

Program 6

```
#include <iostream>
using namespace std;
class Liczba {
double x;
public:
Liczba (double xx=12.5) : x{xx} {cout<<"***\n"; } //1
Liczba (const Liczba& wzor) : x{wzor.x*10} { }
Liczba& operator = (const Liczba & wzor)
{cout<<"$$$\\n"; x=wzor.x-5; return *this;}
void wypisz( ) {cout<<x<<endl;}
};
4
```

Programowanie obiektowe pierwsze kolokwium teoretyczne – zbiór zadań

```
int main( ){
Liczba obiekt{13.2};
obiekt.wypisz( );
Liczba x{obiekt};
x.wypisz( );
obiekt=x;
obiekt.wypisz( );
return 0;
}
```

Pytania:

* Czy w powyższym programie konstruktor kopiujący jest poprawnie zdefiniowany?

Odpowiedź uzasadnij. *

W powyższym programie konstruktor kopiujący nie jest poprawnie zdefiniowany. Konstruktor kopiujący jest odpowiedzialny za tworzenie nowego obiektu na podstawie istniejącego obiektu, ale w tym przypadku implementacja konstruktora kopiującego nie kopiuje danych poprawnie. Konstruktor kopiujący powinien tworzyć nowy obiekt z takimi samymi danymi jak obiekt źródłowy, ale w tym przypadku mnoży wartość x obiektu źródłowego przez 10, co jest niepoprawne. Poprawna implementacja konstruktora kopiującego powinna wyglądać tak:

```
Liczba (const Liczba& wzor) : x{wzor.x} { }
```

Czy w powyższym programie operator przypisania jest poprawnie zdefiniowany? Odpowiedź uzasadnij.

W powyższym programie operator przypisania jest poprawnie zdefiniowany. Operator przypisania jest odpowiedzialny za przypisanie wartości jednego obiektu do innego. W tym przypadku, operator przypisania odejmuje 5 od wartości x obiektu źródłowego i przypisuje tę zmienioną wartość do x obiektu docelowego. Operator zwraca referencję do obiektu docelowego (*this), co pozwala na łańcuchowe wywołania przypisań. Odpowiednie jest również wyświetlenie komunikatu "\$\$\$" w celu potwierdzenia działania operatora przypisania.

Podsumowując, konstruktor kopiujący musi skopiować dane z obiektu źródłowego, a operator przypisania działa poprawnie i jest zgodny z oczekiwanym zachowaniem.

