Објектно оријентисано програмирање 2

- Домаћи задатак -

Домаћи

Пре него што почнете прочитајте поглавље [УПУТСВО](#page6) на крају овог документа. Ту ћете пронаћи упутства како да попуните овај документ и како треба да доставите своја решења задатака из овог домаћег.

**ЗАДАЦИ**

1. Прочитајте поглавља 2 и 3 књиге Бјарнеа Строуструпа „*Programming Principles and Practices C++"*. Научите значења свих појмова из тихпоглавља одговорите на следећа питања:
   1. Набројте четири дела дефиниције функције:
2. Повратни тип
3. Назив функције
4. Аргументи
5. Тело функције
6. Приликом учитавања са стандардног улаза, шта означава крај једног уноса за целе бројеве и знаковне низове?

Специјални карактер \n (end of line)

1. Шта је објекат?

Део меморије који може да садржи вредност одређеног типа

1. Шта је литерал? Наведите неколико типова литерала.



Логички: true и false, Знаковни: ‘a’, ‘b’, Целобројни: 0,1,2

То су константе које представљају вредност без декларисања променљиве

1

Објектно оријентисано програмирање 2

- Домаћи задатак -

1. Шта су променљиве? Наведите основне типове променљивих и њихове уобичајене величине.

Променљива је објекат који има име. Логички тип (bool), 1 бајт

Знаковни тип (char) - 1 бајт, Интеџер – 4 бајта, Реални (floating point) - 8

бајтова

1. Која је разлика између операција = и ==? Ког типа је њихов резултат?

Операција == пореди 2 вредности и враћа вредност типа bool (true ако

су једнаке). Операција = додељује вредност, и враћа показивач

(референцу) на објекат коме је додељена вредност.

1. Шта је дефиниција, а шта декларација?

Декларација придружује име и тип објекту, а дефиниција придружује

вредност

1. Шта је иницијализација и по чему се разликује од доделе?

Иницијализација је додељивање вредности у тренутку конструкције

Објекта.

1. Како се знаковни низови надовезују у Це++ језику?

Оператором +=, тј конкатенацијом се копира садржај знаковних низова

У нови низ.

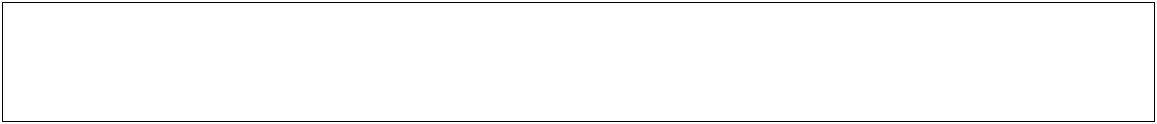
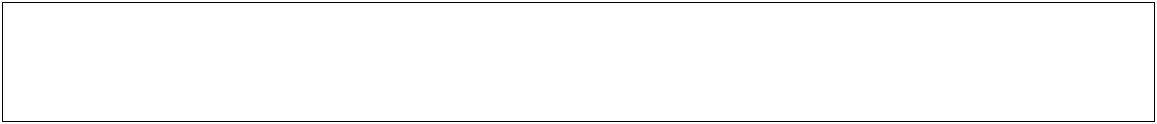
1. Шта је типска безбедност и зашто је важна?

Скуп правила које намеће језик, таквих да се сваки објекат може

Користити само у складу са својим типом. Важна је зато што нам

Омогућава дефинисано и предвидиво понашање операција.

1. Да ли је смештање вредности **double** типа у променљиву **int** типа типски безбедно? Зашто?



Не, зато што double заузима више простора од int па може доћи до губитка података

2

Објектно оријентисано програмирање 2

- Домаћи задатак -

II. Напишите добро структуиран и коментарисан Це++ програм који садржи три функције описане у наставку. Када се овде каже „хексадецимални број“ мисли се на стринг са хексадецималним записом броја.

1. Функција 1 – прима неозначени цео број и враћа стринг са записом тог броја у хексадецималној форми. Функција треба да баци изузетак у случају неодговарајућег улаза.
2. Функција 2 – прима два хексадецимална броја (стрингове са хексадецималним записом броја) и враћа њихов збор исто у хексадецималном облику. Функција треба да баци изузетак у случају неодговарајућег улаза.
3. Функција 3 – прима хексадецимални број и враћа целобројну вредност која одговара том хексадецималном запису. Функција треба да баци изузетак у случају неодговарајућег улаза.
4. Позовите ове функције из мејн функције. Употребите трај (try) и кеч (catch) механизме.
5. Проверите исправност вашег програма помоћу примера:
   1. исправног улаза
   2. неисправног улаза
   3. празног улаза

3

Објектно оријентисано програмирање 2

- Домаћи задатак -

1. Напишите програм који очекује од корисника да унесе две вредности истог основног типа (**short**, **int** или **double**). Те вредности треба сместити у променљиве **val1** и **val2**. Затим треба одредити мањи и већи од та два броја, њихов збир, разлику, производ и количник, и исписати те резултате на стандардни излаз. Употребите три функције, по једну за сваки тип променљивих. Омогућити кориснику да изабере коју од те три функције (та три типа) жели.

4

Објектно оријентисано програмирање 2

- Домаћи задатак -

IV. Напиши програм који рачуна Фибоначијеве бројеве. Формула за генерисање Фибоначијевих бројева дата је у наставку:

Fn = Fn-1 + Fn-2,

гдејеF1 = 1 иF2 = 2.

Например,F3 = F2 + F1 = 2 + 1 = 3.

Приметите да ће у једном тренутку Фибоначијеви бројеви бити превелики за **int** (или било који други) тип и доћи ће до прекорачења. Размислите како да прекинете рачунање бројева када дође до прекорачења, или непосредно пре него што до прекорачења дође, и то имплементирајте.

Напишите три посебне функције за три различита типа: **int**, **long** и **long**

**long**.

**Питања**:

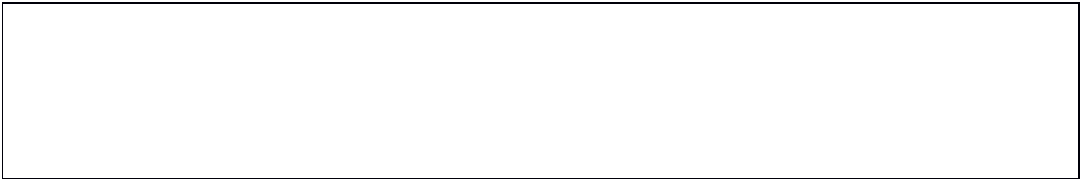
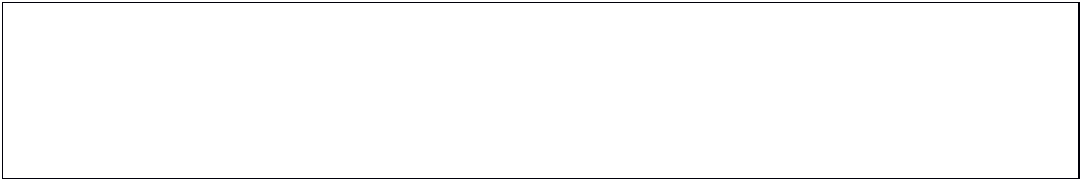
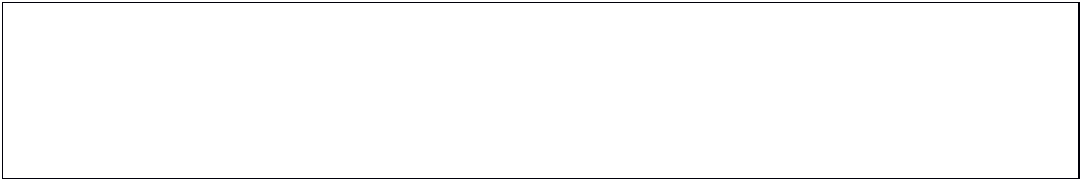
1. Колико првих Фибоначијевих бројева можемо сместити у тип **int**? Која је вредност највећег?

Највећи је 1836311903, и то је 45. Фибоначијев број.

1. Колико првих Фибоначијевих бројева можемо сместити у тип **long**? Која је вредност највећег?

Највећи је 1836311903, и то је 45. Фибоначијев број.

1. Колико првих Фибоначијевих бројева можемо сместити у тип **long long**? Која је вредност највећег?



7540113804746346429, 91. Фибоначијев број.

5

Објектно оријентисано програмирање 2

- Домаћи задатак -

**УПУТСТВО:**

Напишите одговоре у одговарајућа поља. Проширите величину поља ако вам треба више места. Решења за задатке II, III и IV, треба да буде приложено уз овај документ. Све то спакујте у Зип архиву. Архиву треба назвати овако:

***ИМЕ\_ПРЕЗИМЕ\_ИНДЕКС\_domaci.zip***

(Где *ИМЕ, ПРЕЗИМЕ* и *ИНДЕКС* треба заменити са вашим именом, презименом и бројем индекс)

Архива треба да садржи:

Датотеке:

**domaci.docx** (са одговорима на питања)

Директоријуме:

1. – који садржи пројекат и изворни код за задатак 2
2. – који садржи пројекат и изворни код за задатак 3

**IV** – који садржи пројекат и изворни код за задатак 4

* решењима задатака треба користити само оне елементе језика Це++ које смо спомињали током прве три недеље на предавањима. Циљ није да се размећете напредним знањем Це++ синтаксе.

**Решење послати у задатом формату предметном асистенту путем емаил-а.**

* **SUBJECT поље уписати: [OOP2 domaci][Име][Презиме][Индекс]. Рок за предају је уторак, 08.11. у поноћ.**

**Решења која нису исправно форматирана или послата на време неће бити прегледана.**

**ВАЖНО**

Пројектни директоријуми (*II*, *III* и *IV*) **НЕ СМЕЈУ** садржати извршне или објектне датотеке!!! Дозвољене су искључиво следеће датотеке:

.vcxproj

.sln

.filters

.cpp

.h 6

.hpp

**Изоставити** аутоматски направљене директоријуме: *Debug*, *Release*, *ipch* и сл.