ÖREBRO UNIVERSITET

INSTITUTIONEN FÖR TEKNIK

Lösningarna till tentamensuppgifterna sätts ut på kurssidan på nätet i dag kl 13.

Denna tenta kommer att vara färdigrättad Må 24/10 och kan då hämtas på mitt tjänsterum T2221 mellan kl 13 och 15. Vid detta tillfälle har du också möjlighet att diskutera rättningen. Tentor som ej hämtats då placeras på studentexpedition och då har du ingen möjlighet att klaga på rättningen.

Tentamen i Objektorienterad Programmering 5p, Au, D, Fri, Pr, 051019

Hjälpmedel	: Inga		
Tid	: 08 - 13		
Ansvarig lärare	: Gunnar Joki	Tel :	arb 303317
			hem 274825
			mob 070-5474825

Svar till samtliga uppgifter 1-15 ska skrivas på utdelat extra papper. Använd ett papper till uppgifterna 1-5, två papper till uppgifterna 6-10 och ett papper per uppgift till uppgifterna 11-15 Skriv din tentamenskod på varje inlämnat extra papper.

Den maximala poängen för respektive uppgift står angiven efter uppgiftens nummer. Totalt kan 40 poäng erhållas. För betyget 3 krävs ca 20, för betyget 4 ca 28 och för betyget 5 ca 34 poäng.

Om inget speciellt anges gäller frågorna Microsoft C++.

Detta häfte ska du behålla.

Lycka till!

1) (1p) I specifikationen för klassen VALUTA finns de privata medlemsvariablerna namn och kurs definierade enligt:

```
char namn[20];
float kurs;
```

Implementera konstruktorn nedan så att den kopierar parametervärdena till motsvarande medlemsvariabler. Specifikation av konstruktorn enligt:

```
VALUTA(char *namn, float kurs);
```

---

2) (1p) Klassen VALUTA har en medlemsfunktion, kostnad, som beräknar kostnaden för att köpa angivet belopp av önskad valuta. Specifikation enligt:

```
float kostnad(float belopp);
```

Skapa ett objekt av klassen VALUTA som har namn ”EUR”, kurs 9.53 och anropa funktionen kostnad och beräkna och skriv ut kostnaden för att växla beloppet 45.

---

3) (1p) Skapa istället ett dynamiskt VALUTA-objekt med samma data som i uppgift 2 ovan och beräkna och skriv ut kostnaden för samma växling som ovan. Glöm inte att avallokera objektet.

---

4)(1p) Implementera funktionen kostnad, i uppgift 2 ovan, som en medlemsfunktion i klassen VALUTA.

---

5) (1p) Skapa en vektor, valvek bestående av 5 VALUTA-objekt enligt tabellen nedan.

namn	kurs
AUD	5,73
DKK	1,25
CYP	16,27
EUR	9,32
INR	1,704

---

6) (2p) Klassen VALUTA, som vi redan använt, har specifikationen:

```
class VALUTA
{
   private:
      char namn[20];
      float kurs;
   public:
      VALUTA(char *namn = "", float kurs = 0.0);
      void las();
      void skriv();
      float kostnad(float belopp);
      void byt(VALUTA &v);
};
```

Implementera medlemsfunktionen las.

7) (2p) Implementera medlemsfunktionen skriv i klassen VALUTA i uppgift 6 ovan.

---

8) (2p) Implementera medlemsfunktionen byt i klassen VALUTA i uppgift 6 ovan.

---

9) (2p)Använd funktionen byt i uppgift 8 ovan för att byta värden mellan det första och sista elementet i vektorn valvek enligt uppgift 5 ovan.

---

10)(2p)Skriv ett huvudprogram som läser in namn och kurs för en valuta samt skapar ett objekt av klassen VALUTA . Programmet ska sedan fråga efter hur stort belopp som ska växlas och beräkna och skriva ut kostnaden för växlingen.

---

11)(5p)Klassen ELKOMP har en specifikation enligt:

```
// Specifikation av klassen ELKOMPONENT – Elkomp.h

class ELKOMP
{
   protected:
      char namn[30];          // Komponentens namn
      float effekt;           // Komponentens märkeffekt(maxeffekt)
   public:
      ELKOMP(char *namn = "", float effekt = 0.0);
      virtual void las();
      virtual void skriv();
      bool mindre_effekt(ELKOMP ek);
      // Sant om objekets effekt mindre än ek:s effekt
};
```

Implementera alla medlemsfunktioner i klassen ELKOMP samt skriv ett huvudprogram som skapar två elkomponenter och skriver ut dem med lägsta effekten först.

12(5p) Resistorer är elkomponenter och har specifikationen :

```
// Specifikation av klassen RESISTOR – Resistor.h

class RESISTOR : public ELKOMP
{
   private:
      float resistans;  // Resistorns resistans i ohm
   public:
      RESISTOR(char *namn="", float effekt=0.0, float resistans=0.0);
      void las();
      void skriv();
      float maxstrom();
      // Returnerar max tillåten ström som √effekt / resistans
};
```

Implementera alla medlemsfunktioner för klassen RESISTOR. och skriv ett huvudprogram som läser in ett RESISTOR-objekt och beräknar och skriver ut objektets maximala ström.

13)(5p)Klassen NOD avbildar punkter i en elektrisk krets och har specifikationen:

```
class NOD
{
   private:
      char namn[10];    // Nodens beteckning
      float potential;  // Nodens potential
      float summastrom; // Summan av ström in i noden som räknas
                        // positiv och ström ut från noden som räknas
                        // negativ
   public:
      NOD(char *namn = "", float potential = 0.0);
      float get_potential();
      void add_summastrom(float strom);// Öka summaströmmen med strom
      void reglera();
      // Reglerar potentialen så att Kichoffs första lag gäller. Är
      // summaströmmen > 0.0 ökas potentialen med 0.1 och är
      // summaströmmen < 0.0 minskas potentialen med 0.1.
};
```

Implementera alla medlemsfunktioner för klassen NOD samt skriv ett huvudprogram som skapar två NOD-objekt och placerar ett RESISTOR-objekt, enligt uppgift 12 ovan, mellan dessa. Det saknas något i RESISTOR-klassen för att dess objekt ska kunna känna till sin in- resp. ut-nod och samarbeta med dem. Lägg till det som saknas och fullborda programmet så att resistorn ber sina noder att skicka potentialen så att den kan räkna ut strömmen som potentialskillnaden dividerat med resistansen. Skriv ut den beräknade strömmen.

---

14)(5p)I textfilen Kurser.txt finns kurser inskrivna för ett stort antal valutor med valutans namn först och sedan kursen i kr enligt:

```
AUD   5.73
DKK   1.25
---   ----
```

Skriv ett program som frågar efter vilken valuta som ska växlas och som letar upp kursen för denna valuta i filen. Om kursen hittas ska programmet fortsätta med att fråga efter beloppet som ska växlas, räkna ut kostnaden för växlingen med hjälp av VALUTA-klassen i uppgift 6 ovan och skriva ut denna kostnad. Hittas ej den sökta valutan i filen ska detta skrivas ut i klartext.

---

15)(5p)Implementera operatorerna + och * för klassen RESISTOR i uppgift 12 ovan, så att + ger seriekoppling av två RESISTOR-objekt och * parallellkoppling. Vid seriekoppling ska resistansen bli summan av de två resistanserna, märkeffekten bli den lägsta av de två märkeffekterna multiplicerat med 2 och namnet bli ”R1+R2” om den första har namnet R1 och den andra R2. Vid parallellkoppling blir resistansen produkten dividerat med summan, märkeffekten samma som för seriekoppling och namnet ”R1*R2”. Skriv sedan ett huvudprogram som skapar och läser in tre RESISTOR-objekt där det första ska ligga i serie med de två övriga parallellt och skriv ut det resulterande RESISTOR-objektet.

Lösningar till tentamen i Objektorienterad Programmering C+, 5p

1)
```
VALUTA::VALUTA(char *namn, float kurs)
{
   strcpy(this->namn, namn);
   this->kurs = kurs;
}
```

2)
```
VALUTA v(”EUR”, 9.53);
cout << ”Kostnad : ” << v.kostnad (45) << endl;
```

3)
```
VALUTA *vp = new VALUTA(”EUR”, 9.53);
cout << ”Kostnad : ” << v->kostnad (45) << endl;
delete vp;
```

4)
```
float VALUTA::  kostnad(float belopp)
{
   return kurs * belopp;
}
```

5)
```
VALUTA valvek[]= {VALUTA(”AUD”, 5.73), VALUTA(”DKK”, 1.25),
                  VALUTA(”CYP”, 16.27), VALUTA(”EUR”, 9.32),
                  VALUTA(”INR”, 1.704)};
```

6)
```
void VALUTA :: las()
{
   cout << ”Ge valutanamn : ”;
   cin >> namn;
   cout << ”Ge valutakurs : ”;
   cin  >> kurs;
}
```

7)
```
void VALUTA :: skriv()
{
   cout << ”Valutanamn : ” << namn << endl;
   cout << ”Valutakurs : ” << kurs << endl;
}
```

8)
```
void VALUTA :: byt(VALUTA &v)
{
   VALUTA temp = v;

   v = *this;
   *this = temp;
}
```

9)
```
valvek[0].byt(valvek[4]);
```

10)
```
void main()
{
   VALUTA v;
   float belopp;

   v.las();
   cout << ”Ge belopp som ska växlas : ”;
   cin >> belopp;
   cout << ”Kostnad : ” <<  v.kostnad(belopp) << endl;
}
```

11)
```
#include <iostream>     // Utelämnas i uppgifterna nedan
#include <conio.h>
#include <cstring>
include ”Elkomp.h”
using namespace std;

ELKOMP:: ELKOMP(char *namn, float effekt)
{
   strcpy(this->namn, namn);
   this->effekt = effekt;
}

void ELKOMP::las()
{
   cout << ”Ge namn : ” ;
   cin   >> cin.getline(namn, sizeof(namn));
   cout << ”Ge märkeffekt : ”;
   cin >> effekt;
   cin.ignore();
}

void ELKOMP :: skriv()
{
   cout << ”Namn : ” << namn << endl;
   cout << ”Effekt : ” << effekt << endl;
}

bool mindre_effekt(ELKOMP ek)
{
   return effekt < ek.effekt);
}

void main()
{
    ELKOMP a, b;

    a.las();
    b.las();
    if (a.mindre_effekt(b))
    {
        a.skriv();
        b.skriv();
    }
    else
    {
        b.skriv()
        a.skriv():
    }
}
```

12)
```
RESISTOR::RESISTOR(char *namn, float effekt, float resistans)
                   : ELKOMP(namn, effekt)
{
   this->resistans = resistans;
}


void RESISTOR::las()
{
   ELKOMP::las();
   cout << ”Ge resistans : ”);
```

```
   cin >> resistans;
}

void RESISTOR::skriv()
{
   ELKOMP::skriv();
   cout << ”Resiatans : ” << resistans << endl;
}

float RESISTOR::maxstrom()
{
   return sqrt(effekt / resistans);
}

void main()
{
   RESISTOR a;

   a.las();
   cout << ”Maxström : ” << a.maxstrom() << endl;
}
```

13)

```
NOD::NOD(char *namn, float potential)
{
   strcpy(this->namn, namn);
   this->potential = potential;
   summastrom = 0.0;
}

void NOD::add_summastrom(float strom)
{
   summastrom += strom;
}

float NOD::get_potential()
{
   return potential;
}

void NOD::reglera()
{
   if (summastrom > 0.0)
      potential += 0.1;
   else if (summastrom < 0.0)
      poptential -= 0.1;
   summastrom = 0.0;
}

void main()
{
   // Här måste vi skapa 2 noder så att vi kan skicka deras adresser
   // till resistorerna som då kommer att känna till sin in-nod och
   // ut-nod. I klassen RESISTOR måste vi lägga till en funktion
   // get_resistans, som returnerar resistansen, medlemsvariablerna
   // NOD *in och NOD *ut och i konstruktorn lägga till två
   // parametrar i form av adresser till in- och ut-noden.

   NOD in(”N1”, 20), ut(”N2”, 10);
   RESISTOR r(”R1”, 1.0, 56, &in, &ut);
      cout << ”Strömmen blir : ” << (in->get_potential() -
                                    ut->get_potential()) /
                                    r.get_resistans();
}
```

14)
```
void main()
{
   ifstream tsin(”Valuta.txt”);
   float kurs, belopp;
   char sokvaluta[10] valutanamn[10]

   cout << ”Vilken valuta ska du köpa ? ”;
   cin.getline(sokvaluta, sizeof(sokvaluta));
   while((tsin>>valutanamn>>kurs)&&(strcmp(sokvaluta,valutanamn))!=0)
   {
   }
   if (strcmp(sokvaluta, valutanamn) == 0)
   {
      VALUTA v(valuta, kurs);
      cout << ”Hur mycket vill du växla ? ”;
      cin >> belopp;
      cout  << ”Kostnad : ” << v.kostnad(belopp) << endl;
   }
   else
      cout << ”Hittar ej kursen för den angivna valutan!”;
}
```

15)
```
RESISTOR RESISTOR::operator+(RESISTOR r)
{
   RESISTOR serie;

   serie.resistans = resistans + r.resistans;
   serie.effekt = 2*(effekt < r.effekt ? effekt:r.effekt);
   strcpy(serie.namn, strcat(strcat(namn, ”+”), r.namn));
   return serie;
}

RESISTOR RESISTOR::operator*(RESISTOR r)
{
   RESISTOR para;

   para.resistans = resistans*r.resistans/(resistans+r.resistans);
   para.effekt = 2*(effekt < r.effekt ? effekt:r.effekt);
   strcpy(para.namn, strcat(strcat(namn, ”*”), r.namn));
   return para;
}

void main()
{
   RESISTOR a, b , c, res;

   a.las();
   b.las();
   c.las();

   res = a + b*c;
   res.skriv();
}
```