Jméno a podpis studenta ……..…..… ID …….. závěrečná zkouška BPPC náhradní termín 2015-03-06

1	2	3	4		♥

Pro každý příklad použijte zvláštní stránku, příklad jasně označte, pište čitelně a přehledně. Vyškrtané raději celé přepište. Dodržujte konvence psaní klíčových slov pro C/C++. Nezaměňujte C++ a jiný jazyk, tato zkouška je z C++. V příkladech s funkcemi vyznačte, co by bylo v `.h` a co v `.c` (příp.`.cpp`) souboru. **Nepoužívejte globální proměnné**. Přečtěte si, co zadání požaduje (požaduje-li funkci, pište kompletní funkci, tj. včetně hlavičky a typu argumentů), má-li se napsat funkce, která je ekvivalentem funkce knihovní, potom se rozumí, že v ní nebude použita žádná knihovní funkce. Potřebujete-li existující funkci z knihovny jazyka C/C++, která není přímo předmětem zadání, je nutné dodržet její standardní volání. Součástí odevzdání je i tento list zadání. K psaní prosím používejte propisku.

1) Pro jazyk C++ definujte:
Pozor: při odpovědích se neodkazujte na jiné příklady z této zkoušky. Zdrojový text není samovysvětlující, vždy uveďte slovní popis.
a) Popište pojmy datový typ struktura a třída (rozdíly v jazyku C a C++, co to je, k čemu to slouží, klíčová slova a definice, příklad použití (deklarace, definice, inicializace), rozdíly) ***(2b)***
b) Vysvětlete rozdíl mezi prací s pamětí (alokace, odalokace) v jazyce C a C++ (uveďte názvy příslušných funkcí či klíčová slova a vysvětlete jejich funkci. Popište rozdíl mezi mechanizmy v C a C++. Uveďte jednoduchý příklad použití (pro proměnnou a pole proměnných)). Popište mechanizmus a ošetření případu neúspěšné alokace. ***(4b)***
c) Vysvětlete pojem polymorfizmus v jazyce C++. (Srovnejte je s „klasickým“ mechanizmem, vyjmenujte výhody a nevýhody. Uveďte možná použití tohoto mechanizmu.) ***(3b)***

2) Napište třídu `CBits`, která zajišťuje práci s bity v poli. Pole je realizováno jako dynamické (s proměnnou délkou), obsahuje pole typu `bool iBity` a počet využitých bitů `iPocet`. Pole bude obsahovat proměnnou typu `bool iValid`, která bude říkat, že je proměnná inicializovaná a má platná data. V definici třídy uvádějte **pouze prototypy** metod. Rozlište mezi hlavičkovým ***(11b)*** a zdrojovým souborem ***(10b)***. Třída musí být použitelná pro následující příklad a musí obsahovat i implicitně vytvářené metody. Operátory implementujte tak, že mají podobné vlastnosti jako operátory standardních typů. Pro změnu velikosti pole využijte privátní metodu `Resize()`. Aktuální instance třídy `CBits` budou počítány pomocí statické proměnné `iCelk`. U metod v hlavičce napište do komentáře, na kterých řádcích jsou volány (viz. čísla na levém okraji následujícího programu).

```
1  {                                                                    (3b)
2   CBits aa("<10;1011101011>"), c, d(512,true), f(8);//inicializace řetězcem "<počet;bity>",
   // bez přímo zadané hodnoty (prázdný), se dvěma (počet a hodnota všech bitů) a jedním parametrem
   // (počet,hodnota je implicitně false).                              (2b)
   // operátor = slouží k přiřazení dat proměnné zprava do proměnné vlevo
3   d = d &= c | f; //operátory | & &= fungují jako standardní bitové operátory,(výsl.je opět hodnota
   // typu CBits s počtem bitů většího z operandů)                      (1b)
4   c = true ^ c; // provede operaci ^ levé hodnoty s každým bitem pole, definováno "mimo třídu"(1b)
5   CBits g(Count(aa)); // Count() vrátí neznaménkový počet nenastavených bitů. Nemění objekt aa,
6   CBits h = c ^ false; // provede operaci ^ pravé hodnoty s každým bitem pole,   (1b)
7   if(aa[1]) // operátor[] třídy CBits, vrací hodnotu na dané pozici v poli iBity  (2b)
8    cout << "Tohle je f: " << f << "Pocet bitu: " << +aa.Pocet()<< endl;
   // tiskne hodnotu instance ve formátu použitém v c'toru z řetězce Pocet() vrácí počet nastav. bitů
9   cout << CBits::Celkem(); // tisk aktualních instancí třídy CBits, viz.statická proměnná
10 }                                                                     (1b)
```

3) Co bude vypsáno na konzole po skončení běhu tohoto programu. Tištěný text prosím napište k příslušnému řádku funkce `main`, kterého se týká. V případě, že daný řádek nevypisuje nic, uveďte u něj slůvko „*nic*“. ***(10b)***

```
class A {
public:
      A(void) {cout << 'x';}
      virtual ~A(void) {cout << 'y';}
      void aaa(void) {cout << 'p'; bbb();}
      virtual void bbb(void){ cout<<'r';}
};


class B:public A {
      A *a;
public:
      B(void) {cout << 's';}
      B(int z) {cout << z;}
      virtual ~B(void) {cout << 't';}
      void aaa(void) {bbb();cout << 'k';}
      virtual void bbb(void) {cout << 'l';}
};

class C:public B {
      B c;
public:
      C(void) {cout << 'm';}
      C(int i) : B() {cout << 'n';}
      virtual ~C(void) {cout << 'o';}
      void operator~() {cout << 'q';}
      void aaa(void) {cout << 'f';bbb();}
      virtual void bbb(void){cout <<'g';}
};

int main ()
{
 B d;              //_______

 A *c = (A*)new C; //_______
 C b(1);           //_______

 b.aaa();          //_______
 b.bbb();          //_______
 c -> aaa();       //_______
 c -> bbb();       //_______

 delete c;         //_______

 A a;              //_______
 c = (A*)&d;       //_______
 c -> aaa();       //_______
 c -> bbb();       //_______

 return 0;
}                  //_______
```