

A TANTÁRGY ADATLAPJA

1. A képzési program adatai

| | |
|-----------------------------|----------------------------------|
| 1.1 Felsőoktatási intézmény | Babeş-Bolyai Tudományegyetem |
| 1.2 Kar | Matematika és Informatika |
| 1.3 Intézet | Magyar Matematika és Informatika |
| 1.4 Szakterület | Informatika |
| 1.5 Képzési szint | alap |
| 1.6 Szak / Képesítés | Informatika |

2. A tantárgy adatai

| | | | | | | | |
|---|--------------------------------------|-----------|---|----------------------|--------|---------------------|-----------------|
| 2.1 A tantárgy neve (hu) | Objektumorientált programozás | | | | | | |
| (en) | Object-oriented programming | | | | | | |
| (ro) | Programare orientată obiect | | | | | | |
| 2.2 Az előadásért felelős tanár neve | Dr. Darvay Zsolt egyetemi docens | | | | | | |
| 2.3 A szemináriumért felelős tanár neve | Dr. Darvay Zsolt egyetemi docens | | | | | | |
| 2.4 Tanulmányi év | 1 | 2.5 Félév | 2 | 2.6. Értékelés módja | vizsga | 2.7 Tantárgy típusa | kötelező – szak |
| 2.8 A tantárgy kódja | MLM5006 | | | | | | |

3. Teljes becsült idő (az oktatási tevékenység féléves óraszama)

| | | | | | |
|---|-----|----------------------|----|-------------------------|-----|
| 3.1 Heti óraszám | 5 | melyből: 3.2 előadás | 2 | 3.3 szeminárium / labor | 3 |
| 3.4 Tantervben szereplő össz-óraszám | 70 | melyből: 3.5 előadás | 28 | 3.6 szeminárium / labor | 42 |
| A tanulmányi idő elosztása: | | | | | óra |
| A tankönyv, a jegyzet, a szakirodalom vagy saját jegyzetek tanulmányozása | | | | | 28 |
| Könyvtárban, elektronikus adatbázisokban vagy terepen való további tájékozódás | | | | | 10 |
| Szemináriumok / laborok, házi feladatok, portofóliók, referátumok, esszék kidolgozása | | | | | 28 |
| Egyéni készségfejlesztés (tutorálás) | | | | | 10 |
| Vizsgák | | | | | 4 |
| Más tevékenységek: | | | | | |
| 3.7 Egyéni munka össz-óraszama | 80 | | | | |
| 3.8 A félév össz-óraszama | 150 | | | | |
| 3.9 Kreditszám | 6 | | | | |

4. Előfeltételek (ha vannak)

| | |
|---------------------|--|
| 4.1 Tantervi | <ul style="list-style-type: none"> Nincs. |
| 4.2 Kompetenciabeli | <ul style="list-style-type: none"> A C programozási nyelv ismerete. |

5. Feltételek (ha vannak)

| | |
|--|---|
| 5.1 Az előadás lebonyolításának feltételei | <ul style="list-style-type: none"> Táblával és videoprojektorral felszerelt előadóterem. |
|--|---|

| | |
|---|--|
| 5.2 A szeminárium / labor lebonyolításának feltételei | <ul style="list-style-type: none"> • Számítógépes terem, C++. |
|---|--|

6. Elsajátítandó jellemző kompetenciák

| | |
|------------------------------------|--|
| Szakmai kompetenciák | <p>C1.1 A programozási paradigmák és a specifikus nyelvi mechanizmusok megfelelő leírása, valamint a szemantikai és a szintaktikai vonatkozások közötti különbség meghatározása</p> <p>C1.2 A meglévő szoftveralkalmazások magyarázata absztrakciós szintek szerint (architektúra, csomagok, osztályok, metódusok) az alapismeretek használatával</p> <p>C1.3 Megfelelő forráskód fejlesztése egy ismert programozási nyelvben és a komponensek egységes tesztelése adott tervezési specifikáció alapján</p> <p>C1.4. Alkalmazások tesztelése adott tesztelési terv alapján</p> <p>C1.5 A programegységek fejlesztése és a kapcsolódó dokumentáció megvalósítása</p> |
| Transzverzális kompetenciák | <p>CT1 A szervezett és hatékony munka szabályainak, a didaktikai-tudományos területhez való felelősségteljes hozzáállás alkalmazása a saját potenciál kreatív értékesítéséhez, a szakmai etika alapelveinek és normáinak tiszteletben tartásával.</p> <p>CT3 Hatékony módszerek és technikák használata tanulásra, információszerzésre, kutatásra és a tudásszerzési kapacitások fejlesztésére, egy dinamikus társadalom igényeinek való megfelelésre, román és egy nemzetközi nyelven történő kommunikációra.</p> |

7. A tantárgy célkitűzései (az elsajátítandó jellemző kompetenciák alapján)

| | |
|--------------------------------------|---|
| 7.1 A tantárgy általános célkitűzése | <ul style="list-style-type: none"> • Az objektumorientált programozás alapvető fogalmainak és elveinek elsajátítása. |
| 7.2 A tantárgy sajátos célkitűzései | <ul style="list-style-type: none"> • A C++ programozási nyelv alapjainak megtanulása. • A hagyományos és az objektumorientált tervezés közötti különbségek felismerése. • Az öröklés, a polimorfizmus, a dinamikus kötés és a sablonok használatának a megértése, újrafelhasználható kód fejlesztése érdekében. • Különböző alkalmazások fejlesztése objektumorientált programozással. • A szabványos könyvtár adatszerkezeteinek és algoritmusainak felhasználása. • Mások által megírt osztályok felhasználása egy adott feladat megoldására. |

8. A tantárgy tartalma

| 8.1 Előadás | Didaktikai módszerek | Megjegyzések |
|---|--|--------------|
| <p>1. A C++ nyelv alapelemei.</p> <ul style="list-style-type: none"> - Lexikális egységek. Operátorok. Konverzió. - Adattípusok. Változók. Állandók. - Változók láthatósága és élettartama. - Névterek. | Előadás, párbeszéd, problematizálás, tanári magyarázat | |
| <p>2. Utasítások és C++ függvények.</p> <ul style="list-style-type: none"> - C++ utasítások. - Függvénydeklaráció és definíció a C++-ban. - Függvények túlterhelése. - Inline függvények. | Előadás, párbeszéd, problematizálás, tanári magyarázat | |

| | | |
|---|---|--|
| <p>3. Az alaptípusoknál összetettebb típusok és a dinamikus memóriakiosztás a C++-ban.</p> <ul style="list-style-type: none"> - Tömb és struktúra. - Mutató és referencia típus. - Memóriaterület lefoglalása és felszabadítása. - Függvényekre hivatkozó mutatók és void függvények. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>4. Moduláris programozás C++-ban.</p> <ul style="list-style-type: none"> - Fejállományok. Könyvtárak. - Absztrakt adattípusok moduláris megvalósítása. - A void típusra hivatkozó mutatók használata az általánosság érdekében. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>5. Objektumorientált programozás C++-ban.</p> <ul style="list-style-type: none"> - Osztályok és objektumok. - Egy osztály tagjai. Hozzáférésmódosítók. - Konstruktorok és destruktorok - UML osztálydiagramok (tagok, hozzáférés). - Ismeretség és aggregáció megadása UML diagramban | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>6. Operátorok túlterhelése. Statikus tagok. Barát függvények és osztályok.</p> | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>7. Öröklés</p> <ul style="list-style-type: none"> - Származtatott osztályok. Egyszeres öröklés. - Helyettesítési elv. - Metódusok felülírása. - Többszörös öröklés. - Osztályok közti specializálási/általánosítási kapcsolat megadása UML-ben. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>8. Polimorfizmus.</p> <ul style="list-style-type: none"> - Virtuális tagfüggvények. - Dinamikus kötés. - Virtuális öröklés. - Kód újrafelhasználás (öröklés/összetétel). - Konverzió (upcast/downcast). | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>9. Objektumorientált tervezés és interfészre alapozott tervezés.</p> <ul style="list-style-type: none"> - Absztrakt osztály, interfész. - Az interfész megadása UML –ben. - Egy adatszerkezetekből álló könyvtár objektumorientált tervezése. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>10. Bemeneti/kiviteli műveletek.</p> <ul style="list-style-type: none"> - Adatfolyamok. Bemeneti/kiviteli osztályhierarchiák. - Formátumozott bemenet/kimenet. Módosítók. - Állománykezelés. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>11. Sablonok.</p> <ul style="list-style-type: none"> - Függvénysablonok. - Osztálysablonok. - Kód újrafelhasználás. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>12. A szabványos C++ könyvtár (STL)</p> <ul style="list-style-type: none"> - Tárolók és bejárók. - A szabványos könyvtár algoritmusai. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |
| <p>13. Kivételkezelés.</p> <ul style="list-style-type: none"> - A kivétel fogalma. - Kivételek kiváltása és kezelése C++-ban. | <p>Előadás, párbeszéd, problematizálás, tanári magyarázat</p> | |

| | | |
|--|--|--------------|
| 14. A Visual C++ fejlesztési környezet. - A grafikus felhatalnáló felület elemei. | Előadás, párbeszéd, problematizálás, tanári magyarázat | |
| <p>Könyvészet</p> <p>1. Alexandrescu, Programarea modernă in C++. Programare generică si modele de proiectare aplicate, Editura Teora, 2002.(Könyvtár).</p> <p>2. R. Andonie, I. Garbacea, Algoritmi fundamentali. O perspectiva C++, Editura Libris, Cluj_Napoca, 1995 (Könyvtár).</p> <p>3. Eckel, Thinking in C++, (second edition), vol. 1 & 2, [www.bruceeckel.com]</p> <p>4. Milewski, C++ in Action, [http://www.relisoft.com/book/], 1997.</p> <p>5. L. Negrescu, Limbajul C++, Ed. Albastra, Cluj-Napoca 1996(Könyvtár).</p> <p>6. Vasile Cioban, Zsolt Darvay, Metode evaluate de programare, UBB-Mate_Info, 1999 (Könyvtár).</p> <p>7. M. Frențiu, B. Pârv, Elaborarea programelor. Metode și tehnici moderne, Ed. Promedia, Cluj-Napoca, 1994. (Könyvtár).</p> | | |
| 8.2 Szeminárium / Labor | Didaktikai módszerek | Megjegyzések |
| Sz1. Függvényre hivatkozó mutató. Absztrakt adattípus. Moduláris programozás. | feladat, egyéni munka, párbeszéd | |
| Sz2. Névtér. Osztály több konstruktorral. | feladat, egyéni munka, párbeszéd | |
| Sz3. Osztály típusú adattag. Kivételkezelés. | feladat, egyéni munka, párbeszéd | |
| Sz4. Osztálysablon. Operátor túlterhelés. | feladat, egyéni munka, párbeszéd | |
| Sz5. Virtuális tagfüggvény. Absztrakt osztály. | feladat, egyéni munka, párbeszéd | |
| Sz6. Típuskonverzió. | feladat, egyéni munka, párbeszéd | |
| Sz7. Saját adatszerkezet és hozzá tartozó bejáró megvalósítása. | feladat, egyéni munka, párbeszéd | |
| L1. Bevezető gyakorlatok a C++ programozási nyelvben. | feladat, egyéni munka, párbeszéd | |
| L2. Egyszerű C++ feladatok függvényekkel. | feladat, egyéni munka, párbeszéd | |
| L3. Adatvédelem moduláris programozással a C-ben. Névterek a C++-ban. | feladat, egyéni munka, párbeszéd | |
| L4. Osztályokra vonatkozó alapvető feladatok. Osztály típusú adattag. | feladat, egyéni munka, párbeszéd | |
| L5. Feladatok kivételkezeléssel. | feladat, egyéni munka, párbeszéd | |
| L6. Feladatok osztálysablonokkal és függvénysablonokkal. | feladat, egyéni munka, párbeszéd | |
| L7. Operátorok túlterhelésére vonatkozó feladatok. | feladat, egyéni munka, párbeszéd | |
| L8. Adatszerkezetek láncolt megvalósítása. Implicit típuskonverzió. | feladat, egyéni munka, párbeszéd | |
| L9. Explicit típuskonverzió. Virtuális tagfüggvények. | feladat, egyéni munka, párbeszéd | |
| L10. Virtuális, illetve absztrakt osztályokkal kapcsolatos feladatok | feladat, egyéni munka, párbeszéd | |
| L11. Saját adatszerkezet és hozzá kapcsolódó bejáró implementálása. | feladat, egyéni munka, párbeszéd | |
| L12. Az STL tárolóinak használata. | feladat, egyéni munka, párbeszéd | |
| L13. Az STL algoritmusainak használata. | feladat, egyéni | |

| | | |
|-----------------------------|--|--|
| | munka, párbeszéd | |
| L14. Projektek kiértékelése | feladat, egyéni munka, párbeszéd, gyakorlati projekt | |

Könyvészet

1. B. Stroustrup, The C++ Programming Language, Addison Wesley, 1997
<http://public.research.att.com/~bs/3rd.html>.
2. B. Stroustrup: The C++ Programming Language Special Edition, AT&T, 2000.
3. Peter Müller: Introduction to Object-Oriented Programming Using C++, Globewide Network Academy (GNA), www.gnacadey.org/ August 31, 1997
<http://www.zib.de/visual/people/mueller/Course/Tutorial/tutorial.html>
4. Mark J. Sebern: STL Iterator Classes, Version 1.2 (1/4/1998),
<http://www.msoe.edu/eecs/ce/courseinfo/stl/iterator.htm>
5. C++ Standard Template Library, <http://www.cppreference.com/cppstl.html>
6. John Copp: C++ Programming Tutorial, <http://cplus.about.com/od/beginnerctutorial/1/blcplustut.htm>
7. B. Parv, A. Vancea, Fundamentele limbajelor de programare, Microinformatica, Cluj-Napoca, 1996 (Könyvtár).

9. A tárgy tartalmának összhangba hozása az episztemikus közösségek képviselői, a szakmai egyesületek és a szakterület reprezentatív munkáltatói elvárásaival.

- A tantárgy tartalma összhangban van a fontosabb egyetemeken oktatott objektumorientált programozásra, illetve C++ programozási nyelvre vonatkozó előadásokkal.

10. Értékelés

| Tevékenység típusa | 10.1 Értékelési kritériumok | 10.2 Értékelési módszerek | 10.3 Aránya a végső jegyben |
|--------------------------|----------------------------------|---|-----------------------------|
| 10.4 Előadás | Az elméleti anyag ismerete | Elméleti teszt | 40% |
| 10.5 Szeminárium / Labor | Szemináriumi tevékenység | Szemináriumi gyakorlatok | 10% |
| | Laboratóriumi tevékenység | Laboratóriumi gyakorlatok és házi feladatok | 15% |
| | Összetett alkalmazások készítése | Projektek | 10% |
| | A gyakorlati anyag elsajátítása | Gyakorlati vizsga | 25% |

10.6 A teljesítmény minimumkövetelményei

- A C és C++ programozási nyelvek alapjainak ismerete. A két nyelv közötti különbségek felismerése.
- Az objektumorientált programozásra vonatkozó fogalmak ismerete.
- A C++ objektumorientált lehetőségeinek kihasználása egy adott alkalmazás fejlesztésekor.
- Az elméleti teszten és a gyakorlati vizsgán legalább 50%-ot kell elérni.

Kitöltés dátuma

Előadás felelőse

Szeminárium felelőse

Dr. Darvay Zsolt, egyet. docens

Dr. Darvay Zsolt, egyet. docens

2022. ápr. 30.

.....

.....

Az intézeti jóváhagyás dátuma

Intézetigazgató

2022. ápr. 30.

Dr. András Szilárd, egyet. docens

.....