Noroff
School of technology
and digital media

Tuesday - Week 1

Primitives

Primitive Types

- boolean - “1 bit”
- byte - 8 bits
- char - 16 bits (Unicode)
- short - 16 bits
- int - 32 bits
- long - 64 bits
- float - 32 bits (decimal, in-precise)
- double - 64 (decimal, ‘precise’)

What do primitives do?

- The hold values
- Nothing else
- They are in lowercase
- Uppercase version are the associated classes (wrapper)
- The classes have the needed methods/functions
    - int & Integer
    - boolean & Boolean

Noroff
School of technology
and digital media

Tuesday - Week 1

Control Structures

Control Structures

- Things that branch code
    - Use parameters to decide direction
- Flow control

Types of Control Structures

- Sequential
  - In order
- Selection
  - Decisions/Branches
- Repetition
  - Multiples passes of the same code

Selection Types in Java

- if , if-else
- switch
- try-catch
- break*

Repetition Types in Java

- for
- while

if-then-else

```
if(condition){
  //code
}

if(condition){
//code
} else {
  //alternative code
}
```

switch

```
switch(expression) {
  case option1:
  // code for option 1
  break;

  case option2:
  // code for option 2
  break;
}
```

switch

```
switch(expression) {
  case option1:
  case option2:
  // code for option 1 & 2
  break;

  default:
  break;
}
```

try-catch

```
try{
  // some code that can cause an error
  } catch(SomeExceptionClass ex) {
  // handle the exception
  }
```

break

- Can be used to terminate a loop
- Not always the best idea

for

```
for(int i = 0; i < 10; i++){
  // do something ten times
  // each time i increases
  // starting from 0
  }
```

do-while

```
while(condition){
  // code that repeats
  // you will must manually end the loop
  // 1) change condition value
  // 2) use break;
}
```

Noroff
School of technology
and digital media

Tuesday - Week 1

Arrays, Lists, & ArrayLists

Collections - Counting from Zero

- Zero based indexing
  - The first item is indexed at zero

- Off-by-one error
  - Look out for “Index out of bounds”
  - java.lang.ArrayIndexOutOfBoundsException

Some Collections

- Arrays
- Lists
- ArrayLists

Array

- An array is the ‘primitive’ version of the collections

```
int nums[] = new int[10];
```

- Will need the Util library (as with most collections)

```
import java.util.Arrays;
```

- Once we switch to IntelliJ or Visual Studio Code they will handle imports like these ***automagically***

Lists

- In Java a List is a generic collection class
- Extended to form others:
    - ArrayList
    - LinkedList
    - Vector
    - Stack

Common List Methods

- Add / Remove
- Get / Set
- Search
- Iteration
- Subsetting

Generics* (A side note)

- It’s common to see an uppercase T in code examples
- T stands for Type
- As in a type of class
- When you use it in code you should replace it with a class that exists in your code base

```
List<T> myList = new List<T>();
// This a ‘generic’ list of type T
```

ArrayList

- This is the usual go to when you have a collection
- Has a plethora of methods
- Combines all the functionality of a List and and Array

Noroff
School of technology
and digital media

Tuesday - Week 1

Tasks

Task 3: Draw a Square

Write and compile a .java file which can be used to print a square of size chosen by the user at run-time.

The square can be made of any character that you choose. (# is probably a good choice)

The file must compile without errors.

example: (after compilation)
I use _ to show a space here (Yours should be blank)

```
$ java program 5
#####
#___#
#___#
#___#
#####
```

Task 4: Nested Rectangle

Write and compile a .java file which can be used to print a rectangle of size chosen by the user at run-time.
The rectangle should have an outer edge and a second inner edge.
There must be one space between in the inner and outer edge.

The rectangle can be made of any character that you choose. (# is probably a good choice)
You may choose the orientation yourself.

The file must compile without errors.

example: (after compilation)
I use _ to show a space here (Yours should be blank)

```
$ java program 7 8

#######
#_____#
#_###_#
#_#_#_#
#_#_#_#
#_###_#
#_____#
#######
```

Task 5: Name Search

- Write a hardcoded sample of 5 contact names
  - First and Last name
- Create a solution that uses one argument to search the list by name and determine if the name is in the list
  - Partial matches should work
  - Display all possible matches

Task 6: BMI Calculator

- Write a program that uses two arguments (args, not scanner):
    - Weight & Height
- It must calculate BMI and then categorise the result:
    - Underweight: BMI is less than 18.5
    - Normal weight: BMI is 18.5 to 24.9
    - Overweight: BMI is 25 to 29.9
    - Obese: BMI is 30 or more

$$\text{BMI} = \frac{(\text{weight in kilograms})}{\text{height in meters}^2}$$