Instuderingsfrågor inför prov 1 i Programmering 1

Prov 1 omfattar: Kursboken *Programmering 1 med C++*, kap 1-6: sid 9-167.

Övningar: Kap 1, sid 34-38. Kap 2, sid 50-51.
Kap 3, sid 67-68. Kap 4, sid 107-105.
Kap 5, sid 125-126. Kap 6, sid 160-164.

Sammanfattning av *C++ språkets egenskaper:* sid 33.

Svar till alla frågor finns i kursboken *Programmering 1 med C++*

Kap 1-2 Introduktion till programmering & Programmeringsmiljön, sid 9-51

1. Nämn några egenskaper av programmeringsspråket C++.
2. Vad innebär det att C++ är ett universellt programmeringsspråk?
3. Är C++ ett interpreterande eller ett kompilerande språk?
4. Är C++ källkod eller maskinkod?
5. Vad är fördelen med case sensitivity i C++?
6. Vilka två steg måste man gå för att se körresultatet av ett C++-program?
7. Beskriv skillnaden mellan kompilerings- och exekveringsfel.
8. Skriver man källkod eller maskinkod när man programmerar?
9. Vilken egenskap borde editorn ha i vilken man skriver programkoden?
10. Vad gör länkningen som ett steg mellan kompilering och exekvering?
11. Redovisa med egna ord de olika typer av fel som kan förekomma vid programmering.
12. Vad bestod den tekniska innovationen av i den datormodell John von Neumann utvecklade 1944?
13. Vilket var det första programmeringsspråk som utvecklades för de första datorerna?
    Vilka egenskaper hade det? Vad är dess största skillnad till dagens programmeringsspråk?
14. Vad karaktäriserar de programmeringsspråk som kallades för lågnivåspråk? Varför ”låg”?
15. Vilket var det första *högnivåspråket?* Varför *”hög”?*
16. Redogör för skillnaderna mellan begreppen *assemblering, kompilering och interpretering.*
17. Nämn ett exempel på programmeringsspråk som använde en av metoderna i frågan ovan.
18. Vad var det första användningsområdet för programmering?
19. Finns det fortfarande kod som används som är skriven i något av de första programmeringsspråken?
    Nämn några sådana samt deras användningsområde.
20. Vilket var det första programmeringsspråk som introducerade *kontrollstrukturer* i programmeringen?
21. Vad är den traditionella, procedurala synen på programmering som rådde på 60- och 70-talet?

22. Vad är den objektorienterade synen på programmering som kom upp på 80-talet?
23. Mellan vilka två programmeringsspråk går historiskt skiljelinjen mellan procedural och objektorienterad programmering? När ungefär inträffade övergången?
24. Vad var anledningen till paradigmskiftet inom programutveckling?
25. Vilka för- och nackdelar har enligt din åsikt den procedurala synen på programmering?
26. Vilka för- och nackdelar har enligt din åsikt den objektorienterade synen på programmering?
27. Definiera med egna ord begreppet algoritm. Nämn några exempel.
28. Vad är skillnaden mellan en algoritm och ett datorprogram?
29. Nämn några exempel på beskrivning av algoritmer. Vilka av dem är relevanta för programmering?
30. Vad är skillnaden mellan instruktioner och villkor i algoritmer?
31. Vilka är de tre grundläggande kontrollstrukturerna i alla procedurala programmeringsspråk?

Kap 3 Att komma igång med C++, sid 52-68

32. Vad är ett *program* I C/C++?
33. Är **`main()`** en funktion eller en klass?
34. Vad är skillnaden mellan en rad- och en blockkommentar, även kodmässigt?
35. Vad är skiljetecknet mellan två satser i C++?
36. Är **`int main()`** i programmet **`MyFirst`** ett anrop eller en definition av **`main()`**?
37. Går det att kompilera programmet **`MyFirst`** om man ersätter **`main()`** med **`Main()`**?
38. Har **`main()`** ett returvärde eller ej?
39. Kan man kompilera ett C++ program som innehåller följande sats? **`cout << 'ab';`**
    Motivera ditt svar.
40. Hur många operatorer finns i följande sats och vad kallas de? **`cout << "C" + "++";`**
41. Går det att kompilera satsen ovan? Om inte, vad kan man göra, för att få utskrivet **`C++`** ?
42. Är C++ ett standardardiserat programmeirngsspråk?
43. Är biblioteket i C++ standardardiserat?
44. Vad är ett **`namespace`** i C++?
45. Hur kan man i programmet **`MyFirst`** (sid 53) slippa skriva **`using namespace std;`**?
46. Nämn de mest grundläggande kriterierna för *God programmeringsstil*.
47. Med vilka stilelement i koden kan man uppnå kriterierna för God programmeringsstil?
48. Vad är radfortsättningstecknet i C++?

Kap 4 Grundbegrepp i programmering, sid 69-110

49. Vad är i C++ kod skillnaden mellan 8, '8' och "8"?
50. Varför har man olika datatyper i programmering?
51. Vad är en datatyp?
52. Vad menas med att C++ är ett strikt typbestämt språk (*strongly typed language*)?
53. Får **`a1`** väljas som variabelnamn i ett C++-program? Hur är det med **`a_1`** och **`1_a`**?
54. Vad är tanken bakom rekommendationen att välja beskrivande variabelnamn?
55. Vilka två steg måste tas i ett C++-program för att kunna använda variabler?
56. Vad menas med initieringen av en variabel?
57. Vad händer om man använder en variabel som man har deklarerat, men inte initierat?
58. Kan man inleda ett C++-program med följande sats? **`number = 5;`**
59. Vilken funktion kan man anropa i C++ för att läsa in data till programmet?
60. Vi vill läsa in ett heltal. Går det med följande satser? **`int x; cin << x;`**
61. Ge några exempel på operatorer i C++.
62. Ställ upp ett uttryck med modulooperatorn % och lös följande problem med uttrycket:

    Idag är torsdag och du vill träffa din kompis om 21 dagar.
    Vilken veckodag är det?

63. Vad gör följande satser? **`string name; cin >> name;`**

    Ge ett exempel på ett nästlat anrop av funktioner.

Kap 5 Enkla datatyper, sid 111-126

64. Vad menas med enkel datatyp? Vad är motsatsen till enkel datatyp?
65. Hur många enkla datatyper har vi i C++? Hur skiljer de sig från varandra?
66. Med vilken operator i C++ kan man mäta en datatyps minnesutrymme?
67. Kan man med C++-kod ta reda på de enkla datatypernas gränser?
68. Vilka regler finns för användning av variabler?
69. Med vilken C++-kod kan man ta reda på ett teckens ASCII-kod?
70. Med vilken C++-kod kan man ta reda på en ASCII-kods tecken?
71. Kan man med koden **`char letter = (char) 97;`** tilldela tecknet a till variabeln letter om 97 är ASCII-koden till tecknet a?
72. Kan man åstadkomma samma sak som i frågan ovan även enklare med koden **`char letter = 97;`**?
73. Vad blir b efter satsen ínt b = letter + letter; om man före denna sats har skrivit koden:

    `char letter = 'a';`

74. Förklara med egna ord vad som menas med begreppet Teckenaritmetik?
75. Varför har den standardiserade delen av ASCII-tabellen inte mer än 127 tecken?
76. Ge ett kodexempel på explicit typkonvertering.

77. Hur kan man med escapesekvenser generera i C++ kod de svenska specialtecknen ä, å, ö, Ä, Å, Ö?
78. Med vilken enkel datatyp representeras decimaltalskonstanter automatiskt?
79. Med vilken enkel datatyp representeras heltalskonstanter automatiskt?
80. Förklara med egna ord int-regeln vid automatisk typkonvertering.
81. Vad blir b efter satsen **`short b = s1 + s2;`** om man före denna sats har skrivit koden:

```
short s1 = 1;
short s2 = 2;
```

    Testa i ett litet program. Förklara resultatet. Rätta till koden om den är fel.

82. Vad blir **`b`** efter satsen **`long b = s1 + s2;`** om man före denna sats har skrivit koden:

```
short s1 = 1;
short s2 = 2;
```

    Testa i ett litet program. Förklara resultatet. Rätta till koden om den är fel.

83. Följande program kompileringsfel. Rätta till felet.

```
#include <iostream>
using namespace std;
int main()
{
   char letter;
   cout << "Tecknet " << letter << " har koden " << (int) letter;

   letter++;

   cout << "\nTecknet " << letter << " har koden " << letter + 1;
}
```

Kap 6 Kontrollstrukturer , sid 127-167

84. Med vilken sats skriver man i C++ ett enkelt val?
85. Med vilken sats skriver man i C++ ett tvåvägsval?
86. Med vilken sats kan man skriva i C++ ett flervägsval?
87. Skriv ett program som läser in tre heltal, hittar och skriver ut det största av dem. Modifiera programmet så att det blir det minsta av de tre talen. Använd endast enkelt val.

Titta på koden **`GissaTal_1`** i kursboken sid 128. Försök att besvara frågorna teoretiskt, annars testa koden:

88. Vad händer om man, när programmet `GissaTal_1` körs, matar in tal > 20 eller < 1?
89. Vad händer om man, när programmet `GissaTal_1` körs, matar in en bokstav istället för tal?
90. Val mellan hur många alternativ görs i programmet `GissaTal_1`?
91. Hur många villkor har `if-else`-satserna? Varför står efter `else` inget villkor?
92. Kan man i programmet `GissaTal_1` ersätta if-else-satserna med separata, enkla if-satser?
    I så fall gör det och testa!
93. Vad är nackdelarna med denna version av `Gissa tal`-spelet?

Titta på koden **`GuessDo`** i kursboken sid 132. Försök att besvara frågorna teoretiskt, annars testa koden:

94. Vad är den praktiska skillnaden mellan programmen `GissaTal_1` och `GissaTal_2`?
95. Kan man i koden se hur många varv `do`-loopen kommer att ha?
96. När avslutas `do`-loopen i `GissaTal_2`? Kan den bli en evighetsloop? Motivera ditt svar.
97. På vilket sätt styr villkoret i `do`-loopen programmet `GissaTal_2`?
98. Kan man i programmet `GissaTal_2` ersätta do-loopen med en while-loop? Om ja, gör det.
99. Efter du har testat Gissa tal både med `do`- och `while`-loopen, vilken av dem föredrar du och varför?
100. Vad är nackdelarna i Gissa tal-spelet, version 2?

Titta på koden **`GissaTal_3`** i kursboken sid 180. Försök att besvara frågorna teoretiskt, annars testa koden:

101. Vilken typ av tal slumpar funktionen `myRand(1, 21)` och i vilket intervall hamnar slumptalen?
102. Vilken typ av tal slumpar funktionen `myRand()` och i vilket intervall hamnar slumptalen? Ange det störst och det minst möjlia. Om du inte vet, testa!
103. Var någonstans är funktionen `myRand()` definierad? Hur ser man det i koden?
104. Val mellan hur många alternativ görs i varje varv av `do`-loopen i programmet `GissaTal_3`?
105. På vilket sätt realiserar programmet användarens ev. önskemål om att avbryta spelet och avslöja programmets hemliga tal?

Testa och experimentera gärna även i fortsättningen med programmen i din favorit IDE:

106. Skriv en `while`-sats som skriver ut de 50 första positiva heltalen.
107. Skriv en `while`-sats som beräknar och skriver ut summan 1 + 2 + 3 + … + 100.
108. Varför talar man i programmeringssammanhang om pseudoslumptal?
109. Vad är en evighetsloop och hur kan man förhindra en sådan?
110. Kan man i koden se hur många varv en `for`-loop kommer att ha?
111. Beskriv med egna ord `for`-satsens flödesplan.
112. Vad är den väsentliga skillnaden mellan en `for`- och en `while`- resp. do-sats?
113. Är `for`-satsens räknare giltig även efter `for`-satsen?
114. Initieras `for`-satsens räknare automatiskt? Vad händer om vi inte gör det?
115. Ser man i koden att `for`-satsens räknare uppdateras? Om ja, var någonstans?
116. Kan `for`-satsen bli en evighetsloop? Motivera ditt svar.
117. Skriv en `for`-sats som skriver ut de 50 första positiva heltalen.
118. Skriv en `for`-sats som beräknar och skriver ut summan 1 + 2 + 3 + … + 100.