

# Zaawansowane C++

## Lista 5: `std::thread` , `std::async`

### Zadanie 1 `std::thread`

Napisz funkcję, która wygeneruje i zwróci liczbowy identyfikator wątku, z którego jest wywoływana, bez wykorzystania typu `std::thread::id`.

W przestrzeni nazw `cpp1ab` napisz funkcję, która wypisze na ekran tekst przekazany jako jej argument, poprzedzony identyfikatorem wątku z poprzedniego punktu. Pamiętaj o synchronizacji dostępu do obiektu `std::cout`, najlepiej z użyciem szablonu `std::lock_guard`.

### Zadanie 2 `std::async`

W przestrzeni nazw `cpp1ab` napisz funkcję, która będzie zawierać co najmniej trzykrotnie zagnieżdżone wywołanie funkcji `std::async`. Każde z nich powinno wypisywać na ekran informację o swoim wywołaniu przy użyciu funkcji z poprzedniego punktu. Argumentem funkcji powinna być polityka wywołania typu `std::launch`, która zostanie użyta do wszystkich wywołań `std::async`.

Przetestuj działanie funkcji z poprzedniego punktu z politykami `std::launch::async` oraz `std::launch::deferred`.

### Materiały pomocnicze:

- `std::threads` [https://www.youtube.com/watch?v=wXBcwHwIt\\_I&t=595s](https://www.youtube.com/watch?v=wXBcwHwIt_I&t=595s)
- threading introduction series [https://www.youtube.com/watch?v=TPVH\\_coGAQs&list=PLk6CEY9XxSIAeK-EAh3hB4fgNvYkYmghp](https://www.youtube.com/watch?v=TPVH_coGAQs&list=PLk6CEY9XxSIAeK-EAh3hB4fgNvYkYmghp)
- `std::async` elementary <https://www.youtube.com/watch?v=ehMg6zvXuMY&t=378s>
- `std::async` advanced <https://www.youtube.com/watch?v=5HWCsmE9DrE&t=865s>