 1) voor de applicatieprogrammeur eenvoudiger te maken is deze class als volgt uitgebreid:

```
class TemperatuurSensor {
public:
    int geefTemperatuur() const; // geef temperatuur in graden Celsius
    bool operator==(int i) const;
    bool operator<(int i) const;
    bool operator>(int i) const;
// verdere code niet gegeven.
};
```

**Zie volgende blad ⇨**

# Academie voor Technology, Innovation & Society Delft

---

KLAS(SEN) : EQ	BLAD : 3 van 3 BLADEN
OND. DEEL : Object Georiënteerd Programmeren in C++	DOCENT : Harry Broeders
AFK. O.E. : <b>OGOPRG</b>	DATUM : -
KWART. : 3	TYPE : <b>proeftentamen</b>
	TIJD : -

---

Tijdens dit tentamen mogen **alle** boeken, dictaten, aantekeningen enz. worden gebruikt.

---

Een applicatieprogrammeur kan nu op de volgende manier kijken of de temperatuur 30 graden is:

```
TemperatuurSensor temperatuur;  
if (temperatuur == 30) {  
    // verdere code niet gegeven.  
}
```

- A. (5) Achter de declaratie van de operator== staat het woordje const. Welke **garantie** geeft dit in het bovenstaande stukje code?
- B. (10) Geef de implementatie van deze operator==.
- C. (5) Er treedt een fout op bij het vertalen van het programma als in het bovenstaande stukje code de voorwaarde van de if als volgt wordt geschreven:  
if (30 == temperatuur) {  
**Verklaar dit!**
- D. (10) Geef alle code aanpassingen/toevoegingen die nodig zijn om de volgende code foutloos te laten werken.  
if (30 == temperatuur) {
- E. (10) Geef aan wat er aan het programma moet worden **toegevoegd** om de volgende applicatiecode foutloos te laten werken:  
cout << "T = " << temperatuur << " graden Celsius";

**3.** Gegeven is de volgende C++ class:

```
class Stack {  
public:  
    int top() const; // geef bovenste element van de stack  
    void pop(); // verwijder bovenste element van de stack  
    void push(int e); // zet het element e bovenop de stack  
    int size() const; // geef aantal elementen op de stack  
private:  
    // verdere code niet gegeven.  
};
```

- (10) Als we de bovenstaande class willen gebruiken kan dit alleen als we een stack van het type int nodig hebben. Pas de bovenstaande class definitie zodanig aan dat we stacks van een **willekeurig** type T kunnen maken. Zorg er daarbij voor dat de implementatie ook efficiënt werkt als de objecten van type T **veel** geheugenruimte gebruiken.

**Succes!**