1. De aanroep **show (x)** toont **x** op het beeldscherm, gevolgd door een spatie. Wat wordt getoond door het volgende code fragment:

**for (int index = 2; index < 16; index \*= 2) {**

**show (index);**

**}**

a. 2 4 6 8 10 12 14

b. 2 4 8

c. 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2

d. 2 3 4 5 6 7 8 9 10 11 12 13 14 15

2. Function **askQuestion ()** stelt een vraag en function **answerIsOk ()** geeft een boolean waarde terug die aangeeft of het antwoord op de laatst gestelde vraag goed is. Welk code fragment leidt er toe dat de vraag gesteld wordt tot 't antwoord goed is:

a.

**do {**

**askQuestion ();**

**} while (!answerIsOk ());**

b.

**do {**

**askQuestion ();**

**} while (answerIsOk ());**

c.

**while (!answerIsOk) {**

**askQuestion ();**

**}**

d.

**while (anwerIsOk) {**

**askQuestion ();**

**}**

3. Wat is de waarde van **x** na het volgende code fragment:

**int square (int aNumber) {**

**aNumber = aNumber \* aNumber;**

**return aNumber;**

**}**

**int x = 10;**

**square (x);**

**x = square (x);**

a. 100

b. 10

c. 0

d. 10000

4. Wat is er mis met de volgende regel code:

**extern int Int = 3;**

a. De naam **Int** mag niet, omdat het een keyword is

b. Deze statement mag **Int** geen waarde geven omdat het geen definitie van **Int** is maar alleen een declaratie

c. Het woord **extern** mag alleen in een header file voorkomen

d. Het moet zijn **EXTERN**, omdat dit een macro is

5. De aanroep **show (x)** toont **x** op het beeldscherm, gevolgd door een spatie. Wat is er mis met het volgende code fragment:

**switch (aFloat) {**

**case 3.141532653588: {**

**show ("Dit is het getal pi");**

**break;**

**}**

**case 2.718281828459: {**

**show ("Dit is het getal e");**

**break;**

**}**

**default: {**

**show ("Dit is zo maar een getal");**

**break;**

**}**

**}**

a. Case labels mogen geen floats zijn. Dit zou ook niet goed kunnen, immers door afrondingsverschillen is het checken van exacte gelijkheid van floating point getallen meestal zinloos.

b. De eerste regel moet zijn **switch (int float) {**

c. Je mag in de cases van een switch geen **{}** gebruiken

d. De **break** statement mag alleen worden gebruikt om uit een lus te springen

6. Als je in plaats van een globale variabele het adres van die globale variabele aan een functie doorgeeft:

a. Kun je die globale variabele in de functie een ander adres geven

b. Kun je die globale variabele in de functie een ander waarde geven

c. Werk je in de functie met een lokale kopie van die globale variabele en kun je hem dus niet veranderen

d. Moet je niet vergeten **delete** aan te roepen, anders heb je een geheugenlek

7. Wat is er mis met de volgende code:

**struct EmployeeRecord {**

**char firstName [64];**

**char lastName [64];**

**char employeeNumber [16];**

**};**

**EmployeeRecord employeeRecord;**

a. Niets

b. Het attribuut employeeNumber moet een **long** of een **int** zijn

c. Het moet geen **char** maar **string** zijn

d. Het moet geen **struct** maar **class** zijn

8. Wat is het effect van het volgende code fragment:

**class Vector {**

**public:**

**float x, y, z;**

**Vector add (Vector v1, Vector v2);**

**};**

a. Er wordt een object type gedefinieerd, maar nog geen objecten van dat type

b. Er wordt een object gecreëerd

c. Er worden twee vectoren opgeteld

d. Er wordt geheugenruimte gereserveerd voor drie floating point getallen

9. Class **House** en class **School** erven beide van class **Building**. Welk van de volgende statements is correct:

a. **Building \*building = new School;**

b. **School \*School = new Building;**

c. **House \*house= new School;**

d. **School \*school = &new School;**

10. Wat is in de eerste plaats het nut van stabiele dunne interfaces in broncode:

a. Executie van de code wordt er sneller van

b. Veranderde lokale ontwerpbeslissingen hebben minder impact op de code als geheel

c. Dit spaart geheugenruimte

d. Software is gemakkelijker te beveiligen