# Project Ideas, Snow

## Main Project:

* Calculate snowmelt and snow cover for every month
* Input rasters: - new snow per month  
   - snow cover (raster from Snow detection code)
* Result rasters: - snow at the end of each month (🡪 should be transferred to next month)  
   - snowmelt per each month (🡪 erosive in this month)

## Code design:

* Load raster files and get the file names  
  🡪 maybe insert a function to check if they have all the same cell size, extent, no data value and so on  
  🡪 use filenames to identify year and month (similar to what we have done in the exercise)
* Code should create folders or check if they are already there to save the result and intermediate result files
* Code should do all the calculations (same raster format needed!)
* (Code should clip raster files)
* Code should write and save the raster files (.tif) using the right geoprojection
* Maybe we should use an dictionary with date as key
* Maybe some statistics for Sebastian like snow covered area in % or max snow height
* Maybe 3 Classes and one main script:   
  - Class for Raster Calculations (high priority, Theresa will start)  
  - Class for read, write data, create folders (high priority, I’ll start)  
  - Class to check input data (same format, geoprojections and so on)

Optional components:

* Implement GUI
* Add Plots
* Magic methods

### Maybe Additional:

* Connect with EroSnow and or Snow Detection
* Improve EroSnow according to what we have learnt in the course

### Open Questions

* Why to create methods/classes and not just use functions in our specific case?  
  🡪 What’s the advantage of classes in our case?  
  🡪 It makes sense with fish species, typical car examples and so on but for our project it’s a bit weird?
* Positional arguments are arrays and string but I don’t use ‘self.path = path’ at all and could delete it as well.
* Why is the code working with self alone but not e.g. with self.path

### Current Status

* Maybe use dictionary to have input files structured
* Lists are working perfectly fine if the files are loaded in the same order  
  🡪 current solution, can be adapted later