

SCIFF 009 – FUNDAMENTALS OF PROGRAMMING

Group Assignment (MATLAB)

STUDENT NAME(S)	STUDENT ID	CONTACT NUMBER
Yeo Yi Xin	20414841	011-2042 2011
Elena Dubau Anak Preston	20409902	013-210 6262
Chuah Jia En	20410604	012-986 0276
Chiew Cui Xuan	20409671	016-504 2929
Lee Yen Yee	20410791	017-635 0693

Table of Content

No.	Content	Page
1.	GUI layout	3
2.	M-scripts	4-18
3.	Explanation for additional features / functions.	19-21
4.	Testing for each scenario	22-26
5.	Testing for GUI menus	27-35
6.	Testing outputs	36-44
7.	Marking Rubric	45

The Graphic User Interface of the Assignment

File Edit Help

Loan Repayment Table:

Loan Amount:

Yearly Interest Rate:

Monthly Payment:

Loan Repayment

Heading

M-Script

```
function varargout = assignment(varargin)
% ASSIGNMENT MATLAB code for assignment.fig
%     ASSIGNMENT, by itself, creates a new ASSIGNMENT or raises the existing
%     singleton*.
%
%     H = ASSIGNMENT returns the handle to a new ASSIGNMENT or the handle to
%     the existing singleton*.
%
%     ASSIGNMENT('CALLBACK',hObject,eventData,handles,...) calls the local
%     function named CALLBACK in ASSIGNMENT.M with the given input arguments.
%
%     ASSIGNMENT('Property','Value',...) creates a new ASSIGNMENT or raises the
%     existing singleton*. Starting from the left, property value pairs are
%     applied to the GUI before assignment_OpeningFcn gets called. An
%     unrecognized property name or invalid value makes property application
%     stop. All inputs are passed to assignment_OpeningFcn via varargin.
%
%     *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%     instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help assignment

% Last Modified by GUIDE v2.5 27-Mar-2022 19:47:29

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @assignment_OpeningFcn, ...
                  'gui_OutputFcn',  @assignment_OutputFcn, ...
                  'gui_LayoutFcn',  [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before assignment is made visible.
function assignment_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to assignment (see VARARGIN)

% Choose default command line output for assignment
```

```

handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes assignment wait for user response (see UIRESUME)
% uiwait(handles.result_table);

% --- Outputs from this function are returned to the command line.
function varargout = assignment_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function loan_amount_Callback(hObject, eventdata, handles)
% hObject handle to loan_amount (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of loan_amount as text
% str2double(get(hObject,'String')) returns contents of loan_amount as a
double
loan_amount=get(hObject, 'string');
loan_amount_num=str2double(loan_amount);
handles.loan_amount_num=loan_amount_num;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function loan_amount_CreateFcn(hObject, eventdata, handles)
% hObject handle to loan_amount (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
% See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUiControlBackgroundColor'))
set(hObject,'BackgroundColor','white');
end

function yearly_interest_Callback(hObject, eventdata, handles)
% hObject handle to yearly_interest (see GCBO)
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of yearly_interest as text
% str2double(get(hObject,'String')) returns contents of yearly_interest
as a double

```

```

yearly_interest=get(hObject, 'string');
yearly_interest_num=str2double(yearly_interest);
handles.yearly_interest_num=yearly_interest_num;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function yearly_interest_CreateFcn(hObject, eventdata, handles)
% hObject    handle to yearly_interest (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function monthly_payment_Callback(hObject, eventdata, handles)
% hObject    handle to monthly_payment (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of monthly_payment as text
%         str2double(get(hObject,'String')) returns contents of monthly_payment
%         as a double
monthly_payment=get(hObject, 'string');
monthly_payment_num=str2double(monthly_payment);
handles.monthly_payment_num=monthly_payment_num;
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function monthly_payment_CreateFcn(hObject, eventdata, handles)
% hObject    handle to monthly_payment (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in PB_calculate.
function PB_calculate_Callback(hObject, eventdata, handles)
% hObject    handle to PB_calculate (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
loan_amount_num=handles.loan_amount_num;
yearly_interest_num=handles.yearly_interest_num;
monthly_payment_num=handles.monthly_payment_num;

% break if stuck in infinite loop

```

```

a=0;
% break if condition is met
breaker=0;
% initialize the output.
output_string = '';

% fixed monthly interest rate
monthly_interest_num=(yearly_interest_num/100)/12;
% (2/100) / 12 = 0.00167

%calculate the interest_num for the first month
interest_num=loan_amount_num*monthly_interest_num;

%if user enter negative value
if (loan_amount_num<0) || (yearly_interest_num <0) || (monthly_payment_num< 0)
    errordlg('You have entered invalid input. Please enter value again!
    ', 'Error');
    set(handles.loan_amount, 'string', '');
    set(handles.yearly_interest, 'string', '');
    set(handles.monthly_payment, 'string', '');
    breaker=1;

% if monthly interest greater than loan amount
elseif loan_amount_num < monthly_payment_num
    errordlg('You have entered invalid input. Please enter value again!
    ', 'Error');
    set(handles.loan_amount, 'string', '');
    set(handles.monthly_payment, 'string', '');
    breaker=1;

%loan cannot be lesser than 5000 and
%yearly interest cannot be lesser than 0.015
elseif (loan_amount_num < 5000) && (yearly_interest_num < 1.5)
    breaker=1;
    choice=questdlg({'Both loan amount and yearly interest are insufficient! Do
you wish to enter value again? '; ...
    'Choose "Yes" to continue '; ...
    'Choose "No" if you want to stop the program.'}, ...
    'Invalid input', 'Yes', 'No', 'Yes');
    switch choice
        case 'No'
            closereq;
        otherwise
            set(handles.loan_amount, 'string', '');
            set(handles.yearly_interest, 'string', '');
            set(handles.monthly_payment, 'string', '');
    end
elseif (loan_amount_num < 5000)
    breaker=1;
    choice=questdlg({'Loan amount too small! (Minimum: RM 5000)'; 'Do you wish to
enter value again? '; ...
    'Choose "Yes" to continue '; ...
    'Choose "No" if you want to stop the program.'}, ...
    'Invalid input', 'Yes', 'No', 'Yes');
    switch choice
        case 'No'
            closereq;
        otherwise
            set(handles.loan_amount, 'string', '');
    end
end

```

```

elseif (yearly_interest_num< 1.5)
    breaker=1;
    choice=questdlg({'Yearly interest too less! (Minimum: 1.5%)'; 'Do you wish
to enter value again? '; ...
    'Choose "Yes" to continue '; ...
    'Choose "No" if you want to stop the program.'}, ...
    'Invalid input','Yes','No','Yes');
    switch choice
        case 'No'
            closereq;
        otherwise
            set(handles.yearlly_interest,'string','');
    end

%re-enter if yearly interest is larger than 100
elseif yearly_interest_num >100
    breaker=1;
    choice=questdlg({'Yearly interest too high! (Maximum: 100%)'; 'Do you wish
to enter value again? '; ...
    'Choose "Yes" to continue '; ...
    'Choose "No" if you want to stop the program.'}, ...
    'Invalid input','Yes','No','Yes');
    switch choice
        case 'No'
            closereq;
        otherwise
            set(handles.yearlly_interest,'string','');
    end

%monthly payment cannot be smaller or equal to the value of interest.
%If not the person that pays the loan will not be able to finish paying
%his/her loan
elseif monthly_payment_num <= interest_num
    breaker=1;
    choice=questdlg({'Monthly Payment must be greater than interest! Do you wish
to enter value again? '; ...
    'Choose "Yes" to continue '; ...
    'Choose "No" if you want to stop the program.'}, ...
    'Invalid input','Yes','No','Yes');
    switch choice
        case 'No'
            closereq;
        otherwise
            set(handles.monthly_payment,'string','');
            set(handles.loan_repayment_num,'string','');
    end
end

if breaker ==1
    return
end

% initialize output header
loan_repayment=strcat({'|          Beginning Balance   ' ...
    ' |          Interest   ' |          Monthly Payment   ' ...
    ' |          Principal   ' |          Ending Balance   |'}], ...
    {'-----'
    '-----'});
set(handles.loan_repayment, 'string', loan_repayment);

```



```

%calculate input value till 0
while loan_amount_num > 0
    a = a + 1;
    %calculate
    interest_num=loan_amount_num*monthly_interest_num;
    principal_num=monthly_payment_num - interest_num;
    result_num=loan_amount_num-principal_num;

    new_line_string=strcat({' ', ...
    num2str(loan_amount_num, '%.2f'), ...
    {' ', ...
    num2str(interest_num, '%.2f'), ...
    {' ', ...
    num2str(monthly_payment_num, '%.2f'), ...
    {' ', ...
    num2str(principal_num, '%.2f'), ...
    {' ', ...
    num2str(result_num, '%.2f'), ...
    {' '});
    output_string = [output_string; new_line_string];
    loan_amount_num = result_num;

    %when loan amount is lesser than monthly payment, the ending balance
    %will be calculated to 0
    if loan_amount_num <= monthly_payment_num
        monthly_payment_num=loan_amount_num;
        interest_num=0;
        principal_num=monthly_payment_num;
        result_num=loan_amount_num-principal_num;
        last_line_string=strcat({' ', ...
        num2str(loan_amount_num, '%.2f'), ...
        {' ', ...
        num2str(interest_num, '%.2f'), ...
        {' ', ...
        num2str(monthly_payment_num, '%.2f'), ...
        {' ', ...
        num2str(principal_num, '%.2f'), ...
        {' ', ...
        num2str(result_num, '%.2f'), ...
        {' '});
        output_string = [output_string; new_line_string; last_line_string];
        break;
    end
    if a > 1000
        break;
    end
end

%output
set(handles.loan_repayment_num, 'string', output_string);

% -----
function file_Callback(hObject, eventdata, handles)
% hObject    handle to file (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% -----
function edit_Callback(hObject, eventdata, handles)
% hObject    handle to edit (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function help_Callback(hObject, eventdata, handles)
% hObject    handle to help (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
notes={'To produce an output, you should: ';
       'a) Enter Valid Input - NO NEGATIVE INPUT, '; ...
       'b) Yearly interest rate should be AT LEAST 1.5 %, '; ...
       'c) Loan amount should be AT LEAST 5000, '; ...
       'd) Monthly payment must be greater than interest.'};
helpdlg(notes, 'About');

% -----
function font_size_Callback(hObject, eventdata, handles)
% hObject    handle to font_size (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function font_color_Callback(hObject, eventdata, handles)
% hObject    handle to font_color (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function bkgrd_col_Callback(hObject, eventdata, handles)
% hObject    handle to bkgrd_col (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function open_Callback(hObject, eventdata, handles)
% hObject    handle to open (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
uigetfile

% -----
function save_Callback(hObject, eventdata, handles)
% hObject    handle to save (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
uiputfile

% -----
function print_Callback(hObject, eventdata, handles)
% hObject    handle to print (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles      structure with handles and user data (see GUIDATA)
printdlg

% -----
function close_prog_Callback(hObject, eventdata, handles)
% hObject      handle to close_prog (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
choice=questdlg('Do you wish to close this program? Current data will not be
saved.',...
                'Close program',...
                'Yes','No','No');
switch choice
    case 'No'
        return;
    otherwise
        close;
end

% -----
function restore_1_Callback(hObject, eventdata, handles)
% hObject      handle to restore_1 (see GCBO)
% eventdata    reserved - to be defined in a future version of MATLAB
% handles      structure with handles and user data (see GUIDATA)
choice=questdlg('Do you want to restore the program back to default?',...
                'Reset program',...
                'Yes','No','No');
switch choice
    case 'Yes'
        set(handles.loan_amount,'string','');
        set(handles.yearly_interest,'string','');
        set(handles.monthly_payment,'string','');
        set(gcf,'Color',[0.97 0.93 0.84]);
        try
            set(handles.loan_repayment_num,'String','');
        end
        set(handles.text2,'backgroundcolor',[0.97 0.93 0.84]);
        set(handles.text3,'backgroundcolor',[0.97 0.93 0.84]);
        set(handles.text4,'backgroundcolor',[0.97 0.93 0.84]);
        set(handles.text5,'backgroundcolor',[0.97 0.93 0.84]);
        set(handles.text2,'String','Loan Repayment Table:');
        set(handles.text2,'ForegroundColor',[0 0 0]);
        drawnow();
        set(handles.text3,'String','Loan Amount:');
        set(handles.text3,'ForegroundColor',[0 0 0]);
        drawnow();
        set(handles.text4,'String','Yearly Interest Rate:');
        set(handles.text4,'ForegroundColor',[0 0 0]);
        drawnow();
        set(handles.text5,'String','Monthly Payment:');
        set(handles.text5,'ForegroundColor',[0 0 0]);
        drawnow();
        set(handles.text2,'String','Loan Repayment Table:');
        set(handles.text2,'FontSize',25);
        drawnow();
        set(handles.text3,'String','Loan Amount:');
        set(handles.text3,'FontSize',15);
        drawnow();

```

```

        set(handles.text4,'String','Yearly Interest Rate:');
        set(handles.text4,'FontSize',15);
        drawnow();
        set(handles.text5,'String','Monthly Payment:');
        set(handles.text5,'FontSize',15);
        drawnow();
    otherwise
        return;
end

% -----
function grey_Callback(hObject, eventdata, handles)
% hObject    handle to grey (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(gcf,'Color',[0.941 0.941 0.941]);
set(handles.text2,'backgroundcolor',[0.941 0.941 0.941]);
set(handles.text3,'backgroundcolor',[0.941 0.941 0.941]);
set(handles.text4,'backgroundcolor',[0.941 0.941 0.941]);
set(handles.text5,'backgroundcolor',[0.941 0.941 0.941]);

% -----
function purple_Callback(hObject, eventdata, handles)
% hObject    handle to purple (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set ( gcf, 'Color', [0.4940, 0.1840, 0.5560] )
set(handles.text2,'backgroundcolor',[0.4940, 0.1840, 0.5560]);
set(handles.text3,'backgroundcolor',[0.4940, 0.1840, 0.5560]);
set(handles.text4,'backgroundcolor',[0.4940, 0.1840, 0.5560]);
set(handles.text5,'backgroundcolor',[0.4940, 0.1840, 0.5560]);
set(handles.text2,'String','Loan Repayment Table:');
    set(handles.text2,'ForegroundColor','w');
    drawnow();
    set(handles.text3,'String','Loan Amount:');
    set(handles.text3,'ForegroundColor','w');
    drawnow();
    set(handles.text4,'String','Yearly Interest Rate:');
    set(handles.text4,'ForegroundColor','w');
    drawnow();
    set(handles.text5,'String','Monthly Payment:');
    set(handles.text5,'ForegroundColor','w');
    drawnow();

% -----
function blue_Callback(hObject, eventdata, handles)
% hObject    handle to blue (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set ( gcf, 'Color', [0 0.4470 0.7410] )
set(handles.text2,'backgroundcolor',[0 0.4470 0.7410]);
set(handles.text3,'backgroundcolor',[0 0.4470 0.7410]);
set(handles.text4,'backgroundcolor',[0 0.4470 0.7410]);
set(handles.text5,'backgroundcolor',[0 0.4470 0.7410]);

```

```

% -----
function yellow_Callback(hObject, eventdata, handles)
% hObject    handle to yellow (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set ( gcf, 'Color', [0.9290, 0.6940, 0.1250] )
set(handles.text2,'backgroundcolor',[0.9290, 0.6940, 0.1250]);
set(handles.text3,'backgroundcolor',[0.9290, 0.6940, 0.1250]);
set(handles.text4,'backgroundcolor',[0.9290, 0.6940, 0.1250]);
set(handles.text5,'backgroundcolor',[0.9290, 0.6940, 0.1250]);

% -----
function green_Callback(hObject, eventdata, handles)
% hObject    handle to green (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set ( gcf, 'Color', [0.4660, 0.6740, 0.1880] )
set(handles.text2,'backgroundcolor',[0.4660, 0.6740, 0.1880]);
set(handles.text3,'backgroundcolor',[0.4660, 0.6740, 0.1880]);
set(handles.text4,'backgroundcolor',[0.4660, 0.6740, 0.1880]);
set(handles.text5,'backgroundcolor',[0.4660, 0.6740, 0.1880]);

% -----
function black_f_Callback(hObject, eventdata, handles)
% hObject    handle to black_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor',[0 0 0]);
drawnow();

% -----
function green_f_Callback(hObject, eventdata, handles)
% hObject    handle to green_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor',[0.4660, 0.6740, 0.1880]);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor',[0.4660, 0.6740, 0.1880]);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor',[0.4660, 0.6740, 0.1880]);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor',[0.4660, 0.6740, 0.1880]);
drawnow();

% -----

```

```

function red_f_Callback(hObject, eventdata, handles)
% hObject    handle to red_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor',[1 0 0]);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor',[1 0 0]);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor',[1 0 0]);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor',[1 0 0]);
drawnow();

% -----
function blue_f_Callback(hObject, eventdata, handles)
% hObject    handle to blue_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor',[0 0.4470 0.7410]);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor',[0 0.4470 0.7410]);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor',[0 0.4470 0.7410]);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor',[0 0.4470 0.7410]);
drawnow();

% -----
function white_f_Callback(hObject, eventdata, handles)
% hObject    handle to white_f (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor','w');
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor','w');
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor','w');
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor','w');
drawnow();

% -----
function style_1_Callback(hObject, eventdata, handles)
% hObject    handle to style_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');

```

```

set(handles.text2,'FontSize',20);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'FontSize',12);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'FontSize',12);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'FontSize',12);
drawnow();

% -----
function style_2_Callback(hObject, eventdata, handles)
% hObject    handle to style_2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'FontSize',22);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'FontSize',14);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'FontSize',14);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'FontSize',14);
drawnow();

% -----
function style_3_Callback(hObject, eventdata, handles)
% hObject    handle to style_3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'FontSize',25);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'FontSize',16);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'FontSize',16);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'FontSize',16);
drawnow();

% -----
function reset_Callback(hObject, eventdata, handles)
% hObject    handle to reset (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function font_Callback(hObject, eventdata, handles)

```

```

% hObject    handle to font (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function restore_Callback(hObject, eventdata, handles)
% hObject    handle to restore_1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
    set(handles.loan_amount,'string','');
    set(handles.yearly_interest,'string','');
    set(handles.monthly_payment,'string','');

    set(gcf,'Color',[0.97 0.93 0.84]);
try
    set(handles.loan_repayment_num,'string','');
end

    set(handles.text2,'backgroundcolor',[0.97 0.93 0.84]);
    set(handles.text3,'backgroundcolor',[0.97 0.93 0.84]);
    set(handles.text4,'backgroundcolor',[0.97 0.93 0.84]);
    set(handles.text5,'backgroundcolor',[0.97 0.93 0.84]);
    set(handles.text2,'String','Loan Repayment Table:');
    set(handles.text2,'ForegroundColor',[0 0 0]);
    drawnow();
    set(handles.text3,'String','Loan Amount:');
    set(handles.text3,'ForegroundColor',[0 0 0]);
    drawnow();
    set(handles.text4,'String','Yearly Interest Rate:');
    set(handles.text4,'ForegroundColor',[0 0 0]);
    drawnow();
    set(handles.text5,'String','Monthly Payment:');
    set(handles.text5,'ForegroundColor',[0 0 0]);
    drawnow();
    set(handles.text2,'String','Loan Repayment Table:');
    set(handles.text2,'FontSize',25);
    drawnow();
    set(handles.text3,'String','Loan Amount:');
    set(handles.text3,'FontSize',15);
    drawnow();
    set(handles.text4,'String','Yearly Interest Rate:');
    set(handles.text4,'FontSize',15);
    drawnow();
    set(handles.text5,'String','Monthly Payment:');
    set(handles.text5,'FontSize',15);
    drawnow();

% -----
function clear_Callback(hObject, eventdata, handles)
% hObject    handle to clear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

    set(gcf,'Color',[0.97 0.93 0.84]);

    set(handles.text2,'backgroundcolor',[0.97 0.93 0.84]);
    set(handles.text3,'backgroundcolor',[0.97 0.93 0.84]);

```



```

set(handles.text4,'backgroundcolor',[0.97 0.93 0.84]);
set(handles.text5,'backgroundcolor',[0.97 0.93 0.84]);
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'ForegroundColor',[0 0 0]);
drawnow();
set(handles.text2,'String','Loan Repayment Table:');
set(handles.text2,'FontSize',25);
drawnow();
set(handles.text3,'String','Loan Amount:');
set(handles.text3,'FontSize',15);
drawnow();
set(handles.text4,'String','Yearly Interest Rate:');
set(handles.text4,'FontSize',15);
drawnow();
set(handles.text5,'String','Monthly Payment:');
set(handles.text5,'FontSize',15);
drawnow();

% -----
function File_Callback(hObject, eventdata, handles)
% hObject    handle to File (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function clear_table_Callback(hObject, eventdata, handles)
% hObject    handle to clear_table (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
try
    set(handles.loan_repayment_num,'string','');
end

% -----
function Clear_Callback(hObject, eventdata, handles)
% hObject    handle to Clear (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% -----
function export_Callback(hObject, eventdata, handles)
% hObject    handle to export (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
[file path] = uiputfile('*.pdf');
print([path file], '-dpdf');

```

```

%-----
% --- Executes on selection change in loan_repayment_num.
function loan_repayment_num_Callback(hObject, eventdata, handles)
% hObject    handle to loan_repayment_num (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: contents = cellstr(get(hObject,'String')) returns loan_repayment_num
%         contents as cell array
%         contents{get(hObject,'Value')} returns selected item from
loan_repayment_num

% --- Executes during object creation, after setting all properties.
function loan_repayment_num_CreateFcn(hObject, eventdata, handles)
% hObject    handle to loan_repayment_num (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: listbox controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'),
get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% -----
function beige_Callback(hObject, eventdata, handles)
% hObject    handle to beige (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
set ( gcf, 'Color', [0.97 0.93 0.84] )
set(handles.text2,'backgroundcolor',[0.97 0.93 0.84]);
set(handles.text3,'backgroundcolor',[0.97 0.93 0.84]);
set(handles.text4,'backgroundcolor',[0.97 0.93 0.84]);
set(handles.text5,'backgroundcolor',[0.97 0.93 0.84]);

```

Explanation of features/ function

- **Menu and context features**

1. **Restore default**

This feature was used to restore the program that was already modified by the user according to their preferences based on the menu provided. The feature also reset the entire data that the user input. So, after clicking on 'restore' in the program, the whole program will return to its original settings. As an example, if the user changed the background colour to purple, changed the size of the font and the style of the font to their desired style and colour, upon clicking on restore, the program will immediately return to its original setting. In addition, the data in the table and the input data will be cleared. This feature can be found when the user clicks on either file or right click. The user should be able to see 'restore (default)' in the program.

2. **Clear settings**

This feature was used to clear the settings only. This means that only the background colour, font style and font colour will be affected. Anything that the user enters and the output from the data will not be affected. So, when the user wants to clear all the modification that they made without affecting the data that they input and the data that was shown in the table, the user can click on 'clear settings' from the menu. This feature can be found when the user right clicks at the background.

3. **Clear table**

Clear table was the feature that user could use to only clear the result. However, the data that was input in the edit box will not be affected since it was not included in the table. This feature can be found when the user right clicks within the table area and once the user clicks on the 'Clear table' option, it will clear the data in the table.

4. **File**

Under the file option from the menu, the user can choose to open their existing file, save their data, print out the file, export the result shown in the table and the value user had input as pdf, close the program or restore the program to its default setting.

5. Edit

Under the edit option, the users were given a choice to change the font size, font colour and background colour. For the font size, there will be 3 -ranging from the smallest font size to the largest font size- with style 1 being the minimum size option while style 3 being the maximum size option. For the font colour, there will be 5 colour choices available. The choices are black, green, red, blue and white. Lastly for the background colour, there will be 5 colour choices as well. The options are beige, purple, blue, yellow and green. Beige was set as the default background colour.

6. Help

As soon as the user click on the 'help' function, the users will be directed to the information of the program. From there, the users will be given a detailed explanation on criteria needed for the program to produce a valid output. As such, the user should enter valid input according to the criteria which are no negative input, the yearly interest rate should be at least 1.5%, the loan amount should be at least 5000 and the monthly payment must be greater than interest.

• **Additional features/function**

1. Question dialog

This feature will appear if user enter loan amount that is less than 5000 and at the same time entering yearly interest is less than 1.5%, a question dialog will appear to tell the users that the loan amount and yearly interest are insufficient, asking them to re-enter the values.

Another situation where the question dialogue appear is if the monthly payment entered by the users is less than the monthly interest after calculation. The question dialogue will pop up, telling the user that the monthly payment is too little. The question dialogue allows the users to choose either 'Yes' or 'No'.

The next situation is if the yearly interest entered by the user is less than 1.5% or more than 100%. The question dialogue will appear which will inform the users that the yearly interest is too little or too high, according to what the user entered. An option was also given by the dialogue either 'Yes' or 'No'.

The last situation is if the loan amount is less than 5000. The question dialogue will appear to inform the user that the entered value is lesser than the minimum criteria

which is 5000. An option was also given by the dialogue for the user to choose either 'Yes' or 'No'.

If the user clicks on 'Yes', the text area with the invalid input will be cleared and the user can re-enter another value. The question dialog will keep appearing until user enter the logic input that meets the criteria that has been set. If user click 'No', the program will automatically close.

2. Error dialog

This feature will appear when the user input the loan amount which is less than monthly payment. An error dialog will pop up to warn that the input loan amount is too small. Another case when the error dialog appear is when all the values that the user input are less than 0 which is a negative value. These pop-ups will trigger the GUI to clear the initial input that user have entered so that the user can easily identify the mistakes that they have made.

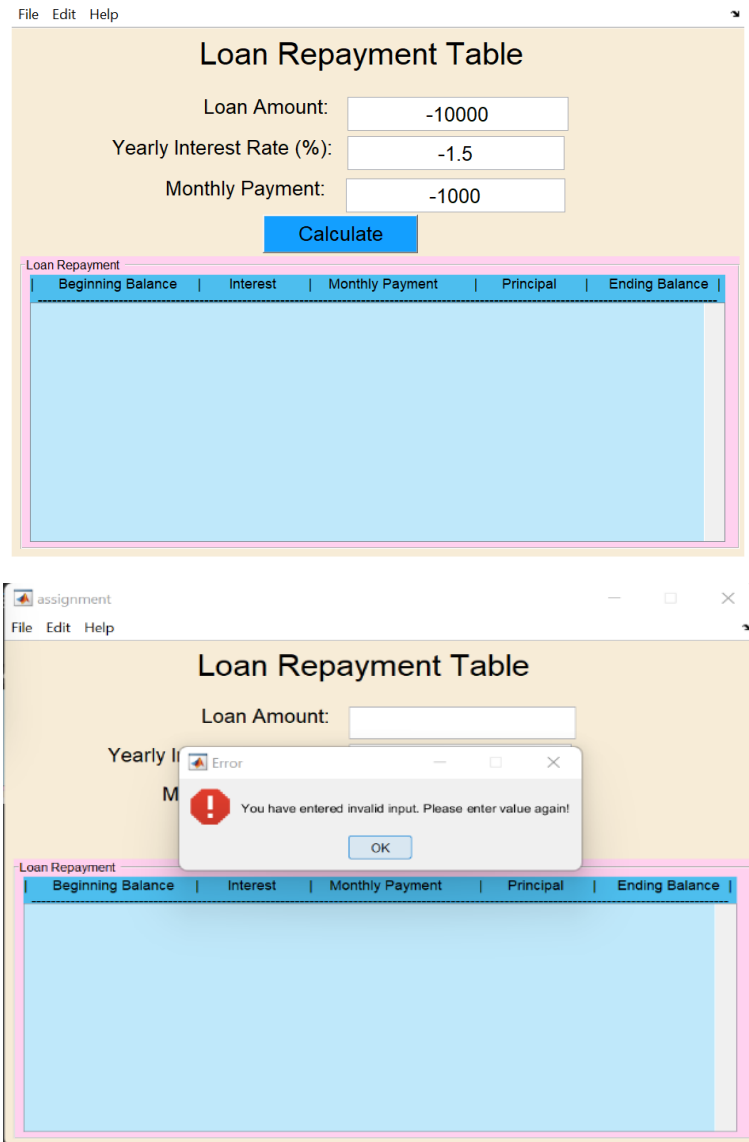
3. Loop

While loop function was included in the code. This loop was used in order to execute a statement multiple time. This makes sure the calculation will continue until the ending balance reached 0. As soon as it reached 0, the calculation will stop.

4. List box

The list box was used because it comes with the scroller. The scroller was needed as it allows the user to programmatically scroll to the bottom if the input needs to be calculated for many times until it reaches 0. This function is very useful when user input loan amount which is very big with low monthly payment. The static box has limitations since the users cannot view the complete data if it is too long whereas the list box does not due to the scrolling properties.

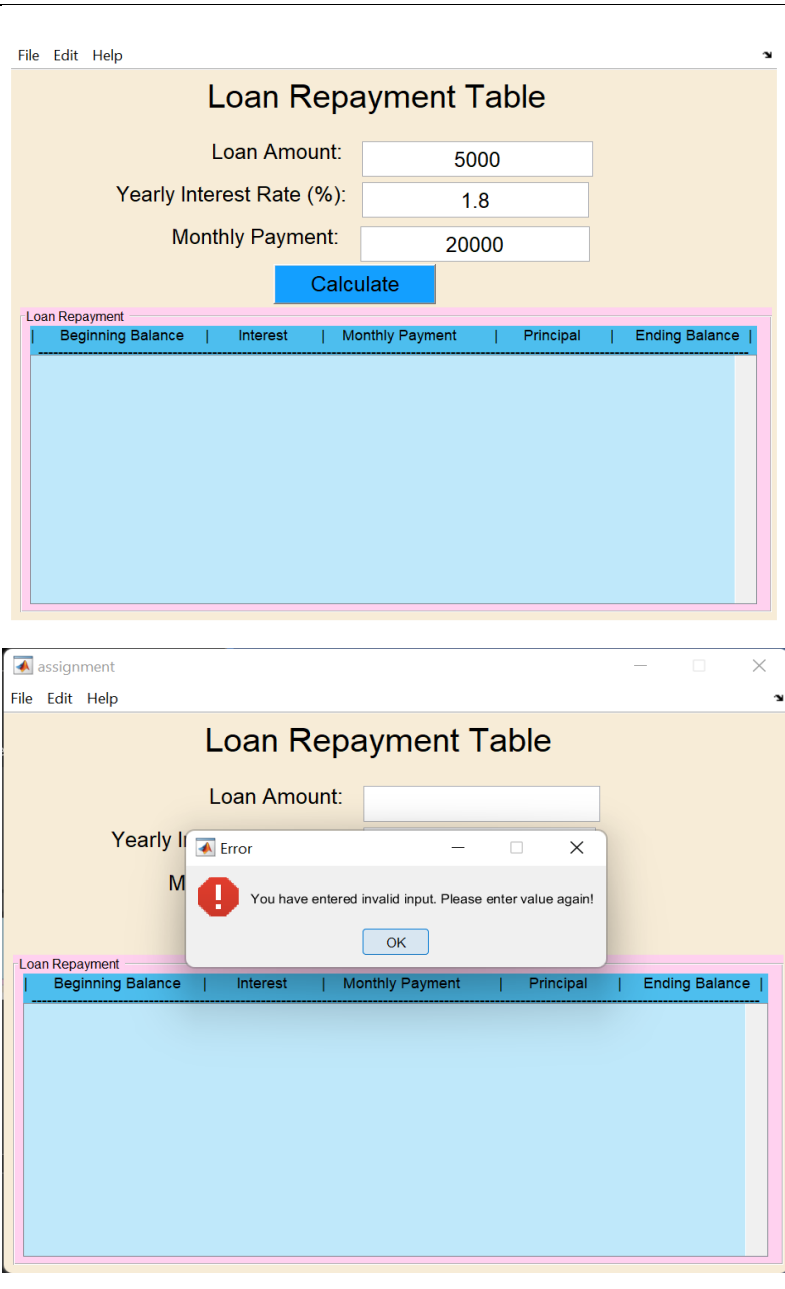
Testing

No	Test case	Output
1	<ul style="list-style-type: none"> Negative value <p>When the user enters negative value which is lower than 0, an error dialog will pop up to alert the user. Users can choose “OK” to re-enter a new value.</p>	 <p>The screenshot displays the 'Loan Repayment Table' application window. The input fields are populated with negative values: 'Loan Amount: -10000', 'Yearly Interest Rate (%): -1.5', and 'Monthly Payment: -1000'. A blue 'Calculate' button is visible below the inputs. Below the inputs is a table with the following headers: 'Beginning Balance', 'Interest', 'Monthly Payment', 'Principal', and 'Ending Balance'. The table body is currently empty. An 'Error' dialog box is overlaid on the application, displaying a red exclamation mark icon and the message: 'You have entered invalid input. Please enter value again!'. The dialog has an 'OK' button at the bottom.</p>

2

- Loan amount
lesser than
monthly
payment

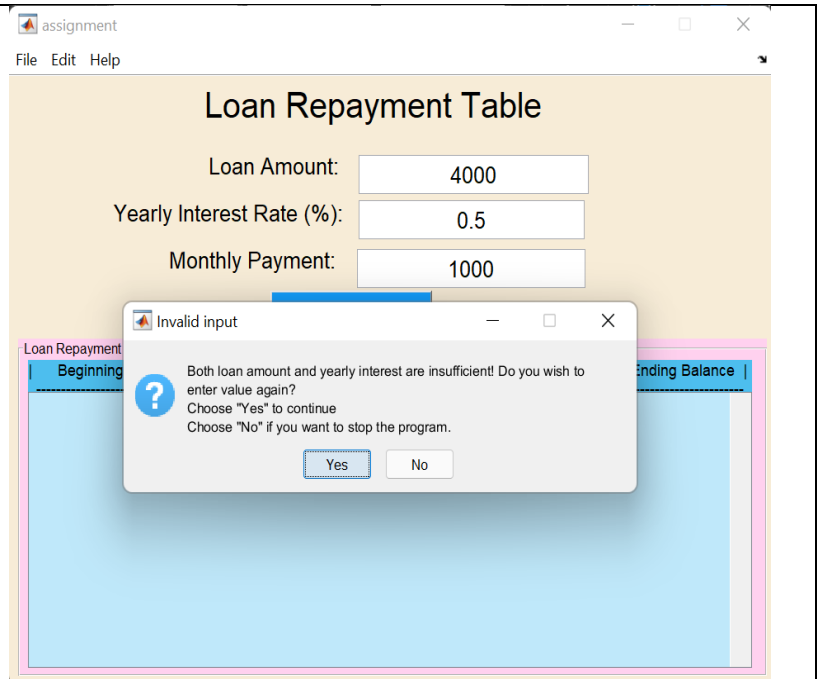
When the loan amount is lesser than monthly payment, an error dialog will pop up to alert the user. Users can choose “OK” to re-enter a new value.



3

- Loan amount less than 5000 and Yearly interest less than 1.5%

When the loan amount is less than 5000 and yearly interest is less than 1.5%, a question dialog about insufficient loan amounts and yearly interest will pop up to alert the user. Users can choose either yes which is to continue entering a new value or no which is to stop the program.



4

- Monthly payment less than monthly interest

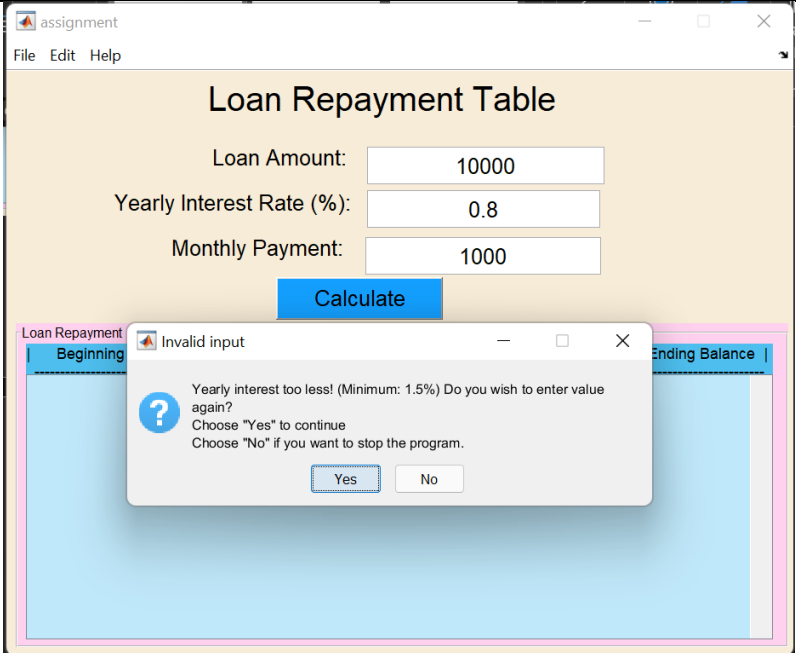
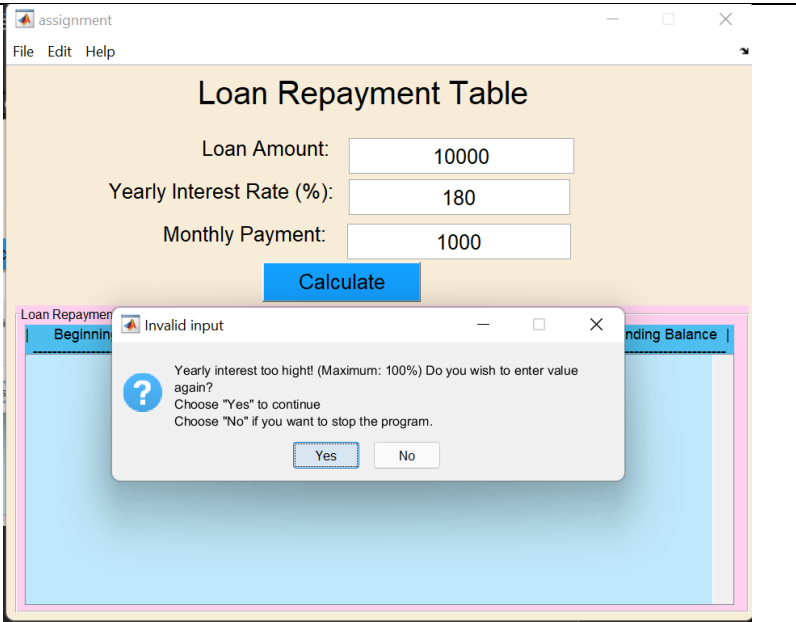
When the monthly payment is less than monthly interest, a question dialog about monthly payment must be greater than monthly interest will pop up to alert the user. User can choose either yes which is to continue entering a new value or no which is to stop the program.

The screenshot shows a window titled 'assignment' with a menu bar (File, Edit, Help). The main content area is titled 'Loan Repayment Table' and contains three input fields: 'Loan Amount:' with value '10000', 'Yearly Interest Rate (%):' with value '80', and 'Monthly Payment:' with value '500'. A blue 'Calculate' button is positioned below these fields. An 'Invalid input' dialog box is overlaid on the window, displaying a question mark icon and the text: 'Monthly Payment must be greater than interest! Do you wish to enter value again? Choose "Yes" to continue Choose "No" if you want to stop the program.' The dialog has 'Yes' and 'No' buttons.

Monthly payment < monthly interest

$$10000 * (80/100) / 12 = 666.67$$

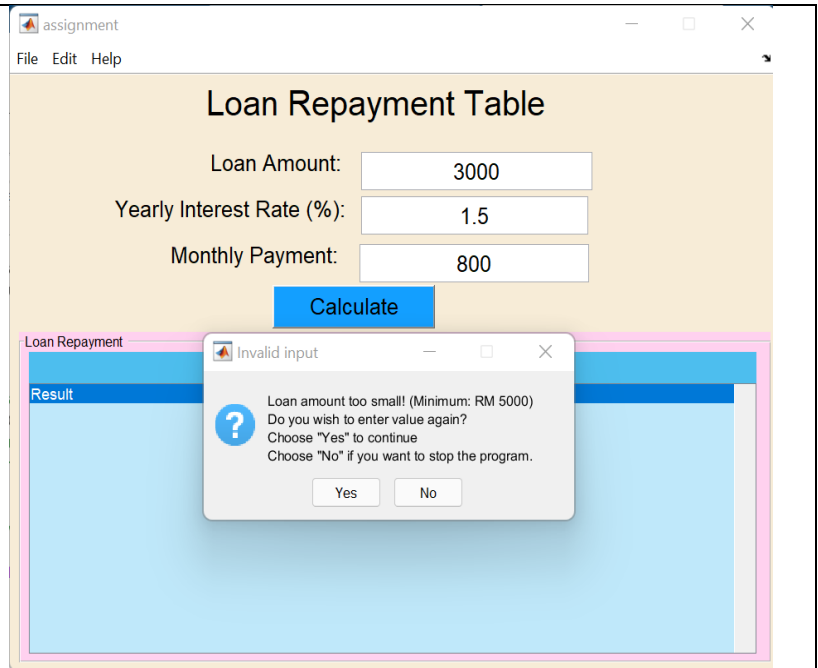
Monthly payment (500) < Monthly Interest (666.67)

<p>5</p>	<ul style="list-style-type: none"> Yearly Interest less than 1.5% <p>When the yearly interest is less than 1.5%, a question dialog about low yearly interest will pop up to alert the user. User can choose either yes which is to continue entering a new value or no which is to stop the program.</p>	
<p>6</p>	<ul style="list-style-type: none"> Yearly Interest more than 100% <p>When the yearly interest is more than 100%, a question dialog about high yearly interest will pop up to alert the user. User can choose either yes which is to continue entering a new value or no which is to stop the program.</p>	

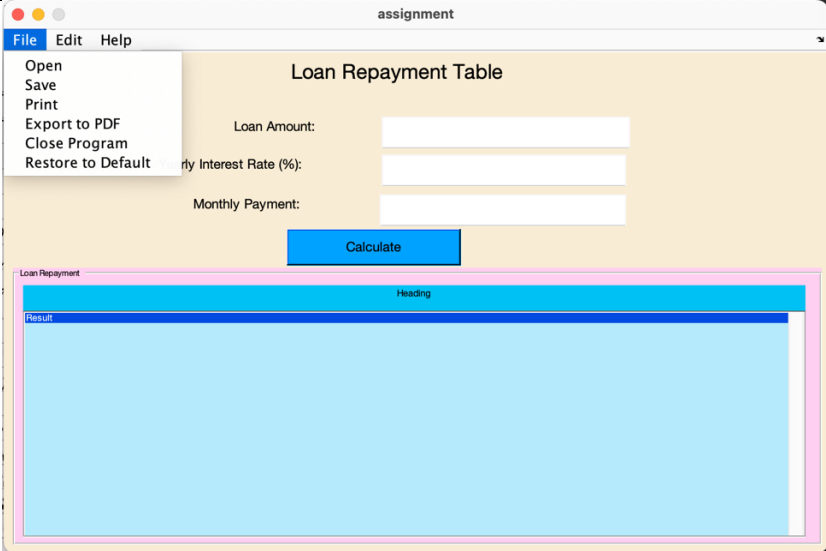
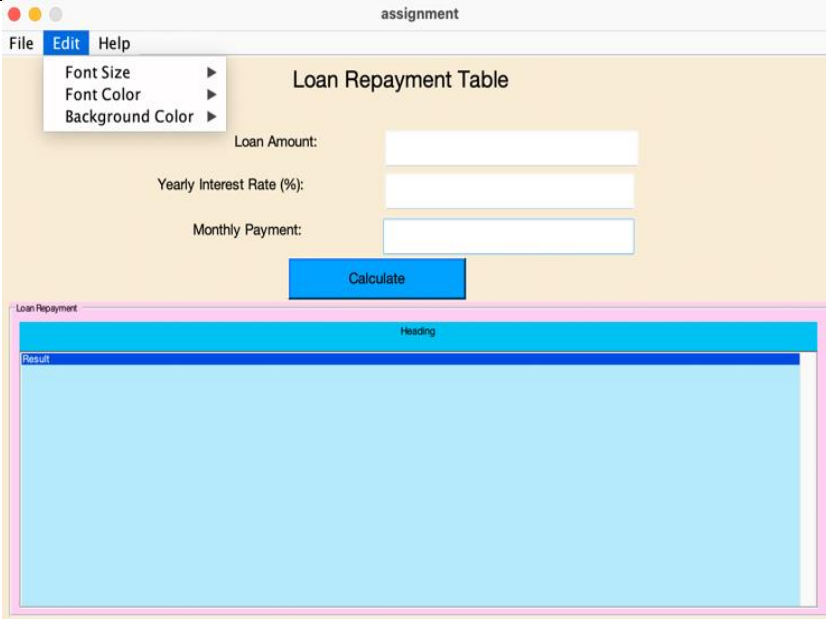
7

- Loan Amount less than 5000

When the loan amount is less than 5000, a question dialog about small loan amount will pop up to alert the user. Users can choose either yes which is to continue entering a new value or no which is to stop the program.



Menu and Context

No	Menu and Context	Example
1	<p>File</p> <ul style="list-style-type: none"> • Open • Save • Print • Export to PDF • Close Program • Restore to default <p>When users click on file which has several options for users such as open file, save file, print file, export file to pdf, close program and restore to default.</p>	
2	<p>Edit</p> <ul style="list-style-type: none"> • Font Size <ol style="list-style-type: none"> 1) Style 1 2) Style 2 3) Style 3 • Font Color <ol style="list-style-type: none"> 1) Black 2) Green 3) Red 4) Blue 5) White 	

- Background color

- 1) Beige
- 2) Grey
- 3) Purple
- 4) Blue
- 5) Yellow
- 6) Green

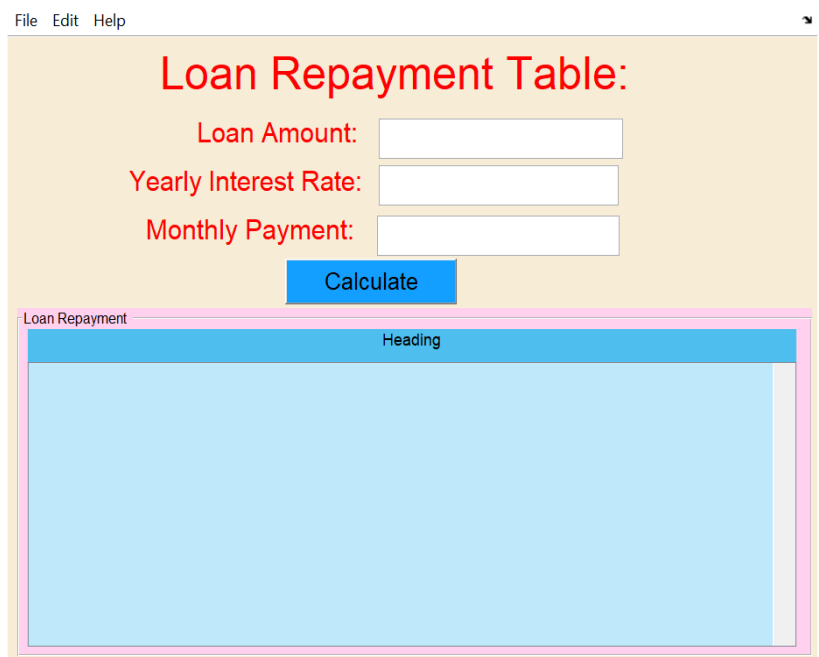
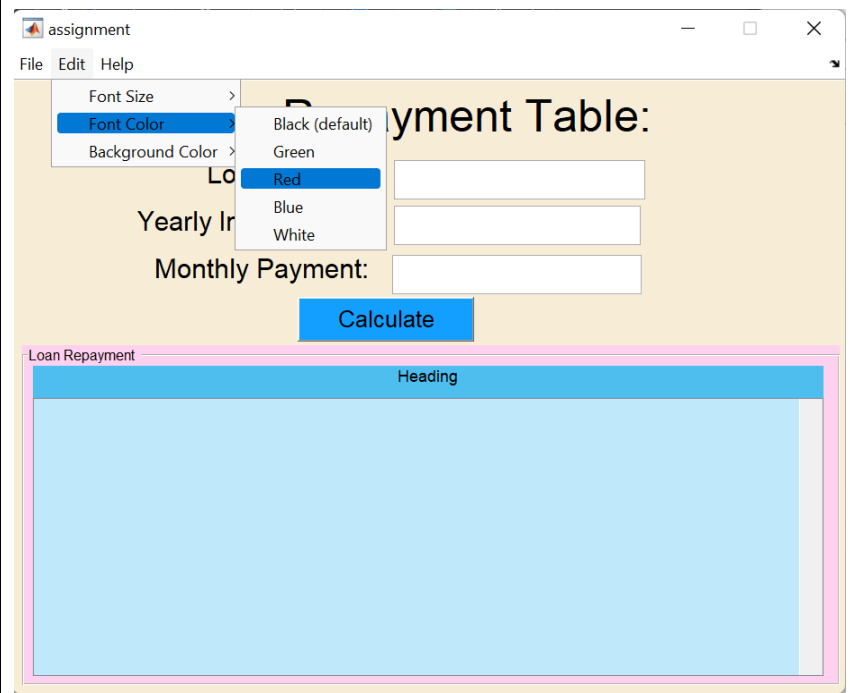
When the users click on edit, users can change the program options such as font size, font color and background color whichever they desire and comfortable with.

Font Size

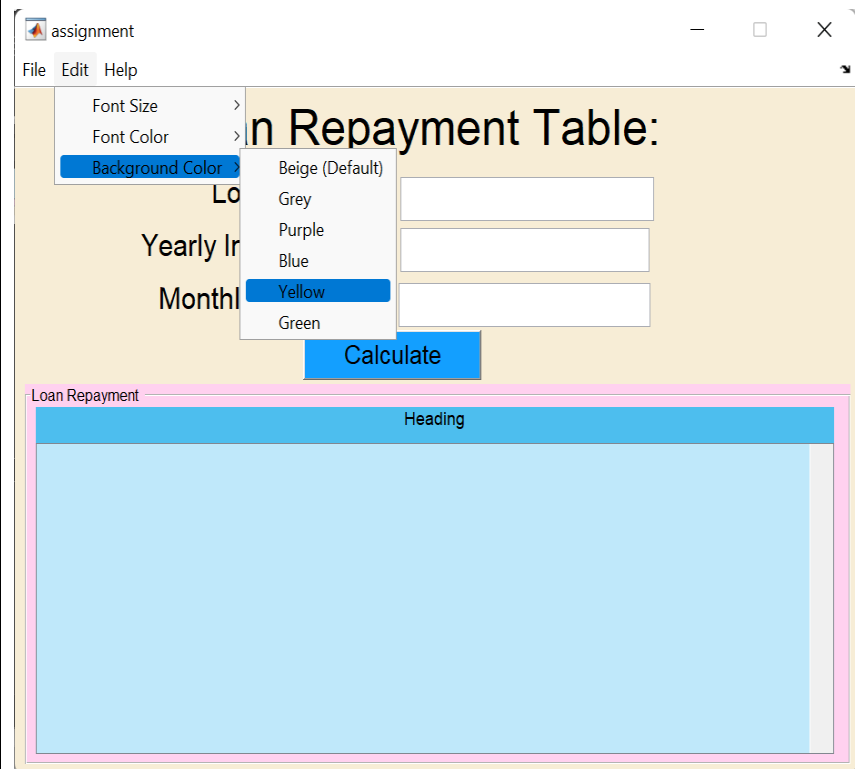
The screenshot shows a window titled 'assignment' with a menu bar containing 'File', 'Edit', and 'Help'. The 'Edit' menu is open, showing options: 'Font Size', 'Font Color', 'Background Color', 'Style 1', 'Style 2', and 'Style 3'. The 'Loan Repayment Table' form is visible in the background, featuring a beige background with the title 'Loan Repayment Table:'. It includes three input fields: 'Loan Amount:', 'Yearly Interest Rate:', and 'Monthly Payment:', each followed by a text box. Below these fields is a blue 'Calculate' button. At the bottom, there is a table with a pink border and a blue header row labeled 'Heading'.

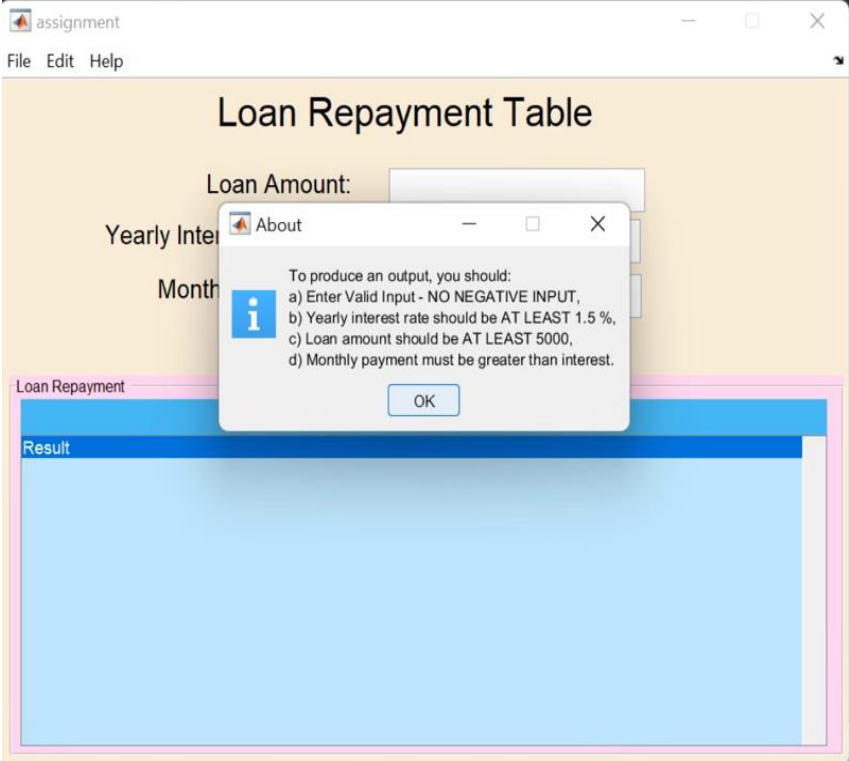
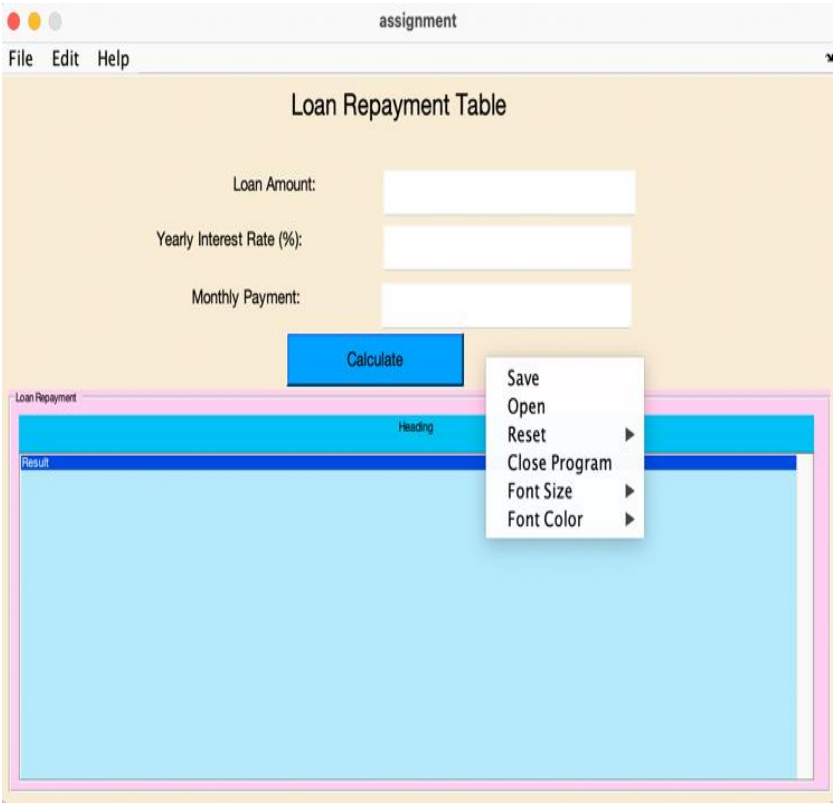
The screenshot shows the same 'assignment' window, but the 'Edit' menu is closed. The 'Loan Repayment Table' form is clearly visible with its beige background and title. It contains the same three input fields and the 'Calculate' button. Below the form, the table with the pink border and blue header row is also visible.

Font colour

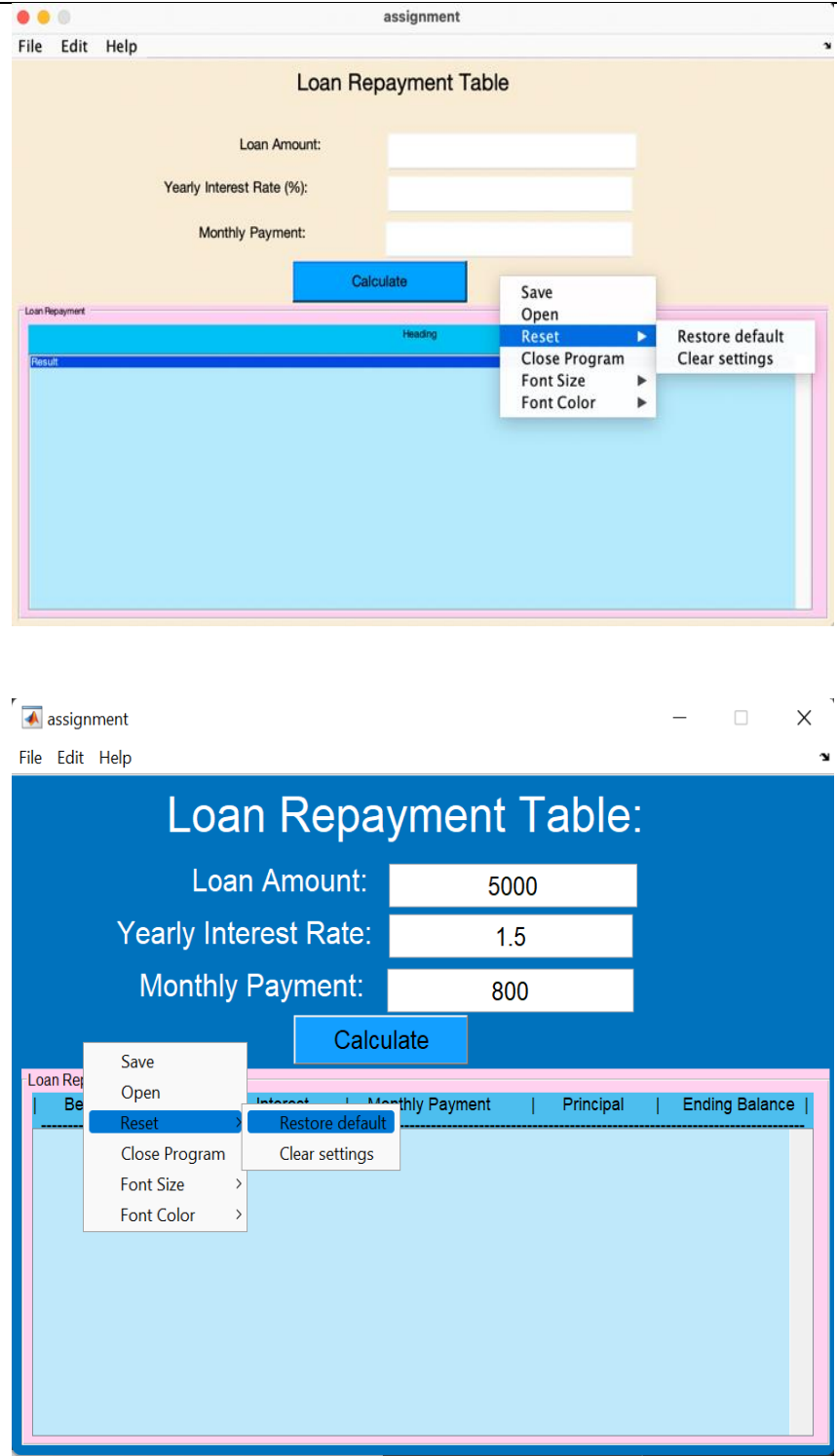


Background colour



<p>3</p>	<p>Help</p> <p>When users click on help, a pop up will appear with guidance for users who are confused.</p>	
<p>4</p>	<p>Right click</p> <ul style="list-style-type: none"> • Reset <ol style="list-style-type: none"> 1) Restore Default 2) Clear Settings <p>When the user right clicks on the program, there are options such as save, open, reset, close program, font size and font color. Users can change the font size and font color with right clicks as well as closing</p>	

the program or resetting the program which includes restore default or clear settings. For example, when user change their background color to blue, after they click on restore default, it will restore back to its default color.



When the user right clicks on the table, a clear table option will appear for users who wants to clear the data in the table.

Beginning Balance	Interest	Monthly Payment	Principal	Ending Balance
10000.00	1250.00	2000.00	750.00	9250.00
9250.00	1156.25	2000.00	843.75	8406.25
8406.25	1050.78	2000.00	949.22	7457.03
7457.03	932.13	2000.00	1067.87	6389.16
6389.16	798.65	2000.00	1201.35	5187.81
5187.81	648.48	2000.00	1351.52	3836.28
3836.28	479.54	2000.00	1520.46	1815.82
2315.82	289.48	2000.00	1710.52	605.29
2315.82	289.48	2000.00	1710.52	605.29
605.29	0.00	605.29	605.29	0.00

File Edit Help

Loan Repayment Table

Loan Amount:

10000

Yearly Interest Rate (%):

1.5

Monthly Payment:

2000

Calculate

Loan Repayment

Beginning Balance	Interest	Monthly Payment	Principal	Ending Balance
-------------------	----------	-----------------	-----------	----------------

Output

Valid Output

File Edit Help

Loan Repayment Table

Loan Amount:

Yearly Interest Rate (%):

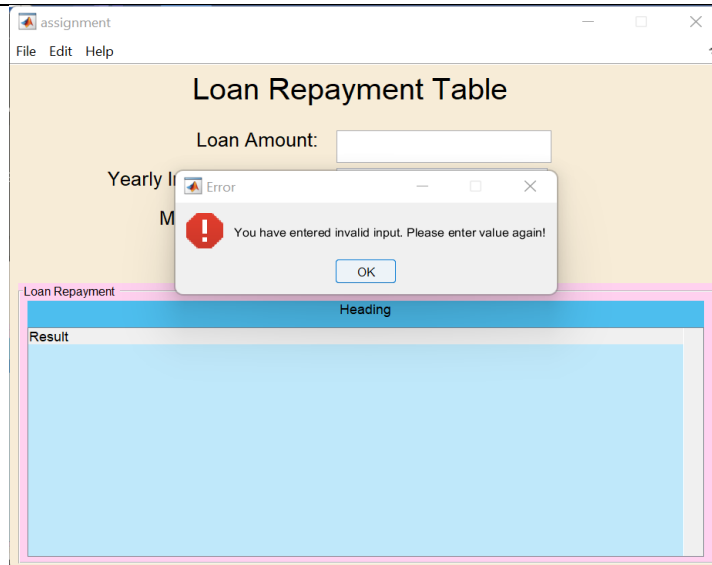
Monthly Payment:

Calculate

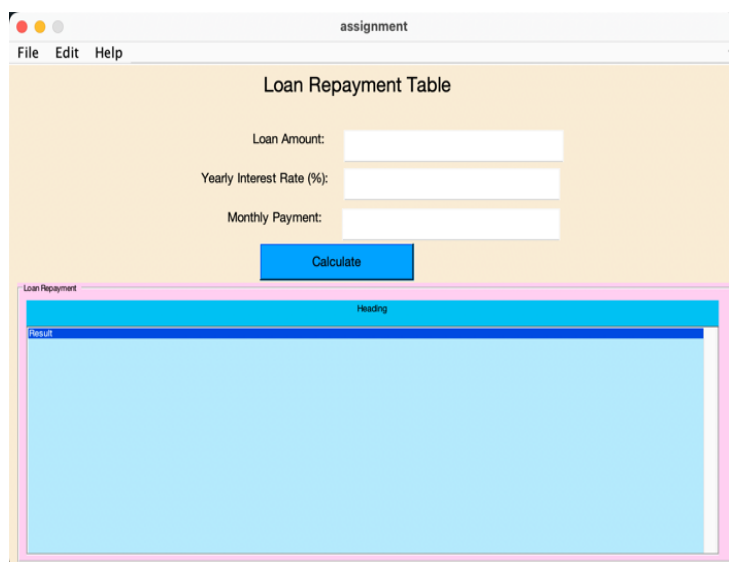
Beginning Balance	Interest	Monthly Payment	Principal	Ending Balance
10000.00	12.50	2000.00	1987.50	8012.50
8012.50	10.02	2000.00	1989.98	6022.52
6022.52	7.53	2000.00	1992.47	4030.04
4030.04	5.04	2000.00	1994.96	2035.08
2035.08	2.54	2000.00	1997.46	37.63
2035.08	2.54	2000.00	1997.46	37.63
37.63	0.00	37.63	37.63	0.00

Invalid Output

No	Example	Output			
1	When user enter a negative input for either one or all loan amount, yearly interest rate or monthly payment.	<div><div>File Edit Help</div><div><div>Loan Repayment Table</div><div><div>Loan Amount:</div><div>-10000</div></div><div><div>Yearly Interest Rate (%):</div><div>-1.5</div></div><div><div>Monthly Payment:</div><div>-2000</div></div><div>Calculate</div></div><div><div>Loan Repayment</div><table><tr><th>Heading</th></tr><tr><th>Result</th></tr><tr><td></td></tr></table></div></div>	Heading	Result	
Heading					
Result					



When user press 'OK', all the values that user have previously input will be cleared and user can re-enter the loan amount, yearly interest rate and monthly payment.



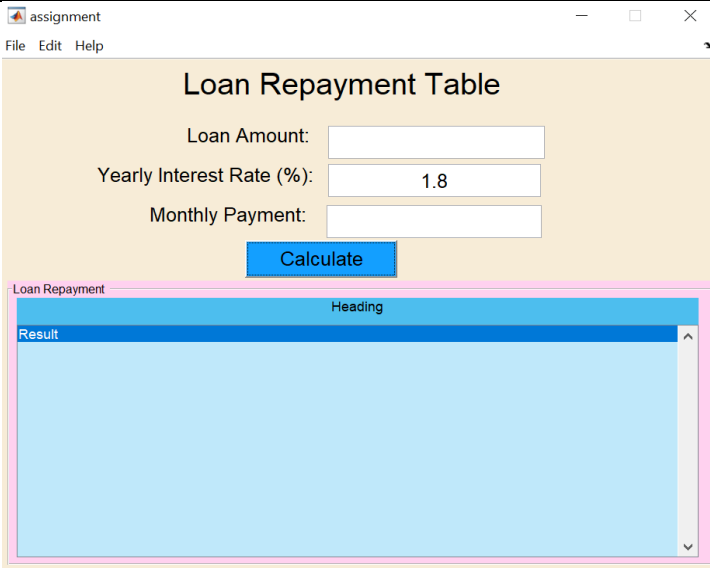
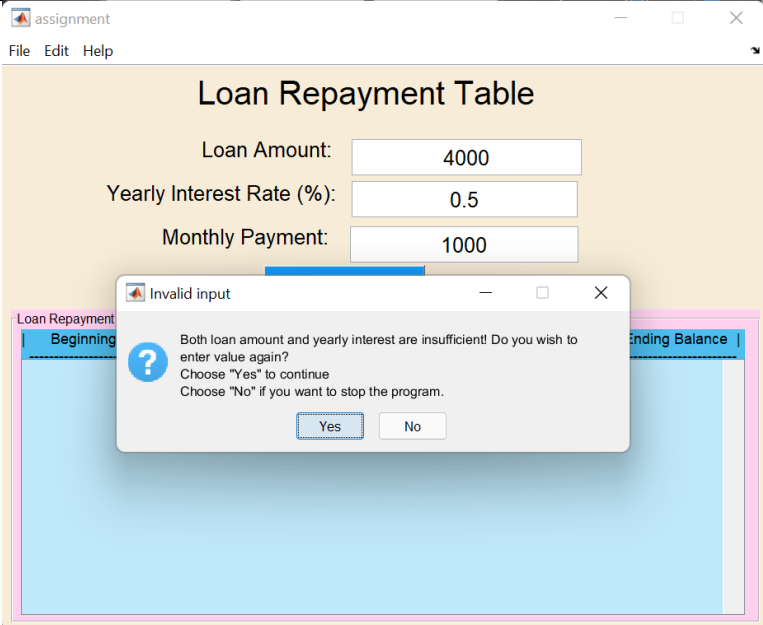
2

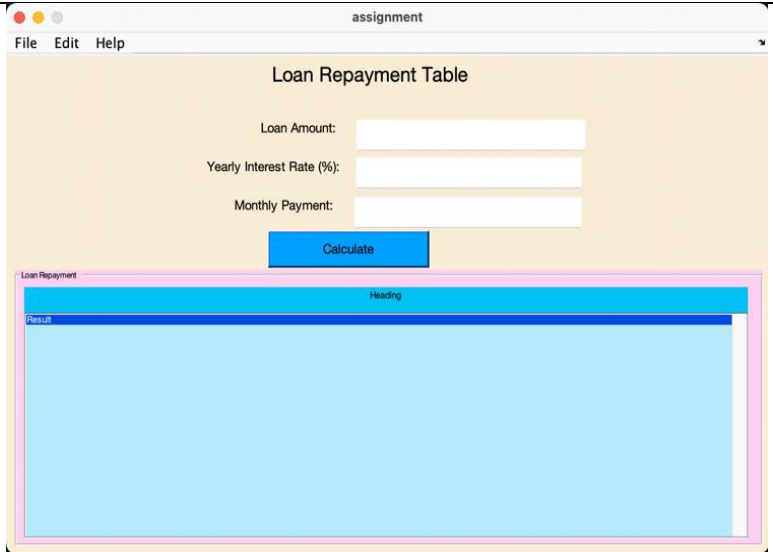
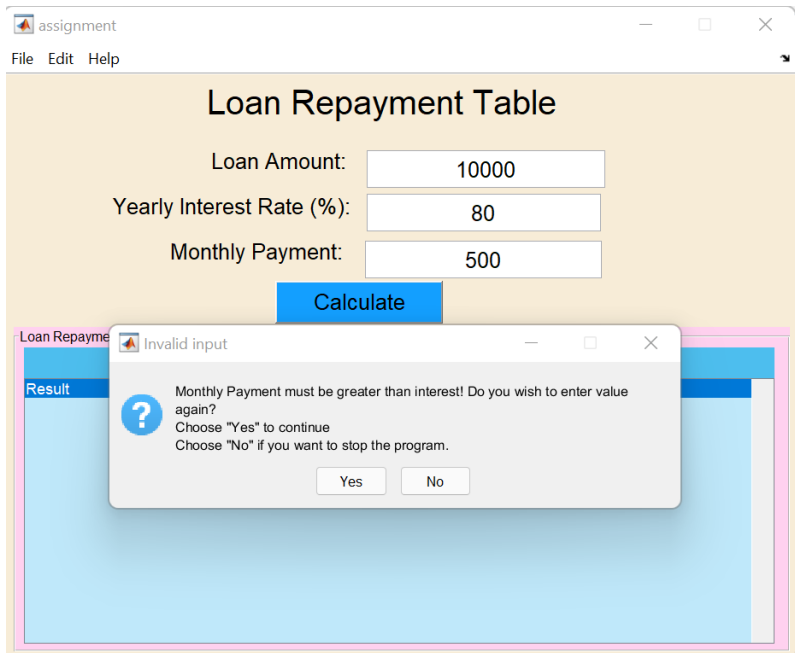
When the user enters
loan amount lesser
than monthly payment

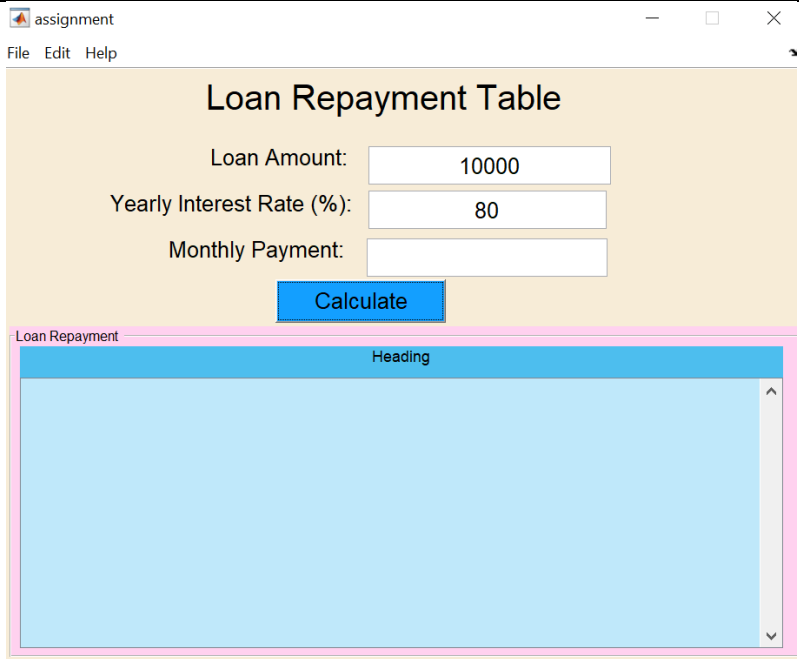
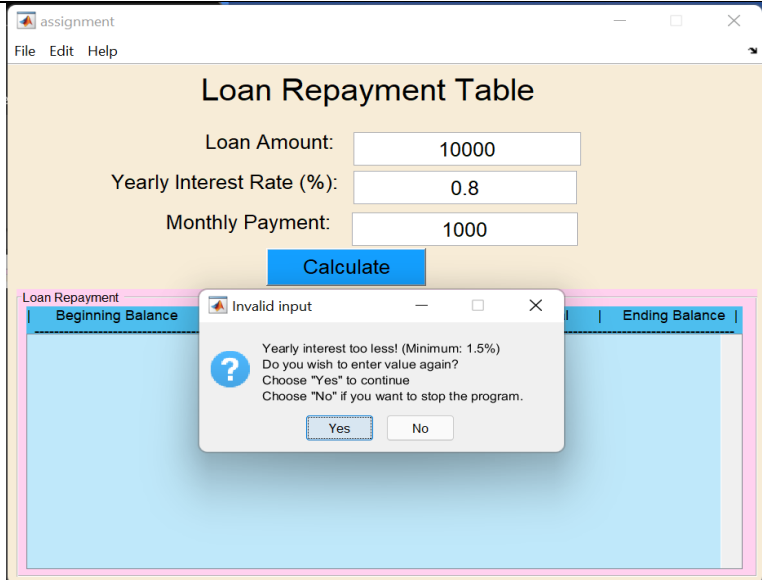
The screenshot shows a window titled "Loan Repayment Table" with a menu bar (File, Edit, Help). It contains three input fields: "Loan Amount:" with the value "5000", "Yearly Interest Rate (%):" with the value "1.8", and "Monthly Payment:" with the value "20000". A blue "Calculate" button is positioned below these fields. Below the button is a table titled "Loan Repayment" with a header row containing "Beginning Balance", "Interest", "Monthly Payment", "Principal", and "Ending Balance". The table body is currently empty and highlighted in light blue.

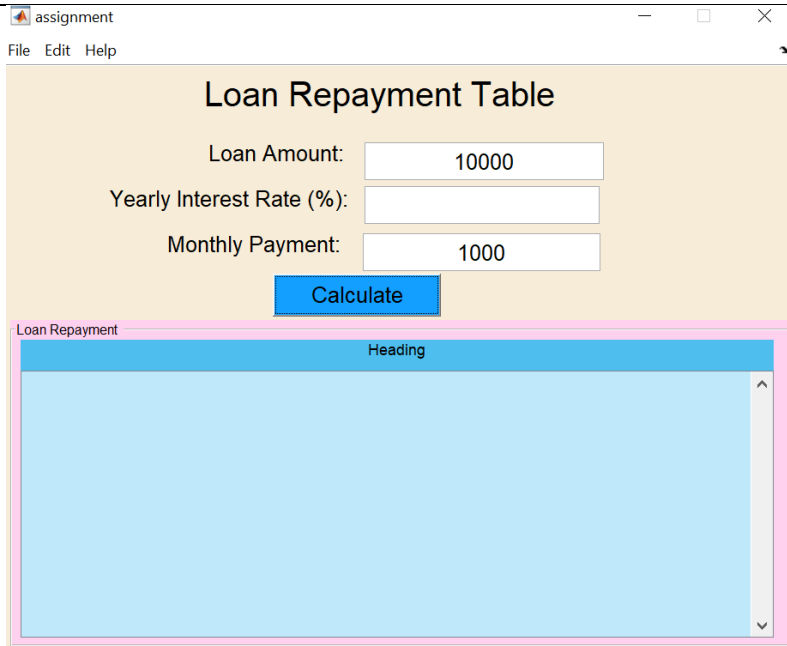
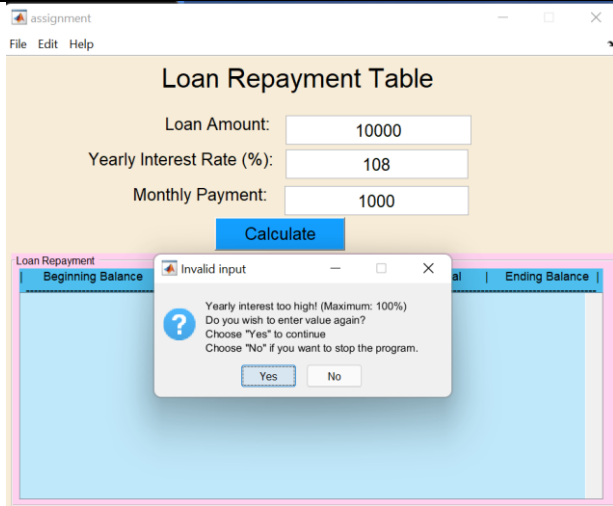
The screenshot shows the same "Loan Repayment Table" window, but with an error dialog box overlaid. The dialog box is titled "Error" and contains a red exclamation mark icon and the text "You have entered invalid input. Please enter value again!". An "OK" button is at the bottom of the dialog. In the background, the "Loan Amount:" field is empty, and the "Yearly Interest Rate (%):" field contains the value "1.8". The "Monthly Payment:" field is also empty. The "Loan Repayment" table is visible below the dialog box.

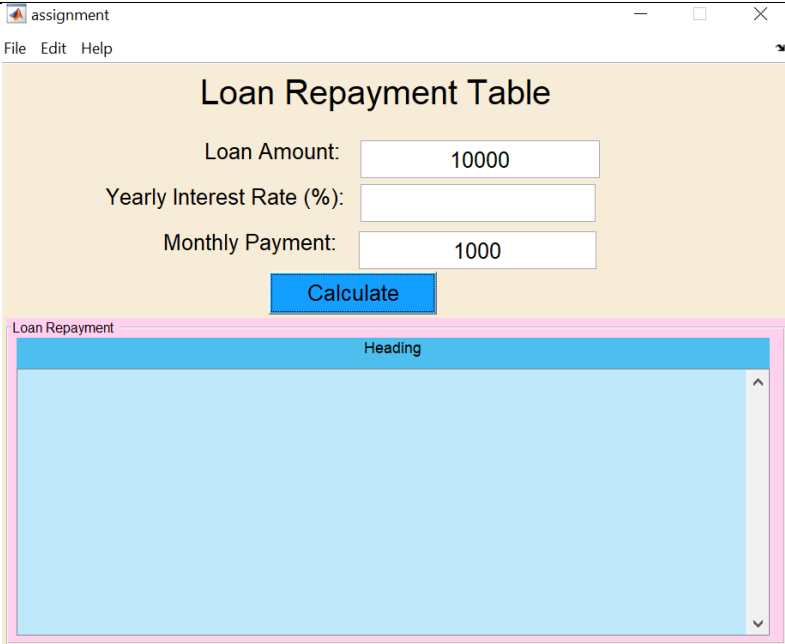
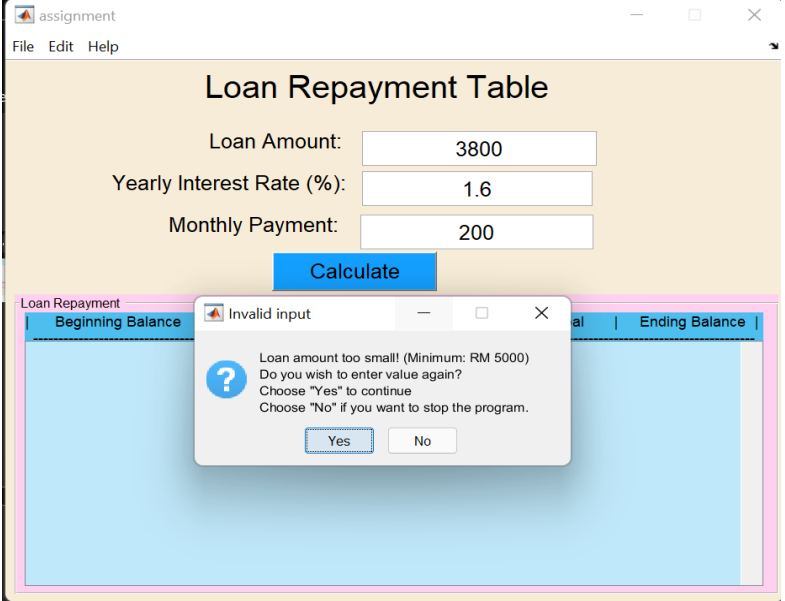
When user press 'OK', the value that user has previously entered for loan amount and monthly interest will be cleared and user can re-enter the loan amount and monthly payment.

		
3	<p>When the user enters loan amount lesser than 5000 and yearly interest lesser than 1.5%</p>	 <p>If the user clicks on 'Yes', the input will be cleared and user must enter valid values such as sufficient loan amount and yearly interest, thus it will calculate. If the user chooses 'No', the program will stop.</p>

		
4	<p>When user enter monthly payment which is then calculated to be lesser than monthly interest</p>	<p>Monthly payment < monthly interest</p> $10000 * (80/100) / 12 = 666.67$ <p>Monthly payment (500) < Monthly Interest (666.67)</p>  <p>If the user clicks on 'Yes', the monthly payment input will be cleared and user must enter valid input such as monthly payment more than monthly interest, thus it will calculate. If the user chooses 'No', the program will stop.</p>

		
5	When user enter yearly interest lesser than 1.5%	 <p>If the user clicks on 'Yes', the yearly interest input will be cleared and user must enter valid values such as yearly interest more than 1.5%, thus it will calculate. If the user chooses 'No', the program will stop.</p>

		
6	When the user enters yearly interest more than 100%	 <p>If the user clicks on 'Yes', the input will be cleared and user must enter valid values such as yearly interest not more than 100%, thus it will calculate. If the user chooses 'No', the program will stop.</p>

		
7	When the user enters loan amount lesser than 5000.	 <p>If the user clicks on 'Yes', the loan amount input will be cleared and user must enter valid values such as loan amount more than 5000, thus it will calculate. If the user chooses 'No', the program will stop.</p>

assignment

File Edit Help

Loan Repayment Table

Loan Amount:

Yearly Interest Rate (%):

Monthly Payment:

Calculate

Loan Repayment

Heading

Marking Rubric

Group	CRITERIA	EXCELLENT	GOOD	SATISFACTORY	UNSATISFACTORY	TOTAL
		10	8-9	6-7	5	
GUI	Interactive/ appropriate	Additional features/ creative/ menu interface for the task	Appropriate interface for the task with menu only	Appropriate interface for the task.	Not appropriate interface for the task.	
Output	Accuracy of output based on range of test data	Accurate and covering all the criteria in question	Accurate and covering majority of criteria in question	Accurate and covering average number of criteria in question	Acceptable output and covering any random criteria in question	
M- Scripts	Scripts and creativity	Able to execute without any error and additional creativity such as use of customize function /control structure wherever it's applicable in meeting basic requirements of the program.	Able to execute without any error and additional creativity such as use of built in function/control structure in meeting basic requirements of the program.	Able to execute without any error and meet basic requirements of the program	Able to execute with inaccurate output and missing of basic requirements of the program.	
Features	Features/ explanation	Detailed explanation on the GUI features	Appropriate and acceptable explanation on the GUI features	Very minimal explanation on the GUI features	Not explaining at all	
Testing	Validation	Include warning messages/ error messages more than twice and providing option to the user to rectify the error	Include warning messages/ error messages minimum twice	Include warning messages/ error messages minimum once	Not including warning messages/ error messages	
Feedback						TOTAL
						/50