PRACTICAL ASSIGNMENT 9

By end of this lab

- Student will perform program integration using Visual Studio.
  - *Game integration and testing*
    1. Create currency convertor, C# program.
    2. Setup GIT Repository.
    3. Add collaborator.
    4. Merge the branches by the owner.
    5. Test the project and record test results.

To complete this lab

- PC
- Visual Studio
- GitHub Desktop Client
- GIT Cloud account

Given Files :

This PPT slides

Test Report

Step 1 : Prerequisites

- Create a new folder and rename it as IndexNumber_PA9_YourName
- Save the C# project inside the folder created

Step 2 : Create currency convertor, C# program. (Student A and Student B).

1. Create a new **Windows Form Application** project and set the location to Desktop, set the name of the project to your Index Number, name and practical number.

Step 2 : Create currency convertor, C# program. (Student A and Student B).

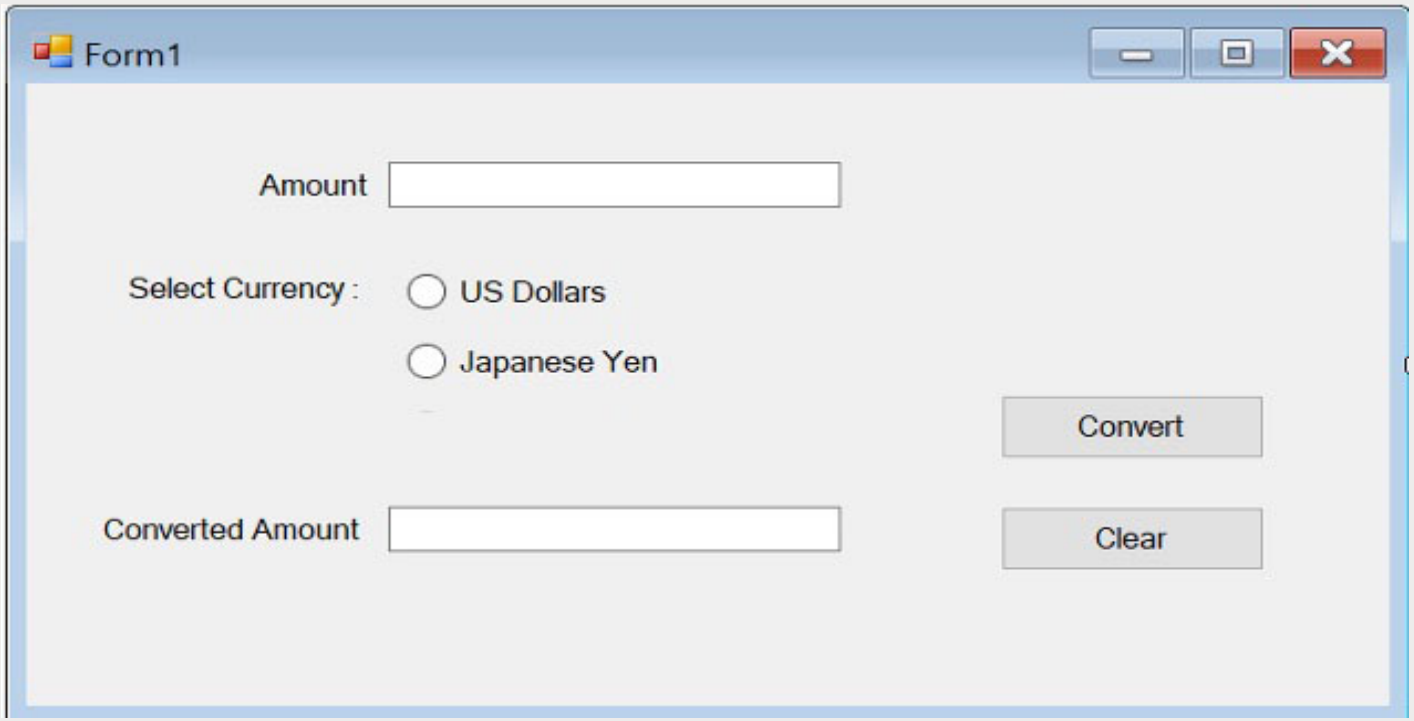
1. Add **3 labels** to your form, **change the text** for the 3 labels to **Amount, Select Currency and Converted Amount.** We will use the labels to provide information for the users.
2. Add **2 textboxes** to your form, **change the name** to **txt_Amount** and **txt_Converted.** The first textbox will be for the user to enter the amount and the second textbox will display the converted amount.
3. Add **2 buttons** to your form, **change the name** of the first button to **btn_Convert** and **change the text to Convert. Change the name** of the second button to **btn_Clear** and **change the text** to **Clear.**
4. Add **2 radio buttons** to your form. **Change the name** of the radio buttons to **rb_US** and **rb_Yen. Change the text** of the radio buttons to **US Dollars** and **Japanese Yen** respectively.

Step 2 : Create currency convertor, C# program. (Student A and Student B).

1. Add **3 labels** to your form, **change the text** for the 3 labels to **Amount, Select Currency and Converted Amount.** We will use the labels to provide information for the users.
2. Add **2 textboxes** to your form, **change the name** to **txt_Amount** and **txt_Converted.** The first textbox will be for the user to enter the amount and the second textbox will display the converted amount.
3. Add **2 buttons** to your form, **change the name** of the first button to **btn_Convert** and **change the text to Convert. Change the name** of the second button to **btn_Clear** and **change the text** to **Clear.**
4. Add **2 radio buttons** to your form. **Change the name** of the radio buttons to **rb_US** and **rb_Yen. Change the text** of the radio buttons to **US Dollars** and **Japanese Yen** respectively.

Step 2 : Create currency convertor, C# program. (Student A and Student B).

- Arrange the components so that your form has a similar layout as below.



Step 3 :
Create Event

- Create the event for the Convert button and add the necessary codes so that after the user enters an amount in the first textbox and clicks on the Convert, the amount will be converted to the selected currency base on the selection of the radio buttons

Step 3 :
Create Event

- Use the following conversion rate for the different currencies:
  - 1 SGD converts to 0.74 US Dollar
  - 1 SGD converts to 81.97 Japanese Yen
- Note: The radio buttons allow the user to select a single option from a group of choices. Each radio button has a property Checked (under Appearance in the Properties window).

Step 3 : Create Event

- We can check if a radioButton is selected in our code by accessing the Checked property of a radio butto; **radioButton.Checked**. If a particular radio button is selected, the
- Checked parameter will return true otherwise it will return false.
- We can use if – else statements to determine which radio button is selected (**.Checked**) to determine the which conversion rate to use.

```
if(radioButton_A.Checked)


//Do something


}
```

Step 4 : Debug and Save Project

- Debug your application, ensure that the conversion returns the correct value.
- Implement the functionality for the Clear button. When the user clicks on the Clear button, the amount and the converted amount will be cleared.
- Implement the functionality to handle exception when the user attempts to convert an invalid number for the amount. The application will display “Please enter a valid amount” after clicking on Convert.
- Debug your application ensure that the functionalities of the application works as intended.
- Save and submit your project.

Part B – Set up GIT repository (Student A and Student B).

- 1. Login into GitHub client
- 2. Create a repository and name it as **IndexNumber_PA#_YourName**
- 3. Click show in explorer
- 4. Copy the Currency convertor project to the **IndexNumber_PA#_YourName** folder
- 5. Commit the changes to Master

Part B – Set up GIT repository (Student A and Student B).

- 6. Eg: Message : Currency convertor project created by YourName
- 7. Publish the repository
- 8. Access to GitHub Cloud and verify the contents for master branch
- 9. Invite a collaborator to your repository
- 10. Before performing this step, get the email address of your collaborator
  - • *(Eg: Student A need to add in Student B, Student B need to add in Student A)*
  - • *Access to GitHub Cloud*
  - • *Click Settings > Manage Access > Invite a Collaborator*

Part C – Add a collaborator (Student A & Student B) & Update the existing project.

- 1. Invited collaborator will receive an email
- 2. Click the link provided in your email to accept the invitation
- 3. Clone the repository to your local GitHub client
- 4. Student A clones the repository created by Student B
- 5. Student B clones the repository created by Student A
- 6. File > Clone Repository > URL
- 7. Create a new branch > branch1

Part C – Add a collaborator (Student A & Student B) & Update the existing project.

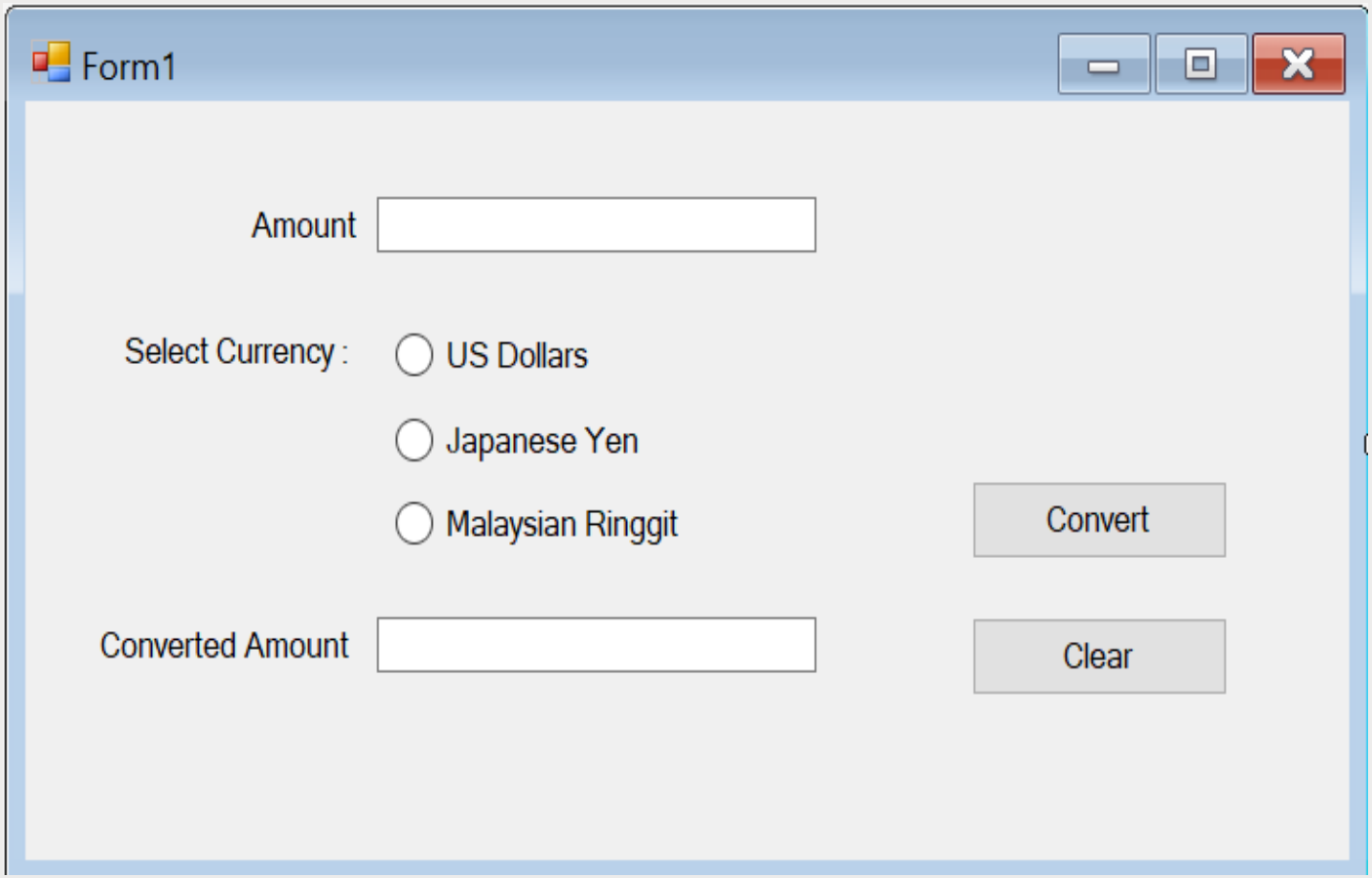
- 8. Click Show in Explorer in GitHub Client
- 9. Copy the currency convertor project to your desktop
- 10. Open the project from the folder
- 11. Add the necessary codes to do conversion for Malaysian Ringgit.
  - *1 SGD converts to 3.01 Malaysian Ringgit*

Add a **radio button** to your form. **Change the name** of the radio button to **rb_Ringgit. Change the text** of the radio button to **Malaysian Ringgit** respectively.

Part C – Add a collaborator (Student A & Student B) & Update the existing project.

- 12. Perform testing.
- 13. Save and close the project as **IndexNumber_PA#_YourName_CCProject**
- 14. Access to GitHub Desktop Client
- 15. Click > Show in Explorer
- 16. Copy the updated project and paste/replace it in the repository
- 17.Add in a commit message “Updated by YourName”
- 18. Commit the file to **Branch1.**
- 19. Access to Github cloud
- 20. Verify the new branch creation and content
- 21. Update your other team member about the changes in the repository

Part C – Add a collaborator (Student A & Student B) & Update the existing project.



Part D – Merge the branches by the owner.

- 1. Access to GitHub Cloud
- 2. Click Pull Request
- 3. Type in a message (eg: Merging Version 2 to Master) – eg: Updated by YourName
- 4. Merge the contents from branch1 to Master branch
- 5. Access to the repository and download the currency convertor project to your desktop
- 6. Unzip the file and access to the folder contents
- 7. Test the project for all currency conversions.

Part E – Test and record test results

Test Case ID	Description	Input Data	Expected Result	Actual Result	Pass/Fail	Remarks
01	Test US Dollars conversion.	100	74	74	Pass	
02	Test Japanese Yen conversion.	1	81.97	81.97	Pass	
03	Test Malaysian Ringgit.	1	3.01	3.01	Pass	
04	Test Alphabet.	aBd	Please enter a valid amount	Please enter a valid amount	Pass	
05	Test Symbol.	*@+-/=	Please enter a valid amount	Please enter a valid amount	Pass	
06	Test Alphabet with number.	A0n	Please enter a valid amount	Please enter a valid amount	Pass	
07	Test Symbol with number.	-9	Please enter a valid amount	Please enter a valid amount	Pass	Still gave converted amount for negative value
08	Test clear button	Click on clear button	Both text field cleared	Both text field cleared	Pass	

Submission

- C# Project – IndexNumber_PA9_YourName
- Test Report
- Upload to MsTeams