Assignment 3
Hand-in date: Monday 21-09-2015 15:00h

3.1 Write an expression that obtains a random integer between **34** and **55**. Write an expression that obtains a random integer between **0** and **999**. Write an expression that obtains a random number between **5.5** and **55.5**.

3.2 Can the following conversions involving casting be allowed? If so, find the converted result.

```
char c = 'A';
int i = (int)c;

float f = 1000.34f;
int i = (int)f;

double d = 1000.34;
int i = (int)d;

int i = 97;
char c = (char)i;
```

3.3 Write the code that generates a random lowercase letter and explain what it does.

3.4 Let **s1** be " **Welcome** " and **s2** be " **welcome** " strings. Write the code for the following statements:

(a) Check whether **s1** is equal to **s2** and assign the result to a Boolean variable **isEqual**.

(b) Check whether **s1** is equal to **s2**, ignoring case, and assign the result to a Boolean variable **isEqual**.

(c) Compare **s1** with **s2** and assign the result to an **int** variable **x**.

(d) Compare **s1** with **s2**, ignoring case, and assign the result to an **int** variable **x**.

(e) Check whether **s1** has the prefix **AAA** and assign the result to a Boolean variable **b**.

(f) Check whether **s1** has the suffix **AAA** and assign the result to a Boolean variable **b**.

(g) Assign the length of **s1** to an **int** variable **x**.

(h) Assign the first character of **s1** to a **char** variable b.

(i) Create a new string **s3** that combines **s1** with **s2**.

Assignment 3
Hand-in date: Monday 21-09-2015 15:00h

(j) Create a substring of **s1** starting from index **1**.

(k) Create a substring of **s1** from index **1** to index **4**.

(l) Create a new string **s3** that converts **s1** to lowercase.

(m) Create a new string **s3** that converts **s1** to uppercase.

(n) Create a new string **s3** that trims whitespace characters on both ends of **s1**.

(o) Assign the index of the first occurrence of the character e in **s1** to an **int** variable **x**.

(p) Assign the index of the last occurrence of the string **abc** in **s1** to an **int** variable **x**.

3.5 Write a program that prompts the user to enter two strings and reports whether the second string is a substring of the first string.

3.6 Write a program that prompts the user to enter a Danish Social Security number (CPR) in the format **DDDDDD-DDDD**, where **D** is a digit. Your program should check whether the input is valid. Here are sample runs:

```
Enter a SSN: 211088-5435
211088-5435 is a valid Danish social security number

Enter a SSN: 23-23-5435
23-23-5435 is an invalid Danish social security number
```

Not mandatory part: you can implement additional checks if the entered digits can build a real CPR number. E.g.: if the first 6 digits can be date of birth or the last digit is even for girls and odd number for boys.

3.7 What is the output of the following code? Explain the reason.

```
int x = 80000000;
while (x > 0)
   x++;
System.out.println("x is " + x)
```

3.8 What is the output of the following code? Explain the reason.

```
for( ; ; ) {
   //Do something
}
```

Assignment 3
Hand-in date: Monday 21-09-2015 15:00h

3.9 Convert the following for loop statement to a while loop and to a do-while loop:

```
long sum = 0;
for (int i = 0; i <= 1000; i++)
sum = sum + i;
```

3.10 Write a program that reads an unspecified number of integers, determines how many positive and negative values have been read, and computes the total and average of the input values (not counting zeros). Your program ends with the input **0**. Display the average as a floating-point number. Here is a sample run:

```
Enter an integer, the input ends if it is 0: 1 2 -1 3 0
The number of positives is 3
The number of negatives is 1
The total is 5.0
The average is 1.25
```

3.11 Write a program that displays all the leap years, ten per line, from **101** to **2100**, separated by exactly one space. Also display the number of leap years in this period.