

# Praktikum Objektorientierte Programmierung in C++ (WS 2021/2022)

[Dashboard](#) / [My courses](#) / [Wintersemester 2021/2022](#) / [Ingenieurwissenschaften](#) / [Informatik und Angewandte Kognitionswissenschaften](#)  
/ [Praktikum OOP in C++ WS 2021/2022](#) / [Aufgabe 3/Task 3](#)  
/ [A3 Teil 1: Hausaufgabe zur Vorbereitung auf die Präsenz-Gruppe/Part 1: Homework Task for Preparation of the Presence Group](#)

## A3 Teil 1: Hausaufgabe zur Vorbereitung auf die Präsenz-Gruppe/Part 1: Homework Task for Preparation of the Presence Group

In dieser Aufgabe sollen überladene Operatoren am Beispiel Punkte und Kreise in einem kartesischen 2D-Koordinatensystem definiert werden. Programmieren Sie im einzelnen:/

In this task, overloaded operators are to be defined for points and circles in a Cartesian 2D coordinate system as example. Program in detail:

- Definieren Sie eine C++-Struktur (also ohne **typedef**) mit Namen **point2D** mit zwei Gleitpunktzahlen für die x- und die y-Koordinate eines Punktes./  
Define a C++ structure (i.e. without **typedef**) called **point2D** with two floating point numbers for the x- and y-coordinates of a point.
- Definieren Sie einen überladenen Eingabe-Operator **>>** für eine solche Strukturvariable vom Typ **point2D**. Lesen Sie im Rumpf nach einer Eingabeaufforderung zwei Gleitpunktzahlen für die x- und die y-Koordinate eines Punktes von der als Parameter übergebenen Zeichen-Eingabestrom-Referenz ein (siehe Beispiele unten)./  
Define an overloaded input operator **>>** for such a structure variable of type **point2D**. In the body, after a user guidance read two floating point values for the x- and y-coordinates of a point using the character input stream reference which was passed as a parameter (see examples below).
- Definieren Sie einen überladenen Ausgabe-Operator **<<** für eine solche Strukturvariable vom Typ **point2D**. Schreiben Sie die Punktkoordinaten in runden Klammern und mit einem Komma dazwischen auf die als Parameter übergebene Zeichen-Eingabestrom-Referenz (siehe Beispiele unten)./  
Define an overloaded output operator **<<** for such a structure variable of type **point2D**. Write the point coordinates in round brackets and with a comma in between onto the character input stream reference passed as parameter (see examples below).
- Definieren Sie einen überladenen Additions-Operator **+** mit zwei Referenzen auf Punkte vom Typ **point2D** als Operanden. Der Operator soll einen Punkt zurück geben, der in der x- und y-Koordinate jeweils die der beiden übergebenen Punkte addiert hat./  
Define an overloaded addition operator **+** with two references to points of type **point2D** as operands. The operator is to return a point that has the x- and y-coordinates each added of the two points passed.
- Definieren Sie eine weitere C++-Struktur (also ohne **typedef**) mit Namen **circle** (Kreis) mit einem Punkt vom obigen Typ **point2D** als Kreismittelpunkt und einem Radius vom Typ Gleitpunktzahl als Komponenten./  
Define a further C++ structure (i.e. without **typedef**) called **circle** with a point of the above type **point2D** which represents the center of a circle and floating point number which represents the radius as two components.
- Definieren Sie einen überladenen Eingabe-Operator **>>** für eine solche Strukturvariable vom Typ **circle**. Lesen Sie im Rumpf nach einer Eingabeaufforderung den Kreismittelpunkt über den oben definierten Eingabe-Operator für einen Punkt vom Typ **point2d** und eine Gleitpunktzahl für den Radius von der als Parameter übergebenen Zeichen-Eingabestrom-Referenz ein. Überprüfen Sie, dass der eingelesene Radius nicht negativ ist, ansonsten schreiben Sie eine Meldung auf den Standard-Zeichen-Fehler-Ausgabestrom und drehen das Vorzeichen um (siehe rot markierte Zeilen im Beispiel unten)./  
Define an overloaded input operator **>>** for such a structure variable of the type **circle**. In the body, after a prompt, read in the circle center via the input operator defined above for a point of type **point2d** and a floating point number for the radius from the character input stream reference passed as a parameter. Check that the radius read in is not negative, otherwise write a message onto the standard character error output stream and reverse the sign (see read marked lines in the example below).
- Definieren Sie einen überladenen Ausgabe-Operator **<<** für eine solche Strukturvariable vom Typ **circle**. Schreiben Sie die Zeichenkette "**center=**" auf die als Parameter übergebene Zeichen-Ausgabestrom-Referenz, dahinter den Kreismittelpunkt über den oben definierten Ausgabe-Operator, danach "**, radius=**" sowie den Radius selbst und ein Zeilenende (siehe Beispiele unten)./  
Define an overloaded output operator **<<** for such a structure variable of type **circle**. Write the string "**center=**" onto the character output stream reference which was passed as a parameter, behind it the centre of the circle via the output operator defined above for points, then "**, radius=**" followed by the the radius itself, and an end of the line at the end (see examples below).
- Definieren Sie einen überladenen Additions-Operator **+** mit einer Referenz auf einen Punkt vom Typ **point2D** und einer Referenz auf einen Kreis vom Typ **circle** als Operanden. Der Operator soll einen Kreis zurück geben, dessen Kreismittelpunkt aus der Addition des übergebenen Punktes mit dem Kreismittelpunkt berechnet ist - verwenden Sie den obigen Additions-Operator für Punkte - und dessen Radius aus dem übergebenen Kreis kopiert (siehe Beispiele unten)./  
Define an overloaded addition operator **+** with a reference to a point of type **point2D** and a reference to a **circle** as operands. The

operator is to return a circle whose centre is calculated from the addition of the passed point with the centre of the circle - call above addition operator for points - and whose radius is copied from the given circle. (see examples below).

9. Definieren Sie einen überladenen Additions-Operator `+` mit einer Referenz auf einen Kreis vom Typ `circle` und einer Gleitpunktzahl als Operanden. Der Operator soll die übergebene Gleitpunktzahl zum Radius im übergebenen Kreis addieren und die Referenz auf den Kreis wieder zurück geben./

Define an overloaded addition operator `+` with a reference to a `circle` and a floating point number as operands. The operator shall add the passed floating point value to the radius in the passed circle and return the passed reference to the circle.

10. Definieren Sie einen überladenen Vergleichs-Operator `==` mit zwei Referenzen auf Punkte vom Typ `point2D` als Operanden. Der Operator soll die beiden Punkte auf Gleichheit testen - gleiche Werte für die x- und y-Koordinaten - und einen Booleschen Wert zurück geben./

Define an overloaded comparison operator `==` with two references to `point2Ds` as operands. The operator shall compare the two points for equality - same values for the x- and y-coordinates - and return a Boolean value.

11. Definieren Sie einen überladenen Vergleichs-Operator `!=` mit zwei Referenzen auf Punkte vom Typ `point2D` als Operanden. Der Operator soll die beiden Punkte auf Ungleichheit testen - ungleiche Werte für die x- oder die y-Koordinaten - und einen Booleschen Wert zurück geben./

Define an overloaded comparison operator `!=` with two references to `point2d` as operands. The operator shall compare the two circles for inequality - unequal values for the x- or y-coordinates - and return a Boolean value.

12. Definieren Sie einen überladenen Vergleichs-Operator `==` mit zwei Referenzen auf Kreise vom Typ `circle` als Operanden. Der Operator soll die beiden Kreise auf Gleichheit testen - gleiche Mittelpunkte überprüft über Ihren oben definierten Vergleichs-Operator `==` für Punkte und gleicher Wert für den Radius - und einen Booleschen Wert zurück geben./

Define an overloaded comparison operator `==` with two references to `circles` as operands. The operator shall compare the two circles for equality - same center is to be checked using the above defined comparison operator `==` for points and the same value for the radius - and return a Boolean value.

13. Definieren Sie einen überladenen Vergleichs-Operator `!=` mit zwei Referenzen auf Kreise vom Typ `circle` als Operanden. Der Operator soll die beiden Kreise auf Ungleichheit testen - ungleiche Mittelpunkte oder ungleicher Wert für den Radius - und einen Booleschen Wert zurück geben./

Define an overloaded comparison operator `!=` with two references to `circle` as operands. The operator shall compare the two circles for inequality - unequal centers or unequal values for the radius - and return a Boolean value.

14. Schreiben Sie eine `main`-Funktion mit folgenden Definitionen und Anweisungen:/

Write a `main` function with following definitions and statements:

- definieren Sie eine Variable `p` vom Typ `point2D` und initialisieren diese mit den Koordinaten `(1.0, 1.0)` ./  
define a variable `p` of type `point2D` and initialise it with the coordinates `(1.0, 1.0)` .
- definieren Sie ein Feld `c` vom Typ `circle` für 6 Kreise und initialisieren den ersten Kreis in diesem Feld als Einheitskreis mit Mittelpunkt im Ursprung des Koordinatensystem `(0.0, 0.0)` und mit Radius `1.0` ./  
define an array `c` of type `circle` for 6 circles and initialise the first circle in this array as a unit circle with center in the origin of the coordinate system `(0.0, 0.0)` and with radius `1.0` .
- lesen Sie den zweiten und dritten Kreis in diesem Feld über Ihren oben definierten Eingabeoperator `>>` für Kreise vom Standard-Zeichen-Eingabestrom ein./  
read the second and third circles in this array via your input operator `>>` defined above for circles from the standard character input stream.
- geben Sie die ersten drei Kreise im Feld über Ihren oben definierten Ausgabe-Operator `<<` für Kreise auf dem Standard-Zeichen-Ausgabestrom aus./  
output the first three circles in the array via your output operator `<<` defined above for circles on the standard character output stream.
- addieren Sie über Ihren oben definierten Additions-Operator `+` jeweils den obigen Punkt `p` zu den ersten drei Kreisen im Feld und speichern die zurück gegebenen Kreise im vierten bis sechsten Element im Feld./  
add the above point `p` to each of the first three circles in the array via your addition operator `+` defined above and store the resulting circles in the 4th to 6th elements of the array.
- schreiben Sie alle sechs Kreise im Feld innerhalb einer `for`-Schleife über Ihren oben definierten Ausgabe-Operator `<<` für Kreise auf den Standard-Zeichen-Ausgabestrom./  
write all six circles in the array inside a `for` loop via your output operator `<<` defined above for circles onto the standard character output stream.
- lesen Sie eine weitere Gleitpunktzahl-Variable vom Standard-Zeichen-Eingabestrom ein, addieren innerhalb einer `for`-Schleife über Ihren oben definierten Additions-Operator `+` diese Gleitpunktzahl jeweils zu den letzten drei Kreisen im Feld und schreiben diese dann über Ihren Ausgabe-Operator `<<` für Kreise auf den Standard-Zeichen-Ausgabestrom (siehe Beispiel unten)./  
read a further floating-point number variable from the standard character input stream, add this floating-point number to each of the last three circles in the array inside a for loop using your addition operator `+` defined above, and then write it onto the standard character output stream using your output operator `<<` for circles (see example below).
- vergleichen Sie die ersten beiden Kreise im Feld über Ihren obigen Vergleichs-Operator `==` und schreiben eine Zeichenkette wie im Beispiel unten auf den Standard-Zeichen-Ausgabestrom, ob diese gleich sind oder nicht./  
compare the first two circles in the array via your comparison operator `==` defined above and write a string onto the standard character output stream like in the example below whether they are the same or not.

- vergleichen Sie den zweiten und dritten Kreis im Feld über Ihren obigen Vergleichs-Operator **!=** und schreiben eine Zeichenkette wie im Beispiel unten auf den Standard-Zeichen-Ausgabestrom, ob diese ungleich sind oder nicht.  
**compare the second and third circle in the array via your above comparison operator != and write a string onto the standard character output stream like in the example below whether they are unequal or not.**

### Beispiel Programmlauf 1/Example Program Run 1

```
input center:
please input x and y coordinate: 1.1 3.5
please input radius: 2.7
input center:
please input x and y coordinate: -4.5 -2.2
please input radius: 3.1
c[0]: center=(0,0), radius=1
c[1]: center=(1.1,3.5), radius=2.7
c[2]: center=(-4.5,-2.2), radius=3.1
by adding point (1,1) to c[0] to c[2] assign the results to c[3] to c[5]
c[0]: center=(0,0), radius=1
c[1]: center=(1.1,3.5), radius=2.7
c[2]: center=(-4.5,-2.2), radius=3.1
c[3]: center=(1,1), radius=1
c[4]: center=(2.1,4.5), radius=2.7
c[5]: center=(-3.5,-1.2), radius=3.1
input value to get added to the radius of c[3] to c[5]: 1.2
c[3]: center=(1,1), radius=2.2
c[4]: center=(2.1,4.5), radius=3.9
c[5]: center=(-3.5,-1.2), radius=4.3
compare circles:
c[0] and c[1] are not equal
c[1] and c[2] are unequal
```

### Beispiel Programmlauf 2/Example Program Run 2

```
input center:
please input x and y coordinate: 1 2
please input radius: -3
error in operator>>: negative radius, sign changed
input center:
please input x and y coordinate: -4 5
please input radius: 6
c[0]: center=(0,0), radius=1
c[1]: center=(1,2), radius=3
c[2]: center=(-4,5), radius=6
by adding point (1,1) to c[0] to c[2] assign the results to c[3] to c[5]
c[0]: center=(0,0), radius=1
c[1]: center=(1,2), radius=3
c[2]: center=(-4,5), radius=6
c[3]: center=(1,1), radius=1
c[4]: center=(2,3), radius=3
c[5]: center=(-3,6), radius=6
input value to get added to the radius of c[3] to c[5]: -2
error in operator+: negative radius, radius set to 0.0 instead
c[3]: center=(1,1), radius=0
c[4]: center=(2,3), radius=1
c[5]: center=(-3,6), radius=4
compare circles:
c[0] and c[1] are not equal
c[1] and c[2] are unequal
```

Last modified: Saturday, 13 November 2021, 5:20 PM

◀ A2 Upload Teil 1+2/Upload Part 1+2

Jump to...

A3 Teil 2: Präsenzaufgabe/Part 2: Presence Task ▶