Regulamin przedmiotu: Języki Programowania
prof. dr hab. inż. Hanna Paulina Zbroszczyk
Kontakt: hanna.zbroszczyk@pw.edu.pl,
Wydział Fizyki PW, biuro 117b (wejście przez 115)

I) Informacje ogólne:
Zajęcia trwają 15 tygodni (2 godziny wykładu, 2 godziny laboratorium tygodniowo)
Zaliczenie zajęć jest uwarunkowane zaliczeniem zajęć laboratoryjnych.

Prowadzący zajęcia laboratoryjne:
dr hab. inż. Małgorzata Janik
dr inż. Daniel Wielanek
mgr inż. Mateusz Grunwald
mgr inż. Michał Prędota

Materiały do wykładu będą sukcesywnie umieszczane na platformie MS Teams.

II) Organizacja zajęć laboratoryjnych:
- przewidzianych jest 14 zajęć laboratoryjnych (90 min przewidzianych na jedne zajęcia) realizowanych stacjonarnie, ostatnie, 15-te zajęcia są przeznaczone na wystawienie ocen;
- obecność jest obowiązkowa (możliwe są maksymalnie 2 nieobecności);

III) Zasady oceniania na zajęciach punktowanych:
- zajęcia punktowane obejmują wykonanie 14 (czternastu) zadań o zróżnicowanym stopniu trudności, za każde zadanie można otrzymać do 5 pkt;
- w trakcie pisania programu wolno korzystać z napisanych przez siebie programów, zasobów Internetu, dostępnej literatury;
- program należy napisać i oddać na tych samych zajęciach, w przypadku nie skończenia programu studenci otrzymują proporcjonalnie mniej punktów; program należy dokończyć we własnym zakresie i oddać na kolejnych zajęciach z możliwością uzyskania dodatkowego punktu (w przypadku programu kończonego po zajęciach można za niego otrzymać maksymalnie 4 pkt);
- nie oddanie programu może skutkować niedopuszczeniem do realizacji kolejnego programu;
- za każde zadanie można otrzymać 0-5 pkt (zrozumienie zadania: 1 pkt, wykorzystanie formalnych środków języka C++: 2 pkt, aspekty użytkowe oraz strona estetyczna wraz z obszernymi komentarzami w kodzie: 2 pkt)
- przynajmniej jeden, a maksymalnie dwa programy będą pisane przez dwa, niekoniecznie następujące po sobie zajęcia;
- w przypadku nie skończenia programu oceniony zostanie napisany, skompilowany oraz działający jego fragment (w przypadku programu, który nie kompiluje, ani nie wykonuje się poprawnie możliwe jest zdobycie maksymalnie 2 pkt);
- w przypadku nieobecności studenci są zobowiązani do zrealizowania materiału we własnym zakresie i przedstawienia rozwiązania najdalej 2 tygodnie po nieobecności (na zajęciach lub konsultacjach) - w przypadku usprawiedliwionej nieobecności (zwolnienie lekarskie, usprawiedliwienie władz Uczelni, Wydziału) możliwe jest zaliczenie zaległego programu; w przypadku nieobecności nieusprawiedliwionej liczba zdobytych punktów wynosi 0 (zero); nie nadrobienie zaległości (zarówno w przypadku nieobecności usprawiedliwionej i nieusprawiedliwionej) może skutkować niedopuszczeniem do kolejnych zajęć;

IV) Zasady oceniania projektów:
Zajęcia kończą się oddaniem indywidualnego projektu, którego tematyka będzie indywidualnie ustalana z prowadzącym. Zadanie projektowe musi zostać oddane przed zakończeniem semestru (maksymalnie w trakcie ostatnich zajęć w semestrze)
- próby niesamodzielnej pracy będą skutkowały niezaliczeniem projektu oraz brakiem możliwości jego poprawy - **niezaliczeniem przedmiotu!**;
- napisany projekt należy przesłać przed końcem semestru na adres e-mailowy prowadzącego / ew. GitHub;
- program będzie oceniany w skali 0-50 pkt; oceniane będą:
    - zakres merytoryczny zrealizowanego zadania,
    - wykorzystane środki formalne języka C/C++,
    - aspekty użytkowe interfejsu,
    - strona estetyczna;

V) Na ocenę końcowa przedmiotu wpływają :
Wyniki z programów napisanych na zajęciach 14 * 5 pkt = 70 pkt.
Wyniki z projektu: 1 * 50 pkt = 80 pkt;
Ocena pozytywne jest możliwa do uzyskaniu przy zdobyciu min. 35 pkt z programów pisanych na zajęciach i 25 pkt z projektu.
Ocena końcowa wystawiana jest na podstawie procentowego udziału sumy uzyskanych punktów do sumy punktów możliwej do uzyskania (120 pkt) wg. następującej zależności:
>= 60 pkt - 3.0
>= 72 pkt - 3.5
>= 84 pkt - 4.0
>= 96 pkt - 4.5
>= 108 pkt - 5.0

Dla osób uczęszczających na wykłady (możliwa 1 nieobecność), będzie możliwość uzyskanie dodatkowych 10 pkt.

VI) Zaliczenie eksternistyczne
Dla osób programujących w C++ możliwe jest zaliczenie przedmiotu projektem eksternistycznym. Osoby chcące zaliczyć przedmiot w tej formie powinny zgłosić się do prowadzącego najdalej na drugich zajęciach laboratoryjnych, na trzecich zajęciach napiszą kolokwium kwalifikujące do pracy w tym trybie.
Wymagania do projektów eksternistycznych:
- nietrywialny problem, do którego rozwiązania najlepiej nadaje się podejście obiektowe,
- dokładna specyfikacja projektu,
- stworzony projekt z dokumentacją w kodzie źródłowym,
- dokumentacja użytkownika.

I) Zaliczenie projektu eksternistycznego polega na zaliczeniu 3 (trzech) etapów kontrolnych w terminach zajęć podanych w nawiasach: *beta (5), release candidate (10), final (15)*.
II) Po etapie *beta* prowadzący może projekt zdyskwalifikować, dlatego do tego czasu zalecane jest uczestniczenie w zajęciach programowych.
III) Po etapie *release candidate*, w przypadku braku możliwości skończenia projektu zawierającego wszystkie elementy języka omawiane na wykładzie prowadzący może projekt zamknąć. Od tego momentu należy uczestniczyć w zajęciach.

IV) Przy ustalaniu oceny ostatecznej brane pod uwagę są oceny z etapów pośrednich.

Literatutra

1) B. Stroustrup – Język C++ (The C++ Programming Language).
2) J. Grębosz – Symfonia C++ standard, Pasja C++.

Program przedmiotu:

1. Historia i rozwój języka. Porównanie z Pythonem. Kompilacja programu. Wbudowane typy zmiennych.
2. Strumienie wejściowe/wyjściowe (w tym operacje na plikach).
3. Instrukcje warunkowe i pętle. Tablice, stringi, podstawowe kontenery STL.
4. Wskaźniki i referencje. Funkcje.
5. Struktury, unie, klasy.
6. Metody, konstruktory, destruktory, funkcje zaprzyjaźnione.
7. Przeciążanie operatorów.
8. Podział programu na jednostki kompilacji, Makefile, dokumentowanie kodu
9. Dziedziczenie.
10. Metody wirtualne, klasy abstrakcyjne.
11. Kontenery STL, algorytmy STL.
12. Szablony funkcji i klas.
13. Wyjątki.
14. Konwertery oraz konwersje.
15. Przykłady bibliotek rozszerzających.