COSC264 Assignment 1

@tfinlay
ID: 12345678

bundler.py

```
from reportlab.pdfgen import canvas
from reportlab.lib.pagesizes import A4, landscape
from reportlab.platypus import Paragraph, SimpleDocTemplate, Spacer, PageBreak
from reportlab.lib.styles import ParagraphStyle
from reportlab.platypus.xpreformatted import PythonPreformatted, XPreformatted
from reportlab.lib.styles import getSampleStyleSheet
from reportlab.lib.units import inch
from pygments import highlight
from pygments.lexers import PythonLexer
import PyPDF2
import os
import textwrap

A4_WIDTH, A4_HEIGHT = A4


"""Adapted from: https://www.reportlab.com/snippets/11/"""
def _2xpre(s,styles):
    "Helper to transform Pygments HTML output to ReportLab markup"
    s = s.replace('<span></span>', '')

    s = s.replace('<span>', '<font>')
    s = s.replace('</span>','</font>')
    s = s.replace('<div class="highlight">','')
    s = s.replace('</div>','')
    s = s.replace('<pre>','')
    s = s.replace('</pre>','')
    for k,c in styles+[('p','#000000'),('n','#000000')]:
        s = s.replace('<span class="{}">'.format(k),'<font color="{}">'.format(c))
    return s


def pygments2xpre(s):
    "Return markup suitable for XPreformatted"
    try:
        from pygments import highlight
        from pygments.lexers import PythonLexer
        from pygments.formatters import HtmlFormatter
    except ImportError:
        return s

    l = PythonLexer()
    h = HtmlFormatter()
    from io import StringIO
    out = StringIO()
    highlight(s,l,h,out)
    styles = [(cls, style.split(';')[0].split(':')[1].strip())
```

```
                for cls, (style, ttype, level) in h.class2style.items()
                if cls and style and style.startswith('color:')]
    return _2xpre(out.getvalue(),styles)


def get_file_name(prompt: str, accept_empty: bool = False):
    filename = input(prompt)
    while not os.path.isfile(filename):
        if accept_empty and filename == '':
            return None
        else:
            print("Could not find file at path: {}".format(filename))
            filename = input(prompt)
    return filename


def add_coversheet(story):
    title = input("Document title (optional): ")
    if title:
        styles = getSampleStyleSheet()
        story.append(Paragraph(title, styles["Title"]))

    coversheet_style = ParagraphStyle(
        'heading',
        fontName='Helvetica',
        fontSize=20,
        leading=22
    )
    story.append(Paragraph(input("Student Name: "), coversheet_style))
    story.append(Paragraph("ID: {}".format(input("Student ID: ")), coversheet_style))
    story.append(PageBreak())


def add_code(story):
    filename = get_file_name("Code file: ")
    display_name = input("Code file display name [{}]: ".format(filename))

    while filename is not None:
        story.append(Paragraph(display_name if display_name else filename, styles['Heading3']))
        with open(filename, "r") as f:
            formatted = pygments2xpre(f.read())
            story.append(XPreformatted(
                formatted,
                styles['Code']
            ))
        story.append(PageBreak())

        filename = get_file_name("Code file (press return to finish): ", True)
        if filename is not None:
```

```
            display_name = input("Code file display name [{}]: ".format(filename))

if __name__ == '__main__':
    styles = getSampleStyleSheet()

    doc = SimpleDocTemplate("out.pdf.tmp", pagesize=landscape(A4), margin=(0,0))
    story = []

    add_coversheet(story)
    add_code(story)

    # Write out
    doc.build(story)

    # Open in PyPDF2
    merger = PyPDF2.PdfFileMerger()
    with open("out.pdf.tmp", 'rb') as f:
        merger.append(f)
        with open(get_file_name("Declaration form PDF path: "), "rb") as f:
            merger.append(f)
            with open("out.pdf", "wb") as f:
                merger.write(f)

    merger.close()
    os.unlink("out.pdf.tmp")
```

Plagiarism Declaration

This form needs to accompany your COSC 264 assignment submission.

I understand that plagiarism means taking someone else's work (text, program code, ideas, concepts) and presenting them as my own, without proper attribution. Taking someone else's work can include verbatim copying of text, figures/images, or program code, or it can refer to the extensive use of someone else's original ideas, algorithms or concepts.

I hereby declare that:

- My assignment is my own original work. I have not reproduced or modified code, figures/images, or writings of others without proper attribution. I have not used original ideas and concepts of others and presented them as my own.
- I have not allowed others to copy or modify my own code, figures/images, or writings. I have not allowed others to use original ideas and concepts of mine and present them as their own.
- I accept that plagiarism can lead to consequences, which can include partial or total loss of marks, no grade being awarded and other serious consequences, including notification of the University Proctor.

Name: ……

Student ID: ……

Signature: ……

Date: ……