

## Lösung Klausur 3, infT

### Fragen zur Sprache C++

Korrekte Multiple-Choice-Antworten sind **fett und kursiv** markiert. Lücken sind mit **fettem Text** ergänzt. Code ist an entsprechender Stelle in *Courier* **fett** angegeben.

1. Wird die Bibliothek für Streams benötigt, ist es egal, ob `<iostream>` oder `<iostream.h>` verwendet wird. Beides bedeutet genau dasselbe.
  - a) richtig
  - b) falsch**
  
2. **a) Eine Funktionsdeklaration kann auch als Funktionsprototyp bezeichnet werden.**
  - b) Ein Funktionsprototyp wird auch als Funktionsdefinition bezeichnet.
  - c) Funktionsprototypen dürfen grundsätzlich nur in Header-Files stehen.
  - d) Wenn eine globale Funktion nirgends in der Form *rückgabewert funktion(argumentenliste)*; vorliegt, so meldet der Compiler einen Fehler.
  
3. Jedes C++-Programm endet nach Ausführung der letzten Anweisungszeile in `main`.
  - a) richtig
  - b) falsch**
  
4. Beim Erzeugen eines Objekts gilt grundsätzlich, dass
  - a) Speicher für die Datenelemente reserviert wird,**
  - b) jedes Element mit einem passenden Wert initialisiert wird,
  - c) Speicher auf dem Heap alloziert wird.
  
5. **a) C++ ist eine hybride Sprache,**
  - b) weil sie auf unterschiedlichen Plattformen läuft,
  - c) weil sie sowohl Strukturen wie auch Klassen kennt.
  
6. Gegeben ist Klasse `class Xy { double d; };`  
 Notieren Sie auf einfachste Art, wie ein Array mit Xy-Elementen global angelegt und mit 100 Elementen initialisiert werden kann.
 

```
Xy arr[100] = { Xy() };
```
  
7. Eine Funktion sei mit Default-Argumenten definiert und bei Aufruf werde ein Argument weggelassen. In diesem Fall müssen
  - a) auch die Argumente davor weggelassen werden.
  - b) auch nachfolgende Argumente weggelassen werden.**
  - c) nie weitere Argumente weggelassen werden.
  
8. Gegeben ist Klasse `class D { const double d; public: D(double d); };`
  - a) Implementieren Sie den Konstruktor, der das Member initialisiert.
 

```
D::D(double d) : d(d) { }
```
  - b) Notieren Sie auf einfachste Art, wie ein Array mit D-Elementen global angelegt und mit 4 Elementen initialisiert werden kann.
 

```
D arr[] = { D(1), D(2), D(3), D(4) };
```

9. In C++ ist jeder Vergleich ein Ausdruck vom Typ `int` mit dem Wert 0 ("false") oder 1 ("true").
- richtig
  - falsch**

10. Die bezeichneten Zeilen des nachfolgenden Codes sind korrekt:

```
int main() {
    int t = 12;
    a) void *pt = &t;
    b) *((int *)pt) = 31;
}
```

11. Die bezeichneten Zeilen des nachfolgenden Codes sind korrekt:

```
int main() {
    int t = 12;
    void *pt;
    a) void **ppt = pt;
    b) int& r = (int)*pt;
}
```

12. Die bezeichneten Zeilen des nachfolgenden Codes sind korrekt:

- `int a, b = a = 5;`
- `int *p, q;`
- `p = &a;`
- `q = &b;`

13. Füllen Sie die Lücken (oder nicht) im Prototypen `replaceChars` nach den Regeln der Kunst. Zur Funktion: sie ersetzt in einem String einen Teilstring durch einen andern Teilstring.

```
void replaceChars(string& modifyMe,
                 const string& findMe,
                 const string& newChars);
```

14. Gegeben ist der Code

```
// ... noetige Bibliotheksdatei einbinden
int main() {
    string s1;
    string *s2 = new string("Hallo, Welt!");
    string s3("Jolly good");
    // ...
}
```

Wie viele Male muss der Compiler für obiges Programm Konstruktor-Code einfügen?

**3** Mal

15. Folgendes ist gegeben:

```
void func(float *p1, float *p2); float x = 1f, y = 2.2;
```

Die Funktion `func()` kann dann wie folgt aufgerufen werden:

- `func(x, y);`
- `func(&x, &y);`
- `func(*x, *y);`

16. Die beiden Zeilen führen stets zum selben Resultat, d.h. können ausgetauscht werden:

```
if (a-- == ++b) || c == 55)
if (c == 55 || a-- == ++b))
a) richtig
b) falsch
```

17. Gegeben:

```
char *p = "Non timeo";
char q[] = "Non timeo";
class R { static const char s[] = "Non timeo"; };
a) sizeof(p) == sizeof(q) liefert true.
b) sizeof(R) == sizeof(q) liefert true.
```

18. Gegeben ist die Klasse History:

```
class History {
    ... p;
public:
    // ...
    ~History();
};
```

Das Member `p` werde benützt, um auf ein Array von `strings` zu zeigen. Welchen Typ muss `p` aufweisen?

```
string *
```

19. Gegeben ist die Klasse History aus Aufgabe 18. Schreiben Sie den Code für den Destruktor.

```
History::~History() { delete[] p; }
```

20. `Q` sei eine Klasse. Um ein Array von Zeigern auf Objekte vom Typ `Q` definieren zu können, muss die Klasse einen Default-Konstruktor besitzen.

- a) richtig
- b) falsch**

21. Gegeben sei der folgende Prototyp `const int& func(void);`.

Dann wird bei der Anweisung `func() = 7;`

- a) der Compiler eine Fehlermeldung ausgeben.**
- b) der Funktion ein neuer Return-Wert zugewiesen.
- c) dem durch den Funktionsaufruf referenzierten Objekt ein neuer Wert zugewiesen.

22. Der von lokalen, nicht als `static` deklarierten Objekten belegte Speicher wird freigegeben, sobald der entsprechende **Block** verlassen wird.

23. Eine Funktion `strlChr()` wird entworfen, die das letzte Auftreten eines Zeichens in einem C-String untersucht und einen Zeiger auf das gefundene Zeichen zurückgibt. Dann kann die Funktion folgenden Prototyp aufweisen:

- a) `char *strlChr(const char *s, int c);`
- b) `char *strlChr(char s[], const int c);`**
- c) `char * const strlChr(const char *s, int c);`

24. Gegeben sei folgender Ausschnitt aus einer Klassendefinition:

```
class DoubleArr {
    double *ptr;
    // ...
public:
    DoubleArr(int n); // Erzeuge ein Array der Laenge n
    // ...
};
```

In der Definition des Konstruktors wird dann mit folgender Anweisung der erforderliche Speicher reserviert:

```
ptr = new double[n];
```

25. Eine Funktion definiere eine Referenz auf ein Objekt `obj` vom Typ `Obj`, welches innerhalb dieser Funktion erzeugt wurde. Liefert die Funktion die Referenz als Rückgabewert, wie muss dann die Deklaration von `obj` in der Funktion lauten?

```
static Obj obj;
```

26. Für die Member-Daten eines Objekts der Klasse `Foo` (mit Konstruktor `Foo()`) wird Speicher angelegt, wenn
- die Klasse definiert wird.
  - der Aufruf `Foo *myY = new Foo;` zur Ausführung kommt.
  - die Zeile `Foo myFoo;` ausgeführt wird.
27. Wird in C++ eine Memberfunktion aufgerufen, so wird ein "verstecktes" Argument mit der Adresse des aktuellen Objekts übergeben. Wie heisst dieses?

```
this
```

28. Gegeben sei die folgende Funktion: `void troja(int& a) { a *= 4; }.`

Dann wird beim Aufruf `troja(2);`

- der Compiler eine Fehlermeldung ausgeben.
  - die Zahl 4 verdoppelt.
  - die Zahl 2 vervierfacht.
29. Gegeben sei eine zwei-dimensionale Matrix `matrix` mit Elementen vom Typ `float`. `matrix` weist drei Zeilen und vier Spalten auf. Notieren Sie den Code, um dem letzten Element in der letzten Zeile den Wert `-7.2` zuzuweisen:

```
matrix[2][3] = -7.2;
```

30. Gegeben: `long z; long& rz = z;.`

Dann wird im Ausdruck `++rz;` der Wert von `z` inkrementiert.

- richtig
  - falsch
31. Angenommen, der Operator `new` reserviert dynamisch den Speicher für ein Objekt einer Klasse. Dann sorgt er auch dafür, dass
- alle von ihm belegten Bytes mit 0 initialisiert werden.
  - ein passender Konstruktor für das Objekt aufgerufen wird.
  - der New-Handler aufgerufen wird, wenn kein Speicherplatz verfügbar ist.
32. Mit der Definition `double *ptr = new float;`
- wird eine Float auf dem Heap erzeugt.
  - wird eine Float erzeugt, implizit in Double konvertiert und auf dem Heap abgelegt.
  - wird vom Compiler eine Fehlermeldung ausgegeben.
33. C++ bietet die Möglichkeit, Konstruktoren, Memberfunktionen und Destruktoren zu überladen.
- richtig
  - falsch

34. Die Funktion `frago()` sei im Namensraum `metro` definiert und wird ausserhalb des Namensraums aufgerufen. Ohne Verwendung einer `using`-Deklaration oder `using`-Direktive wird beim Aufruf dem Namen der Funktion

```
metro::  
vorangestellt.
```

35. Die Klasse `A` gewähre der Klasse `B` direkten Zugriff auf ihre privaten Daten. Formulieren Sie dies codemässig.

```
class B; class A { friend class B; }; class B { /* ... */};
```