|  |  |
| --- | --- |
| **Laboratory Activity No. 9** | |
| **Introduction to GUI Development using Pycharm** | |
| **Course Code:** CPE103 | **Program:** BSCPE |
| **Course Title:** Object-Oriented Programming | **Date Performed:** March 22, 2025 |
| **Section:** 1 – A | **Date Submitted:** March 22, 2025 |
| **Name:** Hermosura, Leigh B. | **Instructor:** Maria Rizette M. Sayo |
| **1. Objective(s):** | |
| This activity aims to familiarize students with the Pycharm framework for GUI Development | |
| **2. Intended Learning Outcomes (ILOs):** | |
| The students should be able to:   * 1. Identify the main components in a GUI Application   2. Create a simple GUI Application using Pycharm Widgets | |
| **3. Discussion:** | |
| A Graphical User Interface (GUI) application is a program that the user can interact with through graphics (windows, buttons, text fields, checkboxes, images, icons, etc..) such as the Desktop GUI of Windows OS by using a mouse and keyboard unlike with a Command-line program or Terminal program that support keyboard inputs only.  Pycharm is an integrated development environment used for programming in Python. It provides code analysis, a graphical debugger, an integrated unit tester, integration with version control systems, and supports web development with Django. | |
| **4. Materials and Equipment:** | |
| Desktop Computer with Anaconda Python or Pycharm Windows Operating System | |
| **5. Procedure:** | |

|  |
| --- |
| 1. Run the program and observe the output.   **Adding an icon**   1. Download any .ico picture from <https://icon-icons.com/> or any similar sites. 2. Place the icon in your folder (ex. Oopfa1<lastname>\_lab8) 3. Run the program again, the program should now have an icon similar to the program below.     **Creating Buttons**   1. Create a new .py file named **gui\_buttons.py** then copy the program as shown below: |

|  |
| --- |
| 1. Run the program and observe the output. 2. Add a new button named button2 named Register to the GUI that will display “this button does nothing.. yet..” when it is hovered.   **Creating Text Fields**   1. Create a new file named **gui\_text.py** and copy the code shown below: |

|  |
| --- |
| 1. Run the program and observe the error. 2. Add an import QLineEdit to the Pycharm.Widgets import 3. Run the program and observe the output. 4. Add the following code below self.textbox.resize()   self.textbox.setText("Set this text value")  4. Run the program again then resize the textbox so that it fits in the Window and that its height is just above the written text’s height.  **Creating Labels**  1. Create a new file called **gui\_labels.py** and copy the following code below: |

|  |
| --- |
| 1. Run the program and observe the output. 2. Add the necessary Widget at the import line to make the program run. 3. Center the label by adjusting the parameters of .move(). This is called Absolute Positioning. 4. Create a new label called “This program is written in Pycharm” and place it at the center and below the Hello World! |
| **6. Supplementary Activity:** |
| **Task**  Create an Object-Oriented GUI Application for a simple Account Registration System with the following required information: first name, last name, username, password, email address, contact number.  Requirements:   * The GUI must be centered on your screen. * The GUI Components should be organized according to the order of information required using Absolute Positioning. * The position of the components should be automatically computed based on the top component. * All the text fields should be accompanied with their corresponding label on the left side while the text field is on the right side. * There should a program title other than the Window Title. * There should be a submit button and clear button at the bottom (submit button on the left, clear button on the right). * The program should be launched on **main.py** while the GUI Codes should be on a separate file called   **registration.py**  For the supplementary activity, please refer to this link <https://github.com/Leigh-Hermosura/CPE-103-OOP-1-A/tree/main/Laboratory-No.9/oopfa1_Hermosura_lab9/supplemetaryActivity> |

|  |
| --- |
| **Questions**   1. What are the common GUI Applications that general end-users such as home users, students, and office employees use? (give at least 3 and describe each)   One of the best examples would be videogames as it allows the user’s input to influence the game through its graphic interface. Second example is Microsoft Office Suite (MS Word, Excel, PowerPoint, etc.) where we can make documents, tabulations, and presentations using these software applications. Lastly, LMS or learning management systems like google classroom which students use to track their school works and access learning materials online.   1. Based from your answer in question 1, why do you think home users, students, and office employees use those GUI programs?   Based on my answer, people use GUI programs in a variety of ways, be it entertainment, work, or academics. GUI offers work efficiency like data tabulations by using specialized programs for these types of tasks and even as tools for entertainment. These programs make tasks faster, easier and manageable compared to traditional methods.   1. How does Pycharm help developers in making GUI applications, what would be the difference if developers made GUI programs without GUI Frameworks such as Pycharm or Tkinter?   Pycharm is a python Integrated Development Environment (IDE) that has a wide selection of different libraries. These libraries, like Tkinter or PyQt, contains built in methods and functions specialized in making GUI. The difference of not using GUI frameworks is that it would be more difficult and inefficient as you would have to create code from scratch or use a web application to serve as your GUI.   1. What are the different platforms a GUI program may be created and deployed on? (Three is required then state why might a program be created on that specific platform)   GUI progams can be created and deployed on IDEs like Pycharm, Visual Studio Code, JupyterLab, among many others, as many of them can be used to create GUI using programming languages. Game engines like Unity, Unreal Engine, and Godot are specifically made for creating videogames which are also a type of GUI. Lastly, desktop platforms like Windows, macOS, and Linux can run GUI programs by using GUI toolkits.   1. What is the purpose of **app = QApplication(sys.argv)**, **ex = App()**, and **sys.exit(app.exec\_())**?   App = QApplication initializes the program, preparing it to run, while the ex = App() creates the GUI of the program, and sys.exit(app.exec\_()) starts the event loop to keep the GUI running and allow a smooth exit when the loop ends or when you quit the program. |
| **7. Conclusion:** |
| GUI applications play a crucial role in simplifying tasks for home users, students, and office employees, offering a user-friendly interface for various activities such as entertainment, work, and academics. Programs like video games, Microsoft Office Suite, and Learning Management Systems provide efficient solutions for their respective tasks. Developers can create these applications with the help of GUI frameworks like Pycharm, Tkinter, and others, which offer predefined libraries that streamline the development process and enhance productivity. Without these frameworks, developers would face challenges in building GUI applications from scratch, leading to inefficiency. Additionally, GUI programs can be created and deployed on various platforms such as IDEs, game engines, and desktop operating systems, providing flexibility and adaptability. Lastly, key components of GUI application code, such as app = QApplication(sys.argv), ex = App(), and sys.exit(app.exec\_()), serve to initialize the program, create the GUI, and maintain the event loop for smooth execution and termination of the application. |
| **8. Assessment Rubric:** |