**5. Aufgabenblatt**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vorname** | **Nachname** | **Matrikelnummer** | | **Fachrichtung** |
| Philipp | Huber | 63326 | | Mabb |
| Marius | Grumer | 63284 | | Mabb |
| **Punkte** | | | xx/yy | | |
| **Bestanden** | | | ja/nein | | |

*1. Aufgabe:*

Der Name eines Konstruktors ist identisch mit dem Namen .........................

**Lösung**: Der Klasse

***Erläuterung****:* Der Name des Ctors muss mit dem Namen der Klasse gleich sein

*2. Aufgabe:*

Wenn bei der Definition eines Objektes der Compiler den Konstruktor mit der passenden Signatur nicht finden kann, wird

1. ein Objekt angelegt und mit Default-Werten initialisiert
2. ein Objekt angelegt und nicht initialisiert
3. kein Objekt angelegt und eine Fehlermeldung ausgegeben

**Lösung**: c)

***Erläuterung****:* Sobald der Compiler die passende Signatur nicht finden kann, weil die Parameter welche angegeben sind falsch sind, kann kein Objekt angelegt werden.

Werden keine Parameter übergeben, wird stattdessen der Standard Ctor aufgerufen

*3. Aufgabe:*

Damit eine Methode für ein konstantes Objekt aufrufbar ist, muss die Methode als ..................... deklariert werden.

**Lösung: als const ….**

*4. Aufgabe:*

Angenommen, eine Klasse enthält eine konstante und eine nicht-konstante Version einer Methode. Dann ruft der Compiler die const Version automatisch für konstante Objekte auf.

***a.*** wahr ***b.*** falsch

**Lösung**: a

***Erläuterung***: In diesem Fall ist Methode überladen. Der Compiler sucht sich die passende Version aus.

*5. Aufgabe:*

Angenommen, ein Objekt wird mit einem anderen Objekt desselben Typs initialisiert. Dann wird folgende Methode bzw. folgender Operator aufgerufen:

1. Zuweisungsoperator
2. Default-Konstruktor
3. Kopier-Konstruktor

**Lösung**: c)

**Erläuterung***:* Laut Skript wird bei der Initialisierung mit folgender Syntax der copyctor aufgerufen: Klassenname objekt2 = objekt1;

*6. Aufgabe:*

Angenommen, die Klasse Measure enthält folgende public Methode:

**bool isGreater(Measure m) const;**

Beim Aufruf der Methode isGreater()

1. verweist der Parameter m auf das übergebene Argument
2. erzeugt der Default-Konstruktor der Klasse Measure das Objekt m, dem dann das Argument zugewiesen wird
3. erzeugt der Kopier-Konstruktor das Objekt m und initialisiert es mit dem übergebenen Argument

**Lösung*: c)***

**Erläuterung: es handelt sich um ein call by value, daher wird der cpyctor aufgerufen**

*7. Aufgabe:*

Angenommen, eine Klasse Test besitzt folgende public Methode:

**void swap(Test& res);**

Die Methode swap() kann dann nur auf die privaten Elemente des aktuellen Objektes, nicht auf die des Argumentes zugreifen.

***a.*** wahr ***b.*** falsch

**Lösung**: a

**Erläuterung***:* eine Methode kann immer nur auf die privaten Elemente des eigenen Objektes zugreifen. Auf private Elemente von übergebenen Klassen kann nicht direkt zugegriffen werden (nur via setter und getter, bzw. den passenden Methoden).

*8. Aufgabe:*

Was ist bei folgender Klassendefinition falsch?

**class Gift**

**{**

**string gift;**

**public:**

**...**

**~Gift();**

**~Gift(string&);**

**};**

**Lösung**:

**class Gift**

**{**

**Private:**

**string gift;**

**public:**

**...**

**~Gift();**

**~~~Gift(string& xyz);~~ //dtor darf nicht überladen werden**

**};**

*9. Aufgabe:*

Bestimmen und korrigieren Sie mögliche Fehler in den Definitionen folgender Zugriffsmethoden:

**class Numeric**

**{**

**private:**

**long x;**

**public:**

**...**

**void set(long n) const { x = n; }**

**long get() { return x; } const**

**};**

**Lösung**

**class Numeric**

**{**

**private:**

**long x;**

**public:**

**...**

**void set(long n) { x = n; }**

**long get() { return x; }**

**};**

//ohne const, da x änderbar ist.