Notes:

• This is a take home exam. You should treat it as you would an invigilated exam, and answer the questions using your own knowledge and without assistance from anyone.

  However, you may refer to the course notes and you may run any programs you are asked to write.

• Use a text editor or standard word processor (such as MSWord or Open Office) to write your answers.

• You have a maximum of five hours to answer the exam. Allow sufficient time to submit your answers; do not leave this to the last minute!

• Upload your answers to the “exam” file area on the Student Portal, https://student portalen.uu.se. A plagiarism check will be made on all uploaded files.

• This exam has 14 questions. The questions are not of equal difficulty.

• Answer all questions. There are no optional questions.
Section A: Python

1. Give an example of how the makeColor function can be used to make a shade of brown. Hint: you can use the Picture tool (and a suitable image) to find the colour components.

2. Describe in your own words what the following Python code does:

```python
for p in getPixels(picture):
    if getX(p) > getWidth(picture)/2:
        setColor(p, makeDarker(getColor(p)))
```

3. If you use setPixel(picture, Xpos, Ypos) to change a pixel outside the bounds of the picture, Python will give an error. Write a functions setPixel2 that will first check whether Xpos and Ypos are within the bounds of the picture before calling setPixel. If the position is outside the picture, your function should do nothing.

4. Write a Python program to increase the redness of red pixels, the blueness of blue pixels, and the greenness of green pixels. A “red” pixel is defined as any pixel that has more red than either green or blue. Similarly for “green” and “blue” pixels. The increase should be by a proportion of the current colour. For example, the image below on the right shows this transformation for an increase in contrast of 120%.

Section B: Prolog

5. Give an example of a Prolog term that could be used to describe an appointment, where an appointment has: a description; a date and time; a duration; and a location.

6. Some rules for selecting a programming language to use for writing an application program are:
• Python is good for image processing;
• Prolog is good for natural language processing, but not good for number crunching;
• Erlang is good for communications, and also for parallel operations.

Assume you have a set of Prolog facts that describe the requirements of the application program. For example:

```
requires( image_processing ).
requires( number_crunching ).
requires( minimum_wage_programmers ).
requires( pretty_interface ).
```

Write a Prolog predicate, `suitable(Lang)` that expresses these rules for selecting a programming language. The body of each rule should test one or more of the `requires` facts.

7. Use Prolog grammar rules to write a parser for a language of cooking instructions, where the instructions are: “use \text{`\langle container\rangle`}”, “add \text{`\langle ingredients\rangle`}”, and “\text{`\langle process\rangle`}”. \text{`\langle container\rangle`} is either “bowl” or “pan”. \text{`\langle ingredients\rangle`} is either a single \text{`\langle ingredient\rangle`} or “\text{`\langle ingredient\rangle` and `\langle ingredients\rangle\rangle`". \text{`\langle ingredient\rangle`} can be “egg”, “butter”, “sugar”, “milk”, or “flour”. \text{`\langle process\rangle`} is either “whisk”, “stir”, or “\text{`\langle cook\langle temperature\rangle\langle duration\rangle\rangle`}". \text{`\langle temperature\rangle`} and \text{`\langle duration\rangle`} are both numbers. Any instruction can be followed by “and then \text{`\langle instructions\rangle\rangle".}

For example, your parser should be able to recognise instructions such as these:

- \text{`[cook,190,25]`}
- \text{`[use.bowl.and.then.add.eggs\]`}
- \text{`[use.bowl.and.then.add.eggs.and.butter.and.sugar.and.then.whisk\]`}

**Section C: Erlang**

8. Write an Erlang pattern that will result in the variable \text{`X`} having the value \text{`ok`} when matched against the tuple \text{`{not.so,\{fast,\{ok\}\}}\), but will fail to match against any other tuple.

9. Write an Erlang tuple for which the pattern \text{`{draw,\{\_Height,\_Width\}\}`} will bind \text{`Height`} and \text{`Width`} to 10 and 40 respectively.

10. Give an example of an Erlang term that could be used to describe an appointment. An appointment has a description, a date and time, a duration, and a location.

11. There is a webcam overlooking the Domkyrka in Uppsala. The camera can be instructed to move horizontally and vertically, and to zoom in or out. An Erlang program is to be used to coordinate access to the camera controls from users from all around the world. A user can gain control of the camera by sending the webcam server a “grab” message, which must include its own process identifier. Once the camera has been grabbed, the server will start to listen for messages sent by that process (ignoring other messages). These messages can instruct the server to move or zoom the camera. If no message is received from the user after one minute, the server will resume listening for a new grab message.
Write two Erlang functions: `webcam_server` that listens for a grab message, and `webcam_server(Pid)` that listens for camera control messages from `Pid`. You can assume the existence of functions `camera:zoom`, `camera:rise` and `camera:shift` that control the actual movements of the webcam.

**Section D: General**

12. In three or four sentences, explain the most important differences between variables in Python, Prolog and Erlang. Do not focus on syntactic differences (i.e. the appearance of the variables), but rather on the different ways variables behave and can be used.

13. In three or four sentences, describe the most interesting or unexpected thing you encountered in doing one of the labs or assignments.

14. In 200–300 words, describe what you plan to do in your independent project. State which language you have chosen.