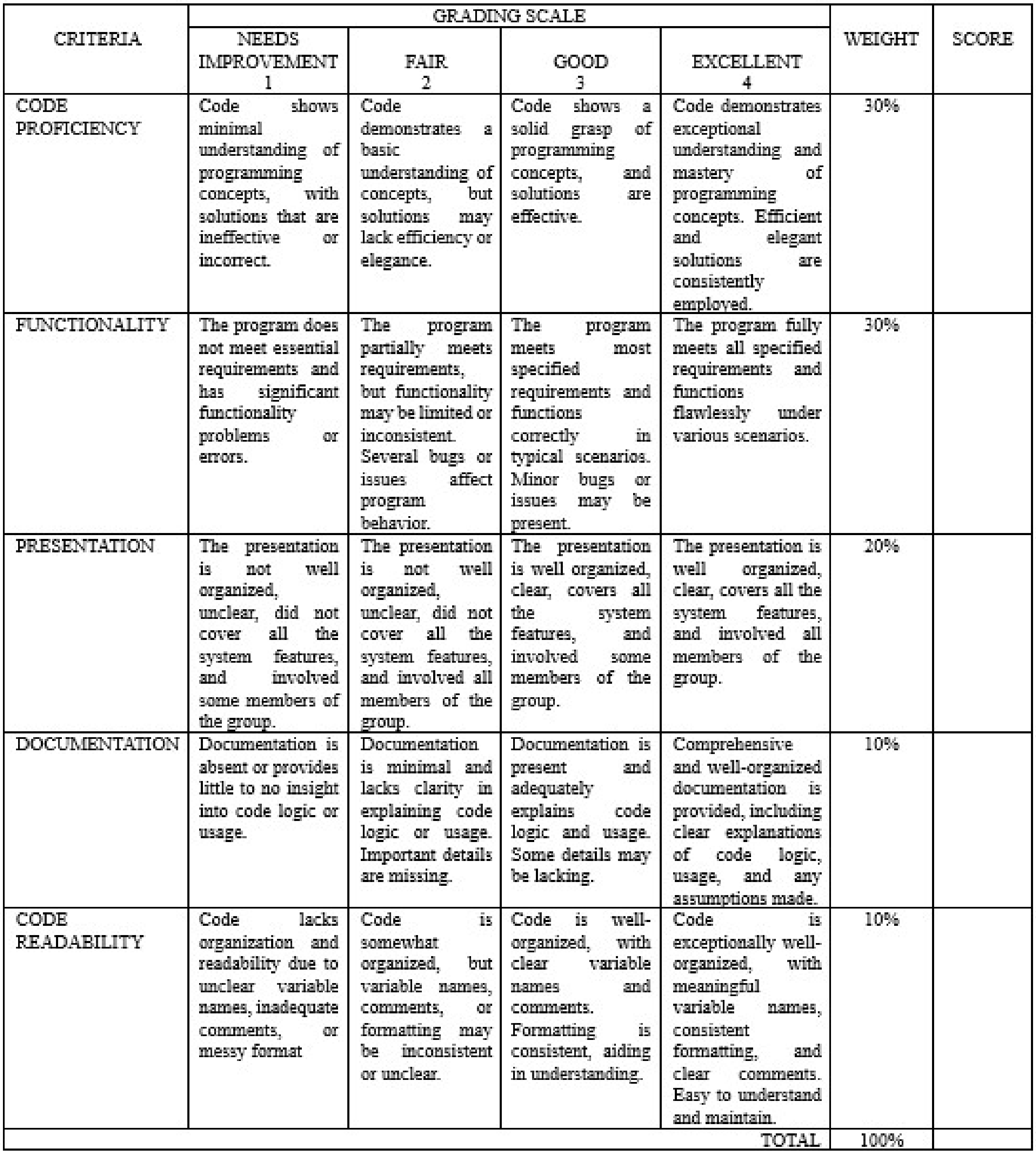
WEEK 5
Prelim Group Research Project

<table>
<tr><td><b>Group Leader (LN, FN MI.):</b></td><td>1</td><td colspan="3">CLASIN, MARYJE</td><td rowspan="5"><b>Score:</b></td><td rowspan="5"></td></tr>
<tr><td rowspan="9"><b>Group Members (LN, FN MI.):</b>
<b>In alphabetical Order</b></td><td>2</td><td colspan="3">GALLO, SHEENA MAE D,</td></tr>
<tr><td>3</td><td colspan="3">LANSANGAN, JOYCE E.</td></tr>
<tr><td>4</td><td colspan="3">LOBERIANO, MICHELLE J.</td></tr>
<tr><td>5</td><td colspan="3">TOLENTINO, KRISTINE MAE T.</td></tr>
<tr><td>6</td><td colspan="3">SORIANO, FRANCE HEDRICH O.</td><td rowspan="5"><b>Date:</b></td><td rowspan="5">09/29/23</td></tr>
<tr><td>7</td><td colspan="3">MIRANDA, KIAN ANDREI L</td></tr>
<tr><td>8</td><td colspan="3">REFREA, MARJOHN</td></tr>
<tr><td>9</td><td colspan="3">SULLA, JOHN PATRICK</td></tr>
<tr><td>10</td><td colspan="3">VILLENA, KEN ANTHONY J.</td></tr>
<tr><td><b>Section:</b></td><td colspan="2">2IT-D</td><td><b>Instructor:</b></td><td>Prof. Janus Raymond C. Tan</td><td><b>Sem./ A.Y.</b></td><td>1st/ 23-24</td></tr>
<tr><td><b>Project Link:</b></td><td colspan="6">https://www.codiva.io/p/aa7a700b-9458-4120-ba1c-c9b8ea4fcc43</td></tr>
<tr><td><b>Youtube Link:</b></td><td colspan="6"><i>Paste the link of your project presentation uploaded in <b>Youtube</b> that explains the features of the project followed by the explanation of your program code <b>(introduce the presenter, show the face of the presenter, and make sure all members should have explained something from the project)</b></i>
https://youtu.be/1cMUqUhSIgw</td></tr>
</table>



CRITERIA	GRADING SCALE				WEIGHT	SCORE
	NEEDS IMPROVEMENT 1	FAIR 2	GOOD 3	EXCELLENT 4		
CODE PROFICIENCY	Code shows minimal understanding of programming concepts, with solutions that are ineffective or incorrect.	Code demonstrates a basic understanding of concepts, but solutions may lack efficiency or elegance.	Code shows a solid grasp of programming concepts, and solutions are effective.	Code demonstrates exceptional understanding and mastery of programming concepts. Efficient and elegant solutions are consistently employed.	30%	
FUNCTIONALITY	The program does not meet essential requirements and has significant functionality problems or errors.	The program partially meets requirements, but functionality may be limited or inconsistent. Several bugs or issues affect program behavior.	The program meets most specified requirements and functions correctly in typical scenarios. Minor bugs or issues may be present.	The program fully meets all specified requirements and functions flawlessly under various scenarios.	30%	
PRESENTATION	The presentation is not well organized, unclear, did not cover all the system features, and involved some members of the group.	The presentation is not well organized, unclear, did not cover all the system features, and involved all members of the group.	The presentation is well organized, clear, covers all the system features, and involved some members of the group.	The presentation is well organized, clear, covers all the system features, and involved all members of the group.	20%	
DOCUMENTATION	Documentation is absent or provides little to no insight into code logic or usage.	Documentation is minimal and lacks clarity in explaining code logic or usage. Important details are missing.	Documentation is present and adequately explains code logic and usage. Some details may be lacking.	Comprehensive and well-organized documentation is provided, including clear explanations of code logic, usage, and any assumptions made.	10%	
CODE READABILITY	Code lacks organization and readability due to unclear variable names, inadequate comments, or messy format	Code is somewhat organized, but variable names, comments, or formatting may be inconsistent or unclear.	Code is well-organized, with clear variable names and comments. Formatting is consistent, aiding in understanding.	Code is exceptionally well-organized, with meaningful variable names, consistent formatting, and clear comments. Easy to understand and maintain.	10%	
				TOTAL	100%	

INSTRUCTION:

In this group research project, you will create a Java class representing a SmartPhone, simulating some of its functionalities using class fields and methods, and practice constructor overloading to initialize the smartphone object. Your task is to design and implement a SmartPhone Class that simulates the behaviors of a smartphone. The smartphone should have attributes and methods to perform actions. The more functionalities or features of the project, the better.

A menu-driven program using Java is required to simulate the creation of the SmartPhone Object as well as the way to allow the user to operate the created SmartPhone object. There should be an option for the user to select on first run or whenever he repeats the entire program execution to choose which among the constructors available will be used to create the SmartPhone object. Remember that the program should be dynamic, allowing the user to enter the SmartPhone details. After successful object creation, the program should allow the user to perform operations on the created SmartPhone object by selecting among the list of operations in a menu. Whenever there is a change in the attribute value involved in such operation, always display the details of the created SmartPhone object. There should be a way to repeat the creation of the SmartPhone object and perform again its operations. The program will only stop once the user tries to select an EXIT Menu item. Test your program in a separate class named Main.java that will contain a main method to execute the program.

PROGRAM CODE

File: SmartPhone.java

```
 import java.util.Scanner;

public class Main {
      public static void main(String[] args) {

           Scanner scanner = new Scanner(System.in);

           Smartphone objSmartphone = new Smartphone();
        objSmartphone.confirmation = false;


        while (objSmartphone.check == true) {
            System.out.println("Menu:");
            System.out.println(" 1. Create SmartPhone");
            System.out.println(" 2. Install Application");
            System.out.println(" 3. Power Button");
            System.out.println(" 4. Charge Phone ");
            System.out.println(" 5. Display SmartPhone Details");
            System.out.println(" 6. Check Battery %");
            System.out.println(" 7. Use Facebook");
            System.out.println(" 8. Use Messenger");
            System.out.println(" 9. Uninstall Application");
            System.out.println("10. Exit");
           System.out.print("Enter your choice: ");
           int choice = scanner.nextInt();
           System.out.print("\n");

            switch (choice) {
                //Decision for Create SmartPhone
                case 1:
                Scanner scanner1 = new Scanner (System.in);
                    System.out.println("-Please select your preferred
choice-");
                    System.out.println("1. Default Smartphone" + "\n"
+ "2. Customize (Owner)" + "\n" + "3. Customize (Owner, Brand, Model)"
                                     + "\n" + "4. Customize (Owner,
Brand, Model, Storage, & RAM )"  );
                      System.out.print("Enter your choice: ");
                int chosenConstructor = scanner.nextInt();
                System.out.print("\n");
                      if (chosenConstructor == 1) {
                          objSmartphone.confirmation = true;
                        break;

                      }
                      else if (chosenConstructor == 2) {
                        System.out.print("Name of Owner: ");
                        objSmartphone.SmartphoneOwner =
scanner1.nextLine();
                          objSmartphone.confirmation = true;
                        System.out.println();
                        break;

                      }
                      else if (chosenConstructor == 3) {
                        System.out.print("Name of Owner: ");
                        objSmartphone.SmartphoneOwner =
scanner1.nextLine();
                        System.out.print("Phone Brand: ");
                        objSmartphone.SmartphoneBrand =
scanner1.nextLine();
                        System.out.print("Phone Model: ");
                        objSmartphone.SmartphoneModel =
scanner1.nextLine();
                          objSmartphone.confirmation = true;
                        System.out.println();
                        break;

                      }
                      else if (chosenConstructor == 4) {
                        System.out.print("Name of Owner: ");
                          objSmartphone.SmartphoneOwner =
scanner1.nextLine();
```

```
                          System.out.print("Phone Brand: ");
                          objSmartphone.SmartphoneBrand = 
scanner1.nextLine();
                          System.out.print("Phone Model: ");
                          objSmartphone.SmartphoneModel = 
scanner1.nextLine();
                          System.out.print("Phone Storage (GB): ");
                          objSmartphone.SmartphoneStorage = 
scanner1.nextInt();
                          System.out.print("Phone Ram: ");
                          objSmartphone.SmartphoneRam = 
scanner1.nextInt();
                          objSmartphone.confirmation = true;
                          System.out.println();
                          break;
                     }
                      else {
                        System.out.println("Invalid Input" + "\n");
                        break;
                     }

                 //Decision for Install Application
                case 2:
                      if (objSmartphone.confirmation == true & 
objSmartphone.powerOn == true & objSmartphone.check == true) {
                           objSmartphone.InstallApp();
                           break;

                       } else if (objSmartphone.powerOn == false & 
objSmartphone.confirmation == true) {
                                 System.out.println("Please turn on the 
" + objSmartphone.SmartphoneBrand + " phone first");
                                 break;

                     } else {
                                 System.out.println("Please create a 
SmartPhone first");
                                 break;
                                 }
                  //Decision for Power Button
                 case 3:
                    if (objSmartphone.confirmation == true & 
objSmartphone.check == true) {
                      objSmartphone.SwitchPower();
                        break;

                    } else if(objSmartphone.powerOn == false & 
objSmartphone.check == true) {
```

```
                        System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                        break;

                    } else {
                        System.out.println("Please create a SmartPhone
first.");
                        break;
                    }

                 //Decision for Charge Phone
                 case 4:
                    if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                        objSmartphone.Charging();
                        break;

                    } else if(objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                        System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                        break;
                    }
                   else {
                        System.out.println("Please create a SmartPhone
first");
                        break;
                      }

                  //Decision for Display SmartPhone Details
                 case 5:
                if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                    objSmartphone.displayDetails();
                           break;


                }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                      System.out.println(objSmartphone.SmartphoneBrand
+ " must be powered on first");
                        break;
                }
                else {
                      System.out.println("Please create a SmartPhone
first.");
                        break;
                  }
                 //Decision for  Check Battery %
                case 6:
                      if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                           objSmartphone.CheckBatteryLevel();
                        break;


                      }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {

     System.out.println(objSmartphone.SmartphoneBrand + " must be
powered on first");
                           break;
                      }
                      else {
                        System.out.println("Please create a SmartPhone
first.");
                        break;

                      }

                 //Decision for Use Facebook
                case 7:
                      if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                           objSmartphone.OpenFacebook();
                        break;


                      }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                           System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                           break;
                      }
                      else {
                           System.out.println("Please create a
SmartPhone first.");
                           break;

                      }

                //Decision for Use Messenger
                case 8:

                      if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                            objSmartphone.OpenMessenger();
                      break;


                      }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                            System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                            break;
                      }
                      else {
                            System.out.println("Please create a
SmartPhone first.");
                            break;

                      }

                 //Decision for Uninstall Application
                case 9:

                      if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                            objSmartphone.UninstallApp();
                            break;


                      }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                            System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                            break;
                      }
                      else {
                            System.out.println("Please create a
SmartPhone first.");
                            break;

                      }

                 //Decision for Exit
                case 10:
                      if (objSmartphone.confirmation == true &
objSmartphone.powerOn == true & objSmartphone.check == true) {
                            objSmartphone.ExitMenu();
                            break;
                      }else if (objSmartphone.powerOn == false &
objSmartphone.confirmation == true) {
                           System.out.println("Please turn on the " +
objSmartphone.SmartphoneBrand + " phone first");
                           break;
                      }
                      else {
                           System.out.println("Please create a
SmartPhone first.");
                           break;

                      }



           }

           }
        }
     }
```

File: Main.java

```
import java. util.Scanner;

    public class Smartphone {
     //Declaring class fields
           String SmartphoneBrand;
           String SmartphoneModel;
           int SmartphoneStorage;
           int SmartphoneRam;
           String SmartphoneOwner;
           boolean powerOn;
        boolean confirmation;
        int batteryLevel;
        boolean facebookAppInstalled;
        boolean messengerAppInstalled;
        String status;
        String currentMessage;
           boolean check;

     //declaring default constructor
    public Smartphone () {
          this.SmartphoneBrand = "Xiaomi";
          this.SmartphoneModel = "Redmi Note 10 Pro";
          this.SmartphoneStorage = 128;
          this.SmartphoneRam = 8;
          this.SmartphoneOwner = "Althea";
          this.powerOn = false;
       this.confirmation = false;
       this.batteryLevel = 20;
       this.facebookAppInstalled = false;
       this.messengerAppInstalled = false;
```

```
        this.status = "";
        this.currentMessage = "";
        this.check = true;
     }
     //Declaring overloding constructor
     public Smartphone (String SmartphoneOwner) {
           this.SmartphoneBrand = "Xiaomi";
           this.SmartphoneModel = "Redmi Note 10 Pro";
           this.SmartphoneStorage = 128;
           this.SmartphoneRam = 8;
           this.SmartphoneOwner = "SmartphoneOwner";
        this.powerOn = false;
        this.confirmation = false;
        this.batteryLevel = 20;
        this.facebookAppInstalled = false;
        this.messengerAppInstalled = false;
        this.status = "";
        this.currentMessage = "";
        this.check = true;

     }
     //Declaring overloding constructor
     public Smartphone (String SmartphoneBrand, String
SmartphoneModel, String SmartphoneOwner ) {
           this.SmartphoneBrand = SmartphoneBrand;
           this.SmartphoneModel = SmartphoneModel;
           this.SmartphoneStorage = 128;
           this.SmartphoneRam = 8;
           this.SmartphoneOwner = SmartphoneOwner;
        this.powerOn = false;
        this.confirmation = false;
        this.batteryLevel = 20;
        this.facebookAppInstalled = false;
        this.messengerAppInstalled = false;
        this.status = "";
        this.currentMessage = "";
        this.check = true;


     }
     //Declaring overloding constructor
     public Smartphone (String SmartphoneBrand, String
SmartphoneModel, int SmartphoneStorage, int SmartphoneRam, String
SmartphoneOwner ) {
           this.SmartphoneBrand = SmartphoneBrand;
           this.SmartphoneModel = SmartphoneModel;
           this.SmartphoneStorage = SmartphoneStorage;
           this.SmartphoneRam = SmartphoneRam;
           this.SmartphoneOwner = SmartphoneOwner;
        this.powerOn = false;
        this.confirmation = false;
        this.batteryLevel = 20;
        this.facebookAppInstalled = false;
        this.messengerAppInstalled = false;
        this.status = "";
        this.currentMessage = "";
        this.check = true;



      }



      //Declaring Method for SwitchPower
       public void SwitchPower() {
       Scanner scannerForSwitchPower = new Scanner(System.in);
             if (this.powerOn == true & this.confirmation == true) {
                System.out.println("The device " +
this.SmartphoneBrand + "," + " Model: " +  this.SmartphoneModel + "\n"
+
                                         "owned by " +
this.SmartphoneOwner + ", is powered on" + "\n" );
                 System.out.println("Enter 0 to Turn it Off --- Enter
2 to return to the menu");
                 int answerPower = scannerForSwitchPower.nextInt();
                       if (answerPower == 0) {
                           this.powerOn = false;
                           SwitchPower();
                     }

                     else if (answerPower == 2) {
                        System.out.println("Bringing to Menu....");
                     }

                       else {
                           System.out.println(this.SmartphoneBrand + "
is unable to turn on" + "\n");
                       }


            } else if (this.powerOn == false) {
                 System.out.println("The device " +
this.SmartphoneBrand + "," + " Model: " +  this.SmartphoneModel + "\n"
+  "owned by " + this.SmartphoneOwner + ", is powered off" + "\n");
                System.out.println("Enter 1 to Turn it On --- Enter 2
to return to the menu");
```

```
                 int answerPower = scannerForSwitchPower.nextInt();
                       if (answerPower == 1) {
                            this.powerOn = true;
                            SwitchPower();
                      }

                      else if (answerPower == 2) {
                         System.out.println("Bringing to Menu....");
                      }

                       else {
                            System.out.println(this.SmartphoneBrand + "
is unable to turn on" + "\n");
                         }

            } else if (this.powerOn == false && confirmation == true )
{
                  System.out.println("Please turn on the device
first");

            } else {
              System.out.println(this.SmartphoneBrand + "-must be
powered on first-");
       }
     }

      //Declaring method for Charging
      public void Charging() {
        System.out.println("How many minutes you want to charge your
phone?" +  "\n" +
                            "Every minute is = 1%");
        Scanner scannerForCharging = new Scanner(System.in);
        int answerCharge = scannerForCharging.nextInt();

        if (this.powerOn == true) {
            if (batteryLevel + answerCharge <= 100) {
                batteryLevel += answerCharge;
                System.out.println(this.SmartphoneBrand + " is now
charged to " + batteryLevel + "%.");

            }
            else {
                System.out.println(this.SmartphoneBrand + " cannot be
charged beyond 100%.");
            }
        }
        else {
            System.out.println(this.SmartphoneBrand + " must be
powered on to charge.");
        }
    }

      //Declaring method for CheckBatteryLevel
      public void CheckBatteryLevel() {
          System.out.println(this.SmartphoneBrand + " has a battery
level of " + this.batteryLevel + "%.");
      }
         //Declaring method for InstallApp
      public void InstallApp() {
        System.out.println("Press 1 to Install Facebook" + "\n" +
"Press 2 to Install Messenger");
        Scanner scannerForInstallApp = new Scanner(System.in);
          int answerIntallApp = scannerForInstallApp.nextInt();
          if(answerIntallApp == 1) {
             this.facebookAppInstalled = true;
              System.out.println("Facebook app installed on " +
this.SmartphoneBrand + ".");

          }
          else if (answerIntallApp == 2) {
             this.messengerAppInstalled = true;
              System.out.println("Messenger app installed on " +
this.SmartphoneBrand + ".");
          }
          else {
             System.out.println("Unable to install application on " +
this.SmartphoneBrand + ".");
          }
      }

      //Declaring method for UninstallApp
      public void UninstallApp() {
        System.out.println("Press 1 to Uninstall Facebook" + "\n" +
"Press 2 to Uninstall Messenger");
        Scanner scannerForUninstallApp = new Scanner(System.in);
          int answerUninstallApp = scannerForUninstallApp.nextInt();
          if(answerUninstallApp == 1) {
             this.facebookAppInstalled = false;
              System.out.println("Facebook app uninstalled on " +
this.SmartphoneBrand + ".");

          }
          else if (answerUninstallApp == 2) {
             this.messengerAppInstalled = false;
              System.out.println("Messenger app uninstalled on " +
this.SmartphoneBrand + ".");
```

```
     }
     else {
       System.out.println("Unable to uninstall application on "
+ this.SmartphoneBrand + ".");
     }
   }

    //Declaring method for OpenFacebook
   public void OpenFacebook() {
     Scanner scannerForOpenFacebook = new Scanner(System.in);

     if (this.facebookAppInstalled == true & this.powerOn ==
true) {
       System.out.println("Facebook app is now open");

      for (int a = 0; a <= 20 ; a++) {

       System.out.println("Choose what you wish to do" + "\n" +
                      "1. Post Status" + "\n" +
"2. Like post" + "\n" + "3. Comment on post" + "\n" + "4. Close
Facebook" );
       int answerOpenFacebook1 =
scannerForOpenFacebook.nextInt();

       switch (answerOpenFacebook1) {

       case 1:
         PostStatus();
         break;
       case 2:
         LikePost("Mich");
         break;
       case 3:
         CommentOnPost("Sheena","TARA GUYS TAYO AY MAG FARMER
NALANG SA JAPAN TANGGAL NA UTAK KO SA PAGCOCODE");
         break;
       case 4:
         CloseFacebook();
         a = 20;
         break;

       default:
         System.out.println("Invalid input");
         a = 20;
         break;
       }
     }
   }
```

```
    else {
       System.out.println("Facebook app is not installed.
Please install it first");
       System.out.println("Enter 0 to return to the menu");
       int answerOpenFacebook2 =
scannerForOpenFacebook.nextInt();
         if (answerOpenFacebook2 == 0) {
           System.out.println("Bringing to Menu....");
          }

         else {
              System.out.println(this.SmartphoneBrand + "
is unable to turn on" + "\n");
         }
      }
   }
   //Declaring method for PostStatus
   public void PostStatus() {
        Scanner scannerForPostStatus = new Scanner(System.in);
        System.out.print("Posting status on Facebook: ");
     this.status = scannerForPostStatus.nextLine();
     System.out.println("You posted: " + this.status);
     }

   //Declaring method with parameters
   public void LikePost(String reactname ) {
     System.out.println(reactname + " Reacted like to your post "
+ "'" + this.status + "'" + " on Facebook");
      }

     //Declaring method with parameters
   public void CommentOnPost(String reactname2, String comment) {
     System.out.println(reactname2 + " Commented on your post " +
"'" + status + "'"+ "\nComment: " + comment);
     }

     //Declaring method for CloseFacebook
   public void CloseFacebook() {
     System.out.println("Closing Facebook app.");
     }

    //Declaring method for OpenMessenger
  public void OpenMessenger() {
   Scanner scannerForOpenMessenger = new Scanner(System.in);

    if (messengerAppInstalled == true) {
       System.out.println("Messenger app is now open.");
```

```
              for (int a = 0; a <= 20 ; a++) {

                  System.out.println("Choose what you wish to do" +
"\n" +
                                           "1. Received Message" +
"\n" + "2. Check Inbox" + "\n" + "3. Close Messenger");
                  int scannerForOpenMessenger1 =
scannerForOpenMessenger.nextInt();

                  switch (scannerForOpenMessenger1) {

                  case 1:
                  ReceivedMessage("Patricia: ",  "Happy Birthday!");
                  break;
                  case 2:
                  CheckInbox("Liam ", "Carps?" );
                  break;
                  case 3:
                  CloseMessenger();
                  a = 20;
                  break;
                  default:
                  System.out.println("Invalid input");
                      a = 20;
                  break;
                  }
             }

          } else {
             System.out.println("Messenger app is not installed.
Please install it first");
              System.out.println("Enter 0 to return to the menu");

              int scannerForOpenMessenger2 =
scannerForOpenMessenger.nextInt();
                 if (scannerForOpenMessenger2 == 0) {
                      System.out.println("Bringing to Menu....");
                  }

                 else {
                           System.out.println(this.SmartphoneBrand + "
is unable to turn on" + "\n");
                 }
           }
      }
            //Declaring method with parameters
      public void ReceivedMessage(String contact, String message) {
          if (this.messengerAppInstalled && this.powerOn == true) {
              System.out.println(contact + ": " + message);
              currentMessage = message;

          } else {
              System.out.println("Messenger app is not installed or 
the smartphone is not powered on.");
          }
      }

      //Declaring method with parameters
      public void CheckInbox(String contact, String message) {
         if (messengerAppInstalled == true) {
             System.out.println(contact + ": " + message);
              System.out.println("You: " + currentMessage);
              batteryLevel--;
          } else {
              System.out.println("Messenger app is not installed or 
the smartphone is not powered on.");
          }
      }
           //Declaring method for CloseMessenger
      public void CloseMessenger() {
          if (messengerAppInstalled == true & powerOn == true) {
            System.out.println("Closing Messenger app.");
              batteryLevel--;
          } else {
              System.out.println("Messenger app is not installed or 
the smartphone is not powered on.");
          }
      }
      //Declaring method for ExitMenu
      public void ExitMenu() {
     if (powerOn == true & confirmation == true & check == true) {
           System.out.println(this.SmartphoneBrand + " is exiting the 
operation." + "\n" + "GOODBYE!");
           this.check = false;

     }
        else {
           System.out.println(this.SmartphoneBrand + " is alredy 
EXIT!");
        }
    }
      //Declaring method for displayDetails
      public void displayDetails() {
       if (powerOn == true & check == true) {
         System.out.println("Brand: " + this.SmartphoneBrand + "\n" + 
"Model: " + this.SmartphoneModel +  "\n" +
                                      "Storage: " +
this.SmartphoneStorage + " GB" + "\n" + "RAM: " + this.SmartphoneRam +
"\n" +
                                      "Welcome, " +
this.SmartphoneOwner + "!" + "\n");
          }
       else if (this.confirmation == false) {
         System.out.println("Please create smartphone first");


       }

       else {
         System.out.println(this.SmartphoneBrand + " must be powered
on first");
       }
     }
   }
```

WHAT HAVE YOU LEARNED?

There are many things I've learned in this project since I learned how to set up the switch case in all of the methods we construct, thus in this program, the user will choose an option from the menu, but first he or she must run the program. The menu will then be displayed in the output. I realize that if we did not use the switch case statement, the user would see each method one by one, similar to how the print of all the methods would look, but with the switch case statement, we only input the number of the option and it will function immediately. And I learned how to build up and use the overloaded constructor in this code.

Attach your own PDF file **(every group member should turn in his/ her own copy)** as your submission in the assigned task upon turning in using your own account in our LMS following the given template. **(your PDF file should begin with the first page of the given template)**