

| | |
|---|---|
| 1. Ismertesse a C++ nyelvben alkalmazott bővítéseket az egyszerű adattípusok terén, valamint a konzol ki/bemenet megvalósításait! | N |
| 2. Ismertesse a csak C++ nyelvben alkalmazható típuskonverziót, valamint a header fájlok használatánál alkalmazható egyszerűsítést! | N |
| 3. Ismertesse a referencia típusú változók deklarációját, használatát, és a referencia adattagot tartalmazó osztályok konstruktörának megvalósítását! Írjon példát a referencia típus használatára függvényben! | N |
| 4. Ismertesse a C++ nyelvben a függvények alapértelmezett paraméterezősének lehetőségét, és ennek szabályait! | N |
| 5. Ismertesse a C++ nyelvben a függvények túlterhelésének lehetőségét és ennek szabályait! | N |
| 6. Ismertesse a C++ nyelvben a template-ek működését függvény és osztály definiálása során! Írjon példán template-tel deklarált függvényre és használatára! | N |
| 7. Ismertesse a C++ nyelv memóriafoglalás és felszabadítás operátorait dinamikus példányok létrehozására és megszüntetésére! Írjon példát egy n elemű, double típusú adatokat tartalmazó tömb létrehozására és megszüntetésére! | N |
| 8. Ismertesse a C++ hibakezelésben használható try-catch blokk működését! | N |
| 9. Ismertesse az „egysége zárás” objektum-orientált elvet! | N |
| 10. Ismertesse az „adatrejtés” objektum-orientált elvet! | N |
| 11. Ismertesse az „öröklődés” objektum-orientált elvet! | N |
| 12. Ismertesse a „sokalakúság” objektum-orientált elvet! | N |
| 13. Ismertesse a „this” pointer alkalmazását a fordító és a felhasználó szemszögéből! | N |
| 14. Ismertesse a „private”, „protected”, „public” módosítók működését az osztálytagok definiálásakor! | N |
| 15. Ismertesse a „const” és „mutable” módosítók működését az osztálytagok definiálásakor! | N |

| | |
|--|---|
| 16. Ismertesse a statikus adattagok tulajdonságait, megadási és elérési módjait! | N |
| 17. Ismertesse a „barátság” elvét és típusait az osztályok definiálásánál! | N |
| 18. Ismertesse a konstruktur működését! Mely konstruktorkat biztosítja a fordító alapértelmezetten? | N |
| 19. Ismertesse a konstruktur megadásának szabályait! Milyen esetekben kell felülírnuk a fordító által definiált konstruktorkat? | N |
| 20. Ismertesse az adattagok kezdeti értékmegadásának lehetőségeit! Ezek közül melyik az, amelyik referencia típusú adattagok esetén használható? | N |
| 21. Ismertesse példával az 1 paraméterrel rendelkező konstruktur egyszerűsített meghívási lehetőségét! Hogyan tudjuk ezt az egyszerűsítést letiltani? | N |
| 22. Ismertesse a másoló konstruktur megírásának szükségességét okozó szituációt! Honnan tudjuk eldönteni, hogy a fordító a másoló konstruktort, vagy az „=” operátort használja? | N |
| 23. Ismertesse a destruktur definícióját, a destruktur készítés szabályait! Mit mondhatunk a destruktur kézi meghívásáról? | N |
| 24. Ismertesse a névterek definiálásának szükségességét a C++ programokban! Melyik operátorral hivatkozhatunk egy adott névterben található osztályra? | N |
| 25. Ismertesse az osztálypéldányokon végzett műveletek definiálási lehetőségeit! Mely műveleteket nem lehet átdefiniálni? | N |
| 26. Ismertesse az osztályok kétoperandusú műveleteinek átdefiniálási lehetőségeit! Írjon példákat minden egyes lehetőséghez! | N |
| 27. Ismertesse az osztályok egyoperandusú műveleteinek átdefiniálási lehetőségeit! Írjon példákat minden egyes lehetőséghez! | N |
| 28. Ismertesse a kommutatív műveletek átdefiniálási lehetőségét! Miért nem tudjuk a tagfüggvényes módszert alkalmazni? | N |
| 29. Ismertesse a „()” operátor túlterhelési lehetőségeit! | N |
| 30. Ismertesse az „=” operátor túlterhelésének szintaktikáját és a szituációt, amelyben a fordító által biztosított operátor nem működik megfelelően! | N |

| | |
|---|---|
| 31. Ismertesse a „new” és „delete” operátorok túlterhelésének szabályait! | J |
| 32. Ismertesse az I/O operátorok túlterhelésének szabályait! Írjon példát osztálypéldány kiíratásához! | J |
| 33. Ismertesse az „std” névtér „string” osztályát! Adja meg (működés magyarázatával) gyakran használt operátorait és metódusait! | J |
| 34. Ismertesse a string-numerikus adat közti konverzióra használt osztályt! | J |
| 35. Ismertesse a fájlok kezelésére használt osztályt, gyakran használt metódusait és operátorait! | J |
| 36. Ismertesse példával a „kompozíció” elvet osztályok egymásba ágyazására! | J |
| 37. Ismertesse az „aggregáció” elvet osztályok egymásba ágyazására! | J |
| 38. Ismertesse az „öröklődés” elvet osztályok egymásba ágyazására! Mi az öröklődés szintaktikája a C++-ban? | J |
| 39. Csoportosítsa az osztályban található elemeket öröklődési szempontból: mely elemek öröklődnek, és mely elemek nem öröklődnek? | J |
| 40. Ismertesse öröklődés során a leszármazottban található konstruktor paramétereinek és hívásának szabályait, tekintettel az ősben levő privát adattagokra! | J |
| 41. Ismertesse az ősben található osztálytagok elérésének módosítását private és protected öröklődés során! | J |
| 42. Ismertesse az osztálytagok elérését ősosztály típusú pointerrel! Mi a „korai kötés” működése és problémája? | J |
| 43. Ismertesse a C++-ban található „többszörös öröklődés” elvet! Az ismertetést ábrával és program-részlettel illusztrálja! | J |
| 44. Ismertesse példával a „virtuális metódus” elv működését! Mit tartalmaz a VMT (vftable) táblázat? A leszármazottban is ugyanazt a szintaktikát kell használni a virtuális metódus felülírásakor? | J |
| 45. Ismertesse ábrával a „közvetlen bázisosztály” és a „közvetett bázisosztály” fogalmakat! | J |

| | |
|--|---|
| 46. Ismertesse ábrával a virtuális öröklődés szükségességét előidéző szituációt! | J |
| 47. Ismertesse a tisztán virtuális metódus készítésének szintaktikáját! Hogyan nevezzük a legalább 1 tisztán virtuális metódust tartalmazó osztályt? Milyen szabályok vonatkoznak erre az osztályra? | J |
| 48. Ismertesse az „overload” és „override” elvek közti különbséget, amennyiben ős és leszármazottban történő előfordulásról van szó! | J |
| 49. Ismertesse a „static_cast” és „dynamic_cast” kulcsszavak működését! Hol fordulhat elő hibásan interpretált memória-terület? | J |
| 50. Ismertesse egy előre megírt programrendszer (neutról letöltött, vagy eszközzel kapott SDK) használatának lépéseiit C++-ban! | J |
| 51. Ismertesse a vector STL tároló tulajdonságait (memória modell, bejárás, bővíthetőség)! | N |
| 52. Ismertesse a deque STL tároló tulajdonságait (memória modell, bejárás, bővíthetőség)! | N |
| 53. Ismertesse a list STL tároló tulajdonságait (memória modell, bejárás, bővíthetőség)! | N |
| 54. Ismertesse a set, multiset STL tárolók tulajdonságait (memória modell, bejárás, bővíthetőség)! | N |
| 55. Ismertesse a map, multimap STL tárolók tulajdonságait (memória modell, bejárás, bővíthetőség)! | N |
| 56. Ismertesse az STL tároló adaptereket és működésüket! Melyik mire használható? | N |
| 57. Ismertesse az STL iterátorok működését és feladatát egy lista STL tároló esetén! | N |
| 58. Ismertesse az STL tárolókon végrehajtható algoritmusok működését és testre szabási lehetőségeiket! | N |
| 59. Ismertesse a „nyelvi változó”, „karakterisztikus függvény” és a „tagsági függvény” fogalmakat Zadeh szerint! | J |
| 60. Ismertesse példával azt a szituációt, amikor egy fuzzy partíció lefedi az alaphalmazt! Adja meg a szöveges definíciót is! | J |

| | |
|---|---|
| 61. Ismertesse ábrával a „mag”, „tartó”, „a vágat”, „szigorú a vágat” és „magasság” fogalmakat a fuzzy halmazok esetén! | J |
| 62. Ismertesse a Zadeh szerinti „s-norma”, „t-norma” és komplementis képzését fuzzy halmazoknál! | J |
| 63. Ismertesse, hogy mikor alkalmazható egy t-norma, s-norma, komplementis definícióit tartalmazó szabályrendszer fuzzy halmazműveletekhez! Mit alkotnak ilyenkor a szabályrendszer elemei? | J |
| 64. Ismertesse a fuzzy rendszerek általános blokkvázlatát! | J |
| 65. Ismertesse példával a fuzzy következtető módszer működését! | J |
| 66. Ismertessen defuzzifikációs módszereket! | J |
| 67. Ismertesse az aggregációs operátorok definícióját, és 5 axiómáját! | J |
| 68. Ismertesse az általános hatványközép operátort, és a paraméter speciális eseteiben elnevezett értékeit! | J |
| 69. Ismertesse az OWA operátort! | J |