

# PBK

**Author:** Eugene A. Fedotov, Matthew Douglass, Jovaughn Chin

**Version:** 0.1

**Licence:** GNU Free Documentation License

The PBK package aims to be a Portable Kit for open-source Books. It is meant to provide a diverse selection of common graphical diagrams in select subjects. Currently, the focus is on compilers and programming in Java.

## Dependencies

The following libraries are required to run PBK:

- [PGF 3.0](#)

## Libraries

### Agenda (subject to change)

#### 0.1 Sprint 1 : 3/9 - 3/13

1. Decide on a set of diagrams for each person.
2. Pick one diagram. Create a hard (specific with no customization) example that illustrates the exact look.

#### 0.2 Sprint 2 : 3/16 - 3/20

1. Consider what possible options can be implemented.
2. Refactor the code to implement those options.

#### 0.3 Sprint 3 : 3/23 - 3/27

1. Check for issues. Scalability? Color?
2. Add the function to the package.
3. Update doc.

#### 0.4 Sprint 3 : 3/30 - 4/3

1. Work on the next diagram.

#### 0.5 Sprint 4 : 4/6 - 4/9

1. TBD

Arrays

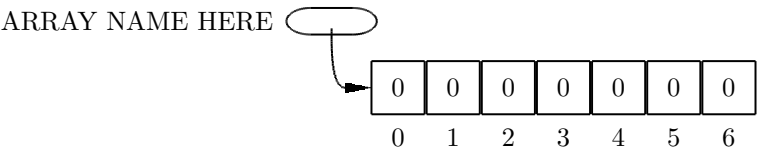


Figure 1: CAPTION HERE

With Arrays the user will be able to input the information for the array name, caption name, length of the array and the actual values of the array.

Expressions

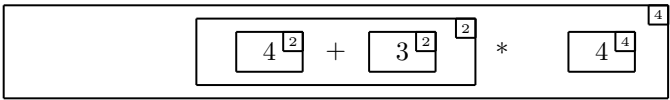


Figure 2: CAPTION HERE

With Expressions the user will be able to edit the caption name and all of the values.

Classes

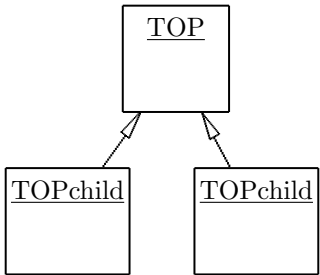


Figure 3: CAPTION HERE

The classes diagrams will be a tree of parents and their children. The name of each class will be editabnle as well as the shape of the tree and the caption name.

## Objects

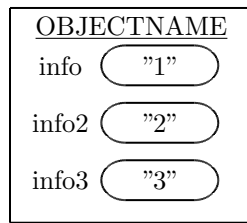


Figure 4: CAPTION HERE

The object diagrams will display a single object with all of its values accepting input. The caption name will be editable as well as the objects name.

## 1 User Input

Users will be able to create diagrams by editing a section near the top of the latex code. This section will be clearly defined, easily editable and understandable. Below is a work in progress of some latex code that the user will have to edit to create a diagram that fits their specs. The example below is specifically used when making array diagrams and is far from final.

```

User Input Below
\newcommand{\ArrayName}
{ARRAY NAME HERE}
\newcommand{\Caption}
{CAPTION HERE}
\newcommand{\ArrayLengthMinusOne}
{ARRAY LENGTH HERE}
\newcommand{\0}
{PLACE 0 VALUE HERE}
\newcommand{\1}
{PLACE 1 VALUE HERE}
\newcommand{\2}
{PLACE 2 VALUE HERE}
\newcommand{\3}
{PLACE 3 VALUE HERE}
\newcommand{\4}
{PLACE 4 VALUE HERE}
User Input Above

```

The user will find the location in the template that clearly states "User Input Below" and edit below that until they come to "User Input Above". The user will edit only the statements that are in CAPS being careful not to edit anything besides that. For example if the user wanted to create an array diagram that holds 4 values they would look for "ARRAY LENGTH HERE". After finding this they would replace the text in CAPS with an integer. The user would then compile the LATEX code to find a array diagram to the users specs. Above the user input area there will be information regarding what inputs are acceptable for the template.

**Change History**

v0.1  
General: Initial version .....1