Terminal Editor

//manages programming assignments. Interns will upload assignments and converted to a zip file. Interns are also able to create assignments on the editor.

```
   /**
     * Return the programming language name based on submitted files extensions
     */
    public function get_pln($filelist) { //filelist array
        foreach ($filelist as $checkfilename) {
            $ext = pathinfo( $checkfilename, PATHINFO_EXTENSION );
            if (isset( self::$languageext [$ext] )) {
                return self::$languageext [$ext];
            }
        }
        return 'default'; //return programming language
    }
```

```
/**
 * open or create a file
*/
function vpl_fopen($filename) {
    global $CFG;

    if (! file_exists( $filename )) { // Exists file?
        $dir = dirname( $filename );
        if (! file_exists( $dir )) { // Create directory?
            if (! mkdir( $dir, $CFG->directorypermissions, true ) ) {
                throw new file_exception('storedfileproblem', 'Error creating a directory to save
files in VPL');
            }
        }
    }
    $fp = fopen( $filename, 'wb+' ); //open file
    if ($fp === false) {
        throw new file_exception('storedfileproblem', 'Error creating file in VPL'); //error
    }
    return $fp;
}
```

```
/**
 * Outputs a zip file and removes it.
 */
function vpl_output_zip($zipfilename, $name) { //zipfilename name of the file with the data
    if (! file_exists($zipfilename)) {
        print_error("Zip file not found");
        die;
    }
```

```
    // Send zipdata.
    $blocksize = 1000 * 1024;
    $size = filesize( $zipfilename );
    $cname = rawurlencode( $name . '.zip' ); $name of file to be shown, without '.zip'
    $contentdisposition = 'Content-Disposition: attachment;';
    $contentdisposition .= ' filename="' . $name . '.zip";';
    $contentdisposition .= ' filename*=utf-8\'\'' . $cname;
    // Get zip data.
    $offset = 0;
    while ($offset < $size) {
        echo file_get_contents( $zipfilename, false,  null, $offset, $blocksize);
        $offset += $blocksize;
    }
    // Remove zip file.
    unlink( $zipfilename );
}
```

Feedback

//implement a feedback for assignments so that the manager is able to give feedback upon grading.

```
// Add a feedback option within form.
        $feedbacks = array();
        if ($gradeitems = $this->_customdata['gradeitems']) {
            foreach ($gradeitems as $itemid => $itemname) {
                $feedbacks['feedback_'.$itemid] = get_string('feedbackforgradeitems', 'grades',
$itemname);
            }
        }
```

Progress

//The intern sees a progress donut that is displayed on the dashboard.

```
/**
 * Course completion status - Displays overall, and individual criteria status for logged in user.
 */
class block_completionstatus extends block_base {

    public function init() {
        $this->title = get_string('pluginname', 'block_completionstatus');
    }

    public function applicable_formats() {
        return array('course' => true);
    }

    public function get_content() {
        global $USER;
        // Get course completion data.
        $info = new completion_info($course);
```

```
        // Don't display if completion isn't enabled!
        if (!completion_info::is_enabled_for_site()) {
            if ($can_edit) {
                $this->content->text .= get_string('completionnotenabledforsite', 'completion');
            }
            return $this->content;

        } else if (!$info->is_enabled()) {
            if ($can_edit) {
                $this->content->text .= get_string('completionnotenabledforcourse', 'completion');
            }
            return $this->content;
        }

        // Load criteria to display.
        $completions = $info->get_completions($USER->id);

        // Check if this course has any criteria.
        if (empty($completions)) {
            if ($can_edit) {
                $this->content->text .= get_string('nocriteriaset', 'completion');
            }
            return $this->content;
        }

        // Check this user is enroled.
        if ($info->is_tracked_user($USER->id)) {

            // Generate markup for criteria statuses.
            $data = '';

            // For aggregating activity completion.
            $activities = array();
            $activities_complete = 0;

            // For aggregating course prerequisites.
            $prerequisites = array();
            $prerequisites_complete = 0;

            // Flag to set if current completion data is inconsistent with what is stored in the
database.
            $pending_update = false;

            // Loop through course criteria.
            foreach ($completions as $completion) {
                $criteria = $completion->get_criteria();
                $complete = $completion->is_complete();

                if (!$pending_update && $criteria->is_pending($completion)) {
                    $pending_update = true;
                }
```

```
            // Activities are a special case, so cache them and leave them till last.
            if ($criteria->criteriatype == COMPLETION_CRITERIA_TYPE_ACTIVITY) {
                $activities[$criteria->moduleinstance] = $complete;

                if ($complete) {
                    $activities_complete++;
                }

                continue;
            }

            // Prerequisites are also a special case, so cache them and leave them till last.
            if ($criteria->criteriatype == COMPLETION_CRITERIA_TYPE_COURSE) {
                $prerequisites[$criteria->courseinstance] = $complete;

                if ($complete) {
                    $prerequisites_complete++;
                }

                continue;
            }
            $row = new html_table_row();
            $row->cells[0] = new html_table_cell($criteria->get_title());
            $row->cells[1] = new html_table_cell($completion->get_status());
            $row->cells[1]->style = 'text-align: right;';
            $srows[] = $row;
        }

        // Display completion status.
        $table = new html_table();
        $table->width = '100%';
        $table->attributes = array('style'=>'font-size: 90%;', 'class'=>'');

        $row = new html_table_row();
        $content = html_writer::tag('b', get_string('status').': ');

        // Is course complete?
        $coursecomplete = $info->is_course_complete($USER->id);

        // Load course completion.
        $params = array(
            'userid' => $USER->id,
            'course' => $course->id
        );
        $ccompletion = new completion_completion($params);
```