The PROGRAMMING PITFALLS Lesson

Co-funded by the Erasmus+ Programme of the European Union

Table of Contents

1. Instructor Guide for Programming Pitfalls

Welcome to the instructor guide for Programming Pitfalls, looking at ethical issues around programing. where we have all our advice and suggestions as to how to deliver lecturers on this topic. Our key belief in developing this content is that you know how best to deliver the important information; we can supply the content to you, but you will find the best way to teach it, in a manner than suits your teaching style. And because of that, we have designed the content to be as flexible as possible, with multiple sets of slides and documents that can be put together in multiple ways, depending on what suits you.

This current document provides a range of content, including a series of key outcomes for this lesson, and some details on the lecture notes and how they explore the key topics in this lesson. Following this a sample exam question and answer are provided as a suggestion as to how this lesson could be turned into part of an examination.

The final section looks at assessments, and clearly the type of assessment and the level of complexity will depend on the stage at which this content is being delivered at, as well as the other content that is being delivered at the same time. You are the lecturer who is delivering this content, so you are the expert at what the best way to assess the students, but included are some interesting alternatives as to what you could do. Additionally, some in-class activities are included that you can do in class, and take home.

2. Learning Outcomes for Programming Pitfalls

On completion of this object, the learner will be able to:

A. Demonstrate a clear understanding of the concepts and models associated with programming errors.

B. Critically assess and evaluate programs with potential for data bias.

C. Review and assess relevant literature, incorporating legislation, policy, directives, academic journals and industry standards.

D. Relate concepts associated with ethics to the development and evaluation of programming error.

E. Select and evaluate models of good programming practice that should be used to prevent programming errors.

F. Compare and contrast how different types of errors and why they occur.

3. Assessment of Programming Pitfalls

There is a range of ways to assess this topic, and below are some suggested approaches for examination papers, for assessments, for in-class activities, and take-home activities.

3.1. Exam Questions

The below question can be used as a single question in an exam paper.

3.1.1. Sample Exam Question

This is out of 100 marks.

1 (a) Explain what the terms *Debugging* and *Testing* means.
(20 marks)
1(b) Describe in detail two programming pitfalls, including the following sections: • A clear overview of the pitfall • A description of the consequences of the pitfall • Some reflections on the pitfall
(40 marks)
1(c) "*Software cannot be good, fast and cheap*", explain what this means and its implications for ethics.
(40 marks)

Potential Solutions:

1 (a) What is meant by the term "Dark Patterns", providing a clear definition of your understanding of the term.

```
Debugging is the process of locating and fixing errors in
computer programs (which are sometimes called "bugs") when
they occur. Testing is the process of checking the quality
of software to help ensure that the programs are working
well, and it might also uncover bugs.

"locating and fixing": 5 marks
"errors are called bugs": 5 marks
"quality of software": 5 marks
"It might uncover bugs": 5 marks
```

(20 marks)

1(b) Describe in detail two programming pitfalls, including the following sections:

- A clear overview of the pitfall
- A description of the consequences of the pitfall
- Some reflections on the pitfall

```
20 marks per pitfall

Overview of pitfall: 6 marks
Consequences of pitfall: 8 marks
Reflections of pitfall: 4 marks
```

(40 marks)

1(c) "*Software cannot be good, fast and cheap*", explain what this means and its implications for ethics.

(40 marks)

```
If software is needed fast and needs to be good, it won't be
cheap, if software is needed cheap and needs to be good, it
won't be fast, and if software is needed fast and needs to
be cheap, it won't be good. 20 marks.

Computer companies need to spend as much time testing their
software as much as programming time. If the software is
developed in a rush, it needs to be extensively tested
before release. For software that could impact people's
lives, the software needs to be extensively tested. 20
marks.
```

(40 marks)

3.2. Continuous Assessments

3.2.1. Create an Infographic or Poster

Assignment #1. Programming Ethics Infographics
Create an infographic or poster based on programming pitfalls:

- It can be an overview of the entire discipline (must be detailed and pretty)
- It can be an infographic of a specific pitfall
- It can be an infographic of how the relationship between ethics and the pitfalls.
- etc.

Accompanying this infographic is going to be a 3-4 page document of the following format (this is only approximate, you can write more if your want):

- Page 1: Cover Page
- Page 2: First half, introduction to what you did
- Page 2-3: Second half of page 2 and all of page 3, description of how you created the infographic with rationale for all the elements present, both format and content.
- Page 4: Conclusions, reflections and Recap on what you did.

3.2.2. Create a Research Paper

Assignment #2. Programming Pitfalls Research Paper
The goal of the assignment is to write a paper about programming pitfalls, it should be made up of two main parts, a literature review section, and experiment section (with a conclusion and references).

The Literature Review Section
This section must explore at least three papers on the topic of programming errors, one of which must be the Hovemeyer and Pugh (2004) seminal paper:

- `Hovemeyer, D. and Pugh, W., 2004. "Finding Bugs is Easy". ACM SIGPLAN Notices, 39(12), pp.92-106.`

For each paper, provide a 300-400 word summary of the paper, including an introduction to the paper, a description of the experiment in the paper, and mention some of the key conclusions of the paper. Select one diagram from the paper to include in your review.

The Experiment Section
This section will document your own exploration of a software that has errors in it, including screengrabs and a description of the context.

Conclusions Section
This section will be 300-500 word conclusion of the paper.

References
All references in Harvard format.

3.2.3. Create an Ethics Essay

Assignment #3. Programming Pitfalls Ethics
Write an essay on the ethics of programming errors. This will involve a bit of reading and research on your behalf, looking at research papers, and other content online like newspaper articles on programming, and websites concerning ethics and programming. An essay usually has three parts:

- Introduction: An overview of the essay highlighting the key arguments
- Main Body: Addressing the key issues and arguments, review your key readings, and critique them, and present your arguments, and critique yourself.
- Conclusions: Your conclusions about this whole area.

The typical structure of an argument is:
1. Claim, Outline the main claim you are making, sometimes called the overall thesis
2. Grounds, Describe the evidence and facts that support your claim, best evidence first
3. Bridge, Explain and underscore how the Grounds supports your Claim
4. Backing, Add any additional logic or reasoning that support the Bridge
5. Counterclaim, Discuss the alternative perspectives that oppose your thesis
6. Rebuttal, Identify the weaknesses in the Counterclaim and present evidence that refutes it.

As part of the process of writing this essay you will have to draft and redraft this essay as part of this process, and it will help you in developing a formal, academic "voice"

3.2.4. Create an Interview

Assignment #4. Programming Pitfalls Interviews
Document an interview with someone who has been a victim of a programming pitfall. To help the person know what a programming pitfall is, write a 500-700 description of pitfalls, including some key examples (include this summary as part of your submission. For the main interview, ask the following questions, but put them in your own words:

- Describe what you were doing when this occurred.
- When did you discover you had an issue?
- How did it make you feel?
- Were you able to do anything about it?
- Did it make you change your interaction with software afterwards?
- Do you think there should be laws stopping companies doing this?

Write a 500-700 word reflection piece on the interview answers.

Marking Scheme

- Programming pitfalls summary – 20%
- Interview Questions – 6 * 10%
- Reflections – 20%

3.3. In-Class Activities

3.3.1. The Worst Programming Ethics Case

This activity works best for classes with more than 20 students.

Reviewing the Ethical cases you have discussed in class, get the students to select which one they think is the most unethical, and you can ask everyone to call out the number of their choice, and count which ones get the more votes, and which don't.

What is interesting is to see how different people define "unethical", some might see it as which has the worst impact on the user, others might see it as the one that is most concealed, others as the one the is most technically complex, others might see it as the one that breaches progrmaming principles, etc.

3.3.2. Rate the Pitfalls

This activity works best for classes with less than 20 students.

Get the students to rate the pitfalls on a scale of 1 – 12 from least unethical to most unethical. Once they have settled on their ordering, compare their answers to a classmate.

What is interesting is to see how different people define "unethical", some might see it as which has the worst impact on the user, others might see it as the one that is most concealed, others as the one the is most technically complex, others might see it as the one that breaches programming principles, etc.

3.3.3. Discussion Cards

Below are a series of cards that you can print out and use in the classroom in a variety of ways. You may wish to divide the class into groups, and have each group work on the same discussion card; or have each group work on different cards.

3.3.4. Bringing it all together

Towards the end of the lecture, ask each of the students to write for one minute about a specific question, it could be something general like "what was the most important thing you have learned today", or something more specific like "programing ethics are …". Then put the students in groups, and get them to share their answers with each other, and finally appoint a leader in each group to share the range of views from their members with everyone else in the class.

3.4. Take-Home Activities

3.4.1. Paying Attention

Tell the students for the rest of the semester, each time you are interacting with software, note any time you encounter any issues, and note the link (if appropriate), do a screengrab, and write a 50-word description.