Algorithms and Data Structures

Assessment 2

System Class Source Code:

//Part A of Algorithms and Data Structures Assessment 2

//imports

**import** java.util.Properties;

**import** java.util.Scanner;

**class** System {

//attributes

**private** **static** String make;

**private** **static** String model;

**public** **static** String answer2;

**public** **static** String answer3;

**public** **static** String answer4;

**private** **static** String checkHDStatus;

**private** **static** **int** speed;

**private** **static** **int** memorySize;

**private** **static** **double** hardDiskSize;

**private** **static** **double** purchaseCost;

**public** **static** **boolean** goodMemorySize;

**public** **static** **boolean** valid = **false**;

**public** **static** **double** input;

**public** **enum** OSType { Windows, Linux, Other};

**protected** **static** OSType detectedOS;

**public** **static** Scanner keyboard = **new** Scanner(java.lang.System.in);

// the constructor

**public** System(String modelIn, String makeIn, **int** speedIn)

{

make = makeIn;

model = modelIn;

speed = speedIn;

}

// methods to return attribute values

**public** String getMake()

{

**return** make;

}

**public** String getModel()

{

**return** model;

}

**public** **int** getSpeed() {

**return** speed;

}

// methods to set the corresponding attributes

**public** **void** setMemory(**int** memorySizeIn)

{

memorySize = memorySizeIn;

}

**public** **static** **void** setHardDisk(**double** hardDiskSizeIn)

{

hardDiskSize = hardDiskSizeIn;

}

**public** **void** setPurchaseCost(**double** purchaseCostIn)

{

purchaseCost = purchaseCostIn;

}

**public** **static** **int** setMemory() {

// try statement used here for exception handling

{

**do** {

**try** {

java.lang.System.out.println("Set system memory (MB) :");

System.memorySize = keyboard.nextInt();

**if** (**true**)

java.lang.System.out.println("");

**break**;

} **catch** (Exception e) {

java.lang.System.out.println("Invalid input");

java.lang.System.out.println();

java.lang.System.out.println("System Exit.");

java.lang.System.exit(0);

}

} **while** (**false**);

}

**return** System.memorySize;

}

**public** **static** **double** setHardDisk() {

// try statement used here for exception handling

{

**do** {

**try** {

java.lang.System.out.println("Set hard disk size (GB):");

System.hardDiskSize = keyboard.nextDouble();

**if** (**true**)

java.lang.System.out.println("");

**break**;

} **catch** (Exception e) {

java.lang.System.out.println("Invalid input");

java.lang.System.out.println();

java.lang.System.out.println("System Exit.");

java.lang.System.exit(0);

}

} **while** (**false**);

}

**return** hardDiskSize;

}

**public** **static** **double** setPurchaseCost() {

System.purchaseCost = 200;

**return** purchaseCost;

}

//check if the hard disk size is below 2GB and print corresponding message.

**public** **static** String checkHDStatus() {

**if** (System.hardDiskSize < 2)

**return** checkHDStatus=("low");

**if** (System.hardDiskSize > 2);

**return** checkHDStatus=("OK");

} //check if the memory size is below 128 MB and return false if so or true if higher.

**public** **static** **boolean** goodMemorySize()

{

**if** (memorySize > 128)

goodMemorySize = **true**;

**if** (memorySize < 128)

goodMemorySize = **false**;

**return** goodMemorySize;

}

// use the checkHDStatus() method and the goodMemorySize() method to diagnose the

// system and display the appropriate messages

**public** **static** String diagnoseSystem() {

java.lang.System.out.println("");

java.lang.System.out.println("\*\*\*\*\*System Diagnosis\*\*\*\*\*");

java.lang.System.out.println("");

java.lang.System.out.println("Hard disk size =" + System.checkHDStatus());

java.lang.System.out.println("Memory Size OK =" + System.goodMemorySize());

**return** (**null**);

}

//method to display system details

**public** **static** String displayDetails() {

System System = **new** System("Ezpad 6 pro", "Shenzhen Jumper Technology Co., Ltd.", 1600000000);

java.lang.System.out.println("");

java.lang.System.out.println("\*\*\*\*\*Computer Details\*\*\*\*\*");

java.lang.System.out.println("");

java.lang.System.out.println("System Make =" + System.make);

java.lang.System.out.println("");

java.lang.System.out.println("System Model =" + System.model);

java.lang.System.out.println("");

java.lang.System.out.println("System Speed =" + System.speed + " hz");

java.lang.System.out.println("");

java.lang.System.out.println("System Purchase Cost= £" + System.setPurchaseCost());

java.lang.System.out.println("");

**return** **null**;

}

//Part B of Algorithms and Data Structures Assessment

// method to display system properties

// positive message displayed if OS = Windows 10

**public** **static** String displaySystemProperties() {

System System = **new** System("Ezpad 6 pro", "Shenzhen Jumper Technology Co., Ltd.", 1600000000);

java.lang.System.out.println("");

java.lang.System.out.println("\*\*\*\*\*System Properties\*\*\*\*\*");

java.lang.System.out.println("");

java.lang.System.out.println("Os name = " + java.lang.System.getProperty("os.name"));

java.lang.System.out.println("Os version = " + java.lang.System.getProperty("os.version"));

java.lang.System.out.println("Os arch = " + java.lang.System.getProperty("os.arch"));

java.lang.System.out.println("User name = " + java.lang.System.getProperty("user.name"));

java.lang.System.out.println("Java version = " + java.lang.System.getProperty("java.version"));

**if** (detectedOS == **null**) {

String OS = java.lang.System.getProperty("os.name", "generic");

**if** ((OS.indexOf("Windows 10") >= 0)) {

detectedOS = OSType.Windows;

java.lang.System.out.println("");

java.lang.System.out.println("Thumbs up! Your operating system is Windows 10");

java.lang.System.out.println("");

java.lang.System.out.println("");

java.lang.System.out.println("");

}

**else** **if** (OS.indexOf("nux") >= 0) {

detectedOS = OSType.Linux;

java.lang.System.out.println("");

java.lang.System.out.println("Thumbs down! Your operating system is Linux");

}

**else** {

detectedOS = OSType.Other;

java.lang.System.out.println("");

java.lang.System.out.println("Your choice of operating system is ok");

}

}

**return** System.toString();

}

// method to set system memory size and hard disk size by calling previous 2 methods

// setMemory(); and setHardDisk();

// Printed statements already included in these methods

// to display the appropriate messages.

**public** **static** **void** setDetails() {

{ {setHardDisk(); } }

setMemory();

}

//Part C of Algorithms and Data Structures Assessment 2

// method to display system textual menu.

// method is repeated until the user quits

**public** **static** **void** TextualMenu() {

// do whiled loop used

**while** (!valid) {

**do** {

// printed statements for menu options

java.lang.System.out.println("|Options: |");

java.lang.System.out.println("| Choice 1 > Print System Details |");

java.lang.System.out.println("| Choice 2 > Display System Properties |");

java.lang.System.out.println("| Choice 3 > Diagnose System |");

java.lang.System.out.println("| Choice 4 > Set Details |");

java.lang.System.out.println("| Choice 5 > Quit the program |");

java.lang.System.out.println("");

java.lang.System.out.println("Mark your selection 1-5 and press enter:");

// Try statement for exception handling

{

**try** {

input = keyboard.nextDouble();

}

**catch** (Exception e){

java.lang.System.out.println("Invalid Input");

**break**;

} }

**if** (input == 1) {

java.lang.System.out.println("");

displayDetails(); }

**if** (input == 2) {

java.lang.System.out.println("");

displaySystemProperties(); }

**if** (input == 3) {

java.lang.System.out.println("");

diagnoseSystem(); }

**if** (input == 4) {

java.lang.System.out.println("");

setDetails();

}

**if** (input == 5) {

java.lang.System.out.println("");

java.lang.System.out.println("Exit.");

**return**;

}

**else** **if** (input < 1 && input > 5) { {

java.lang.System.out.println("Invalid input.");

} }

java.lang.System.out.println("");

java.lang.System.out.println("Please press m to return or x to exit.");

java.lang.System.out.println("");

java.lang.System.out.println("Mark your selection and press enter:");

answer2 = keyboard.next();

}

**while** (answer2.equals("m"));

**if** (answer2.equals("x"))

java.lang.System.out.println();

java.lang.System.out.println("System Exit.");

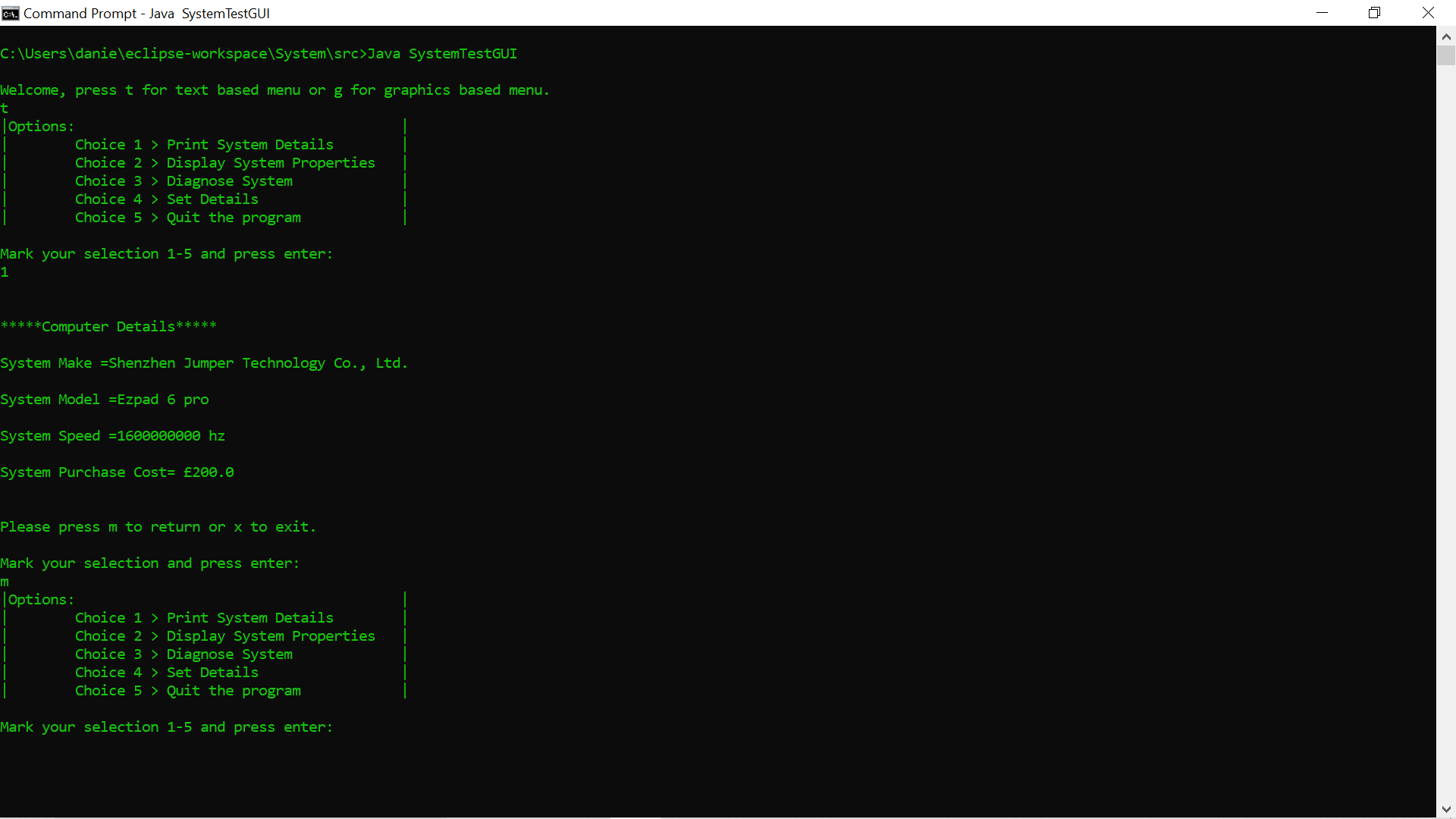
**return**;

}

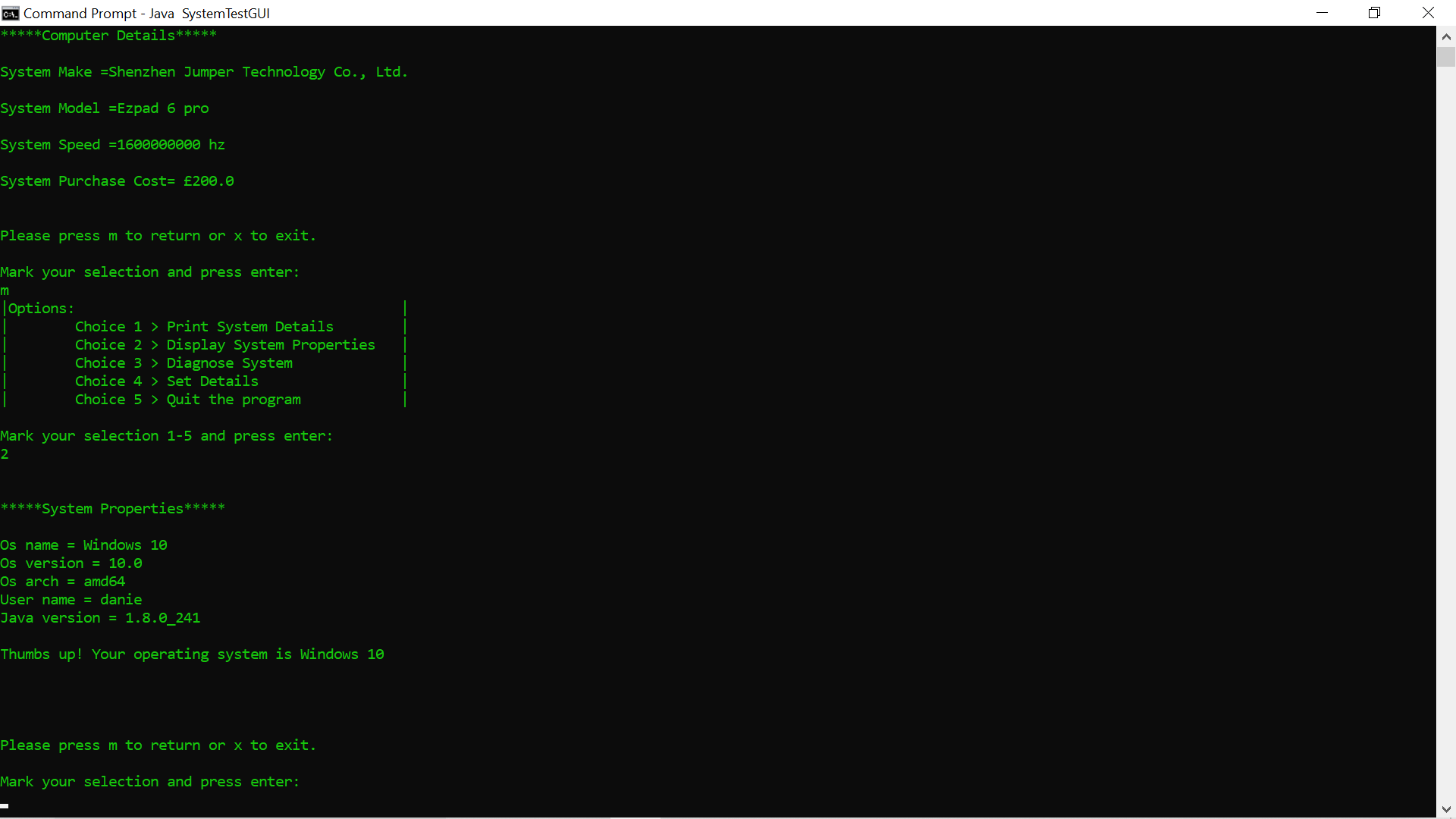
} }

System Class List of Outputs:

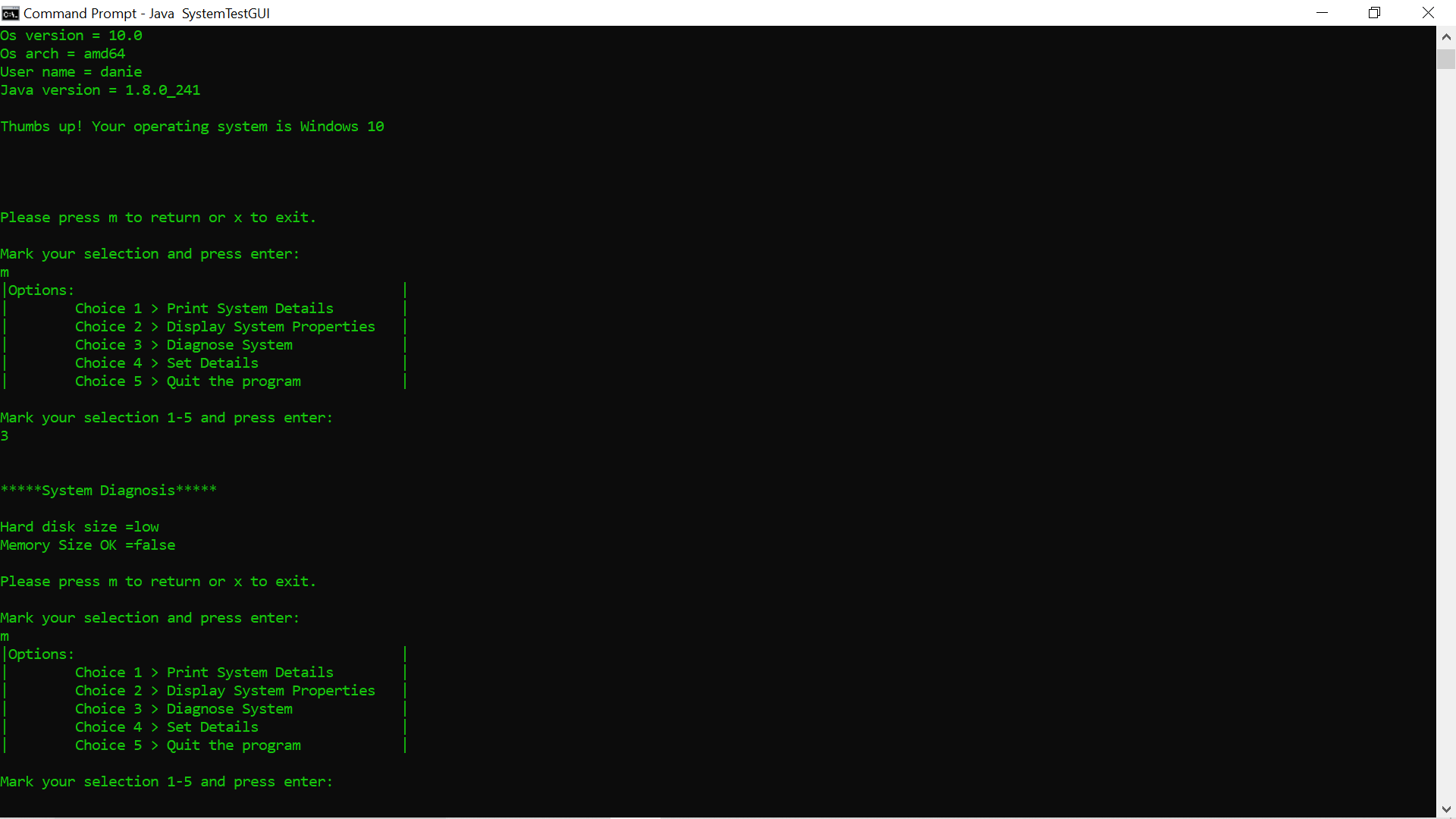
1)



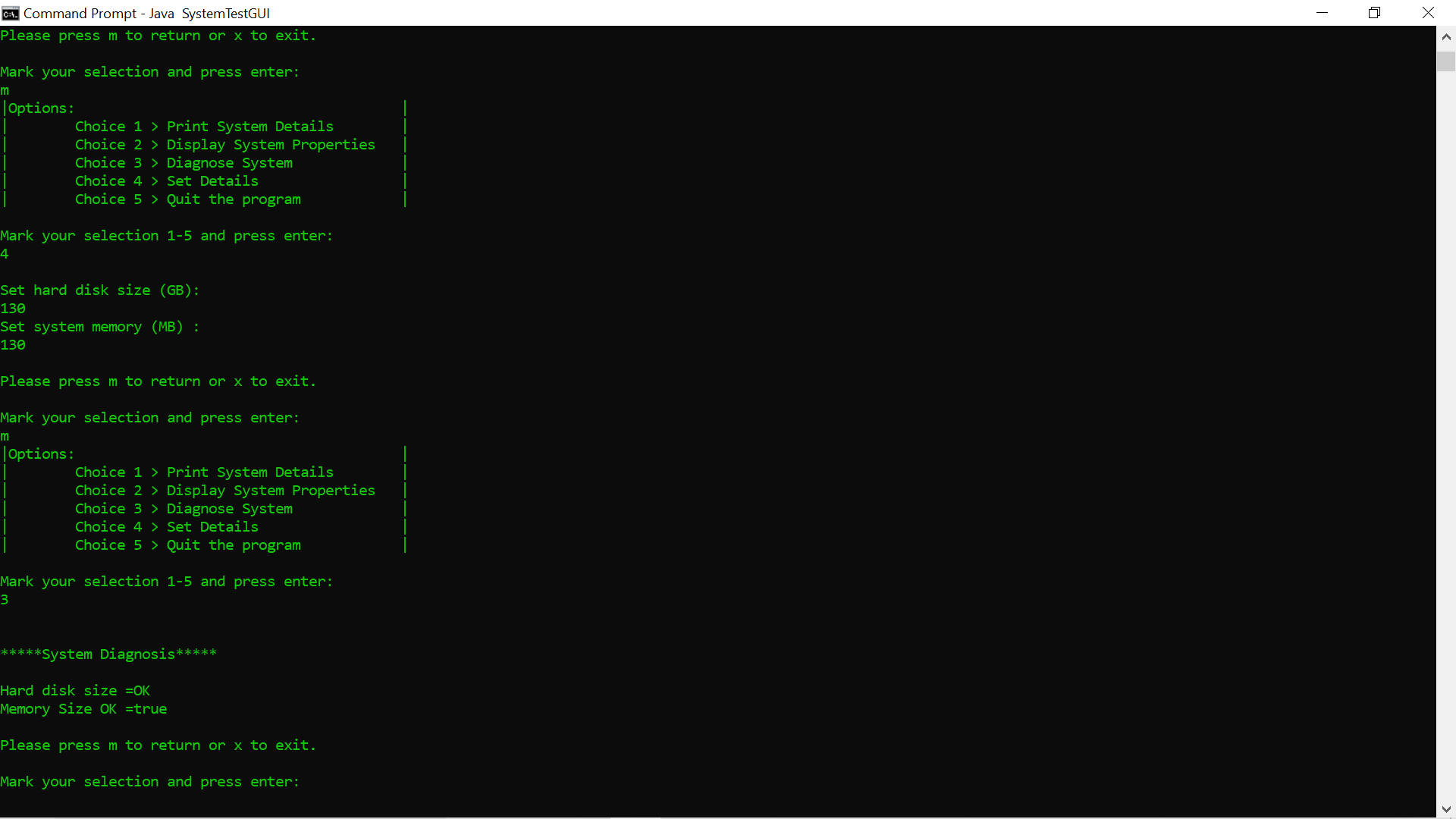
2)



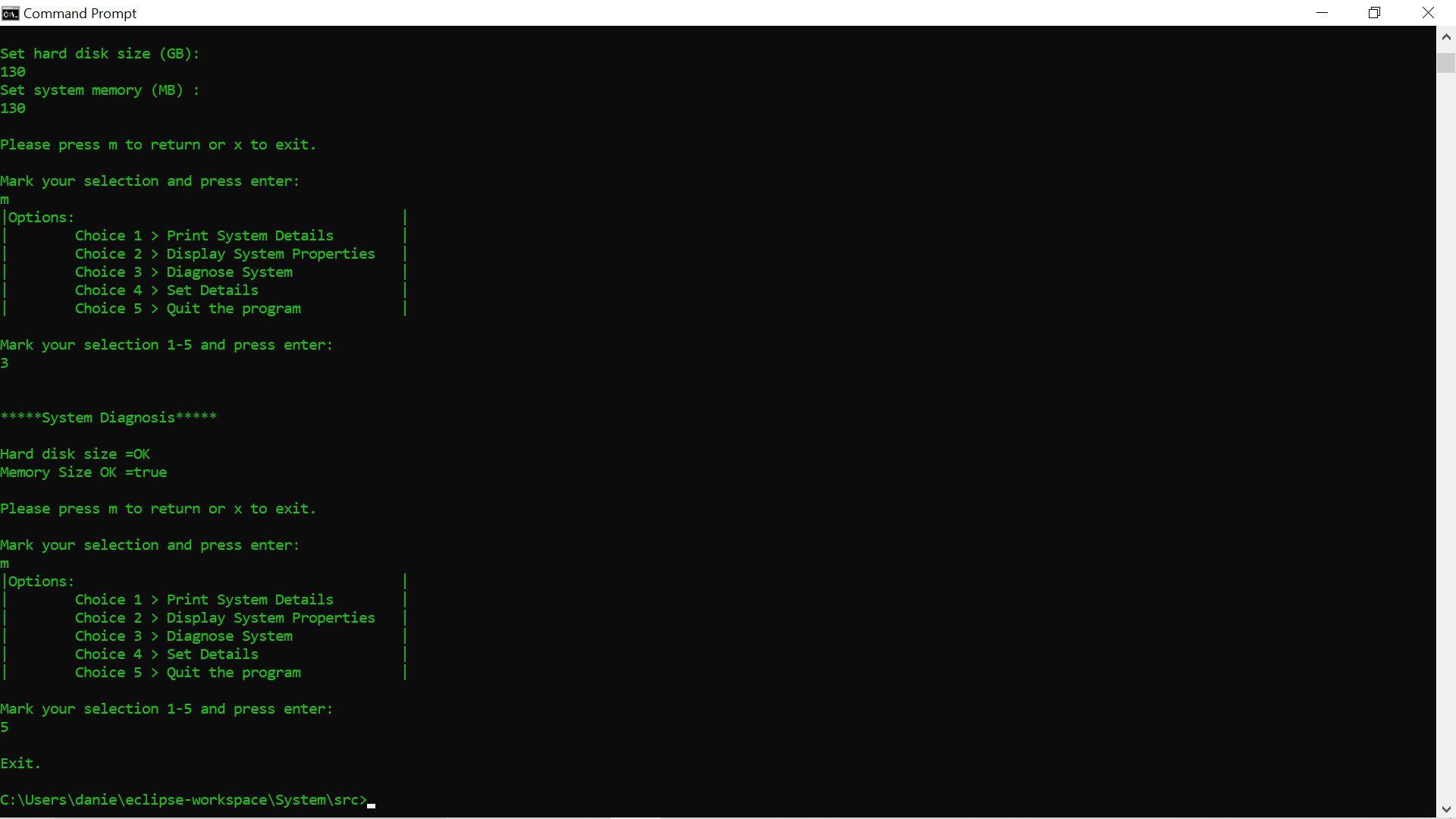
3)



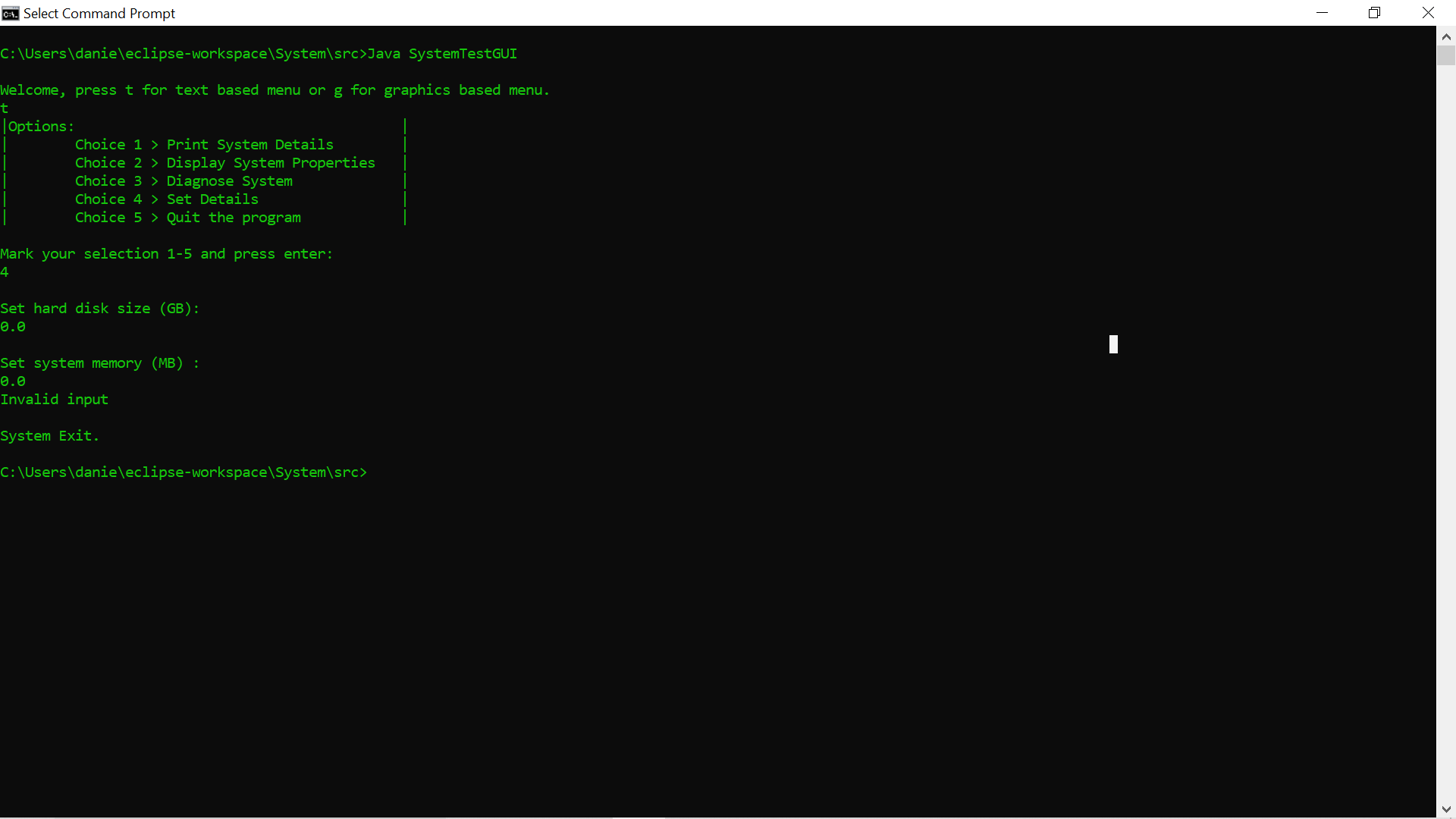
4)



5)



6)



SystemTestGUI Source Code:

//Test Class

//includes Part D of Algorithms and Data Structures Assessment 2

//includes main method to test all methods for assessment.

//imports

**import** java.util.Scanner;

**import** javax.swing.ImageIcon;

**import** javax.swing.JFrame;

**import** javax.swing.JLabel;

**import** javax.swing.JMenu;

**import** javax.swing.JMenuBar;

**import** javax.swing.JMenuItem;

**import** javax.swing.JOptionPane;

//attributes

**public** **class** SystemTestGUI {

**public** **static** JMenuItem *item1* = **new** JMenuItem("Choice 1 – Print System Details");

**public** **static** JMenuItem *item2* = **new** JMenuItem("Choice 2 - Display System Properties");

**public** **static** JMenuItem *item3* = **new** JMenuItem("Choice 3 – Diagnose System");

**public** **static** JMenuItem *item4* = **new** JMenuItem("Choice 4 – Set Details");

**public** **static** JMenuItem *item5* = **new** JMenuItem("Choice 5 – Quit the program");

**private** **static** String *make*;

**private** **static** String *model*;

**public** **static** String *status*;

**private** **static** String *checkHDStatus1*;

**private** **static** String *answer1*;

**private** **static** String *answer2*;

**private** **static** **int** *speed*;

**private** **static** **int** *MemorySize1*;

**private** **static** **double** *purchaseCost* = 200;

**public** **static** **boolean** *goodMemorySize1*;

**public** **static** **boolean** *valid* = **false**;

**public** **static** **int** *memorySize* = 0;

**public** **static** **int** *hardDisk* = 0;

**public** **enum** OSType { ***Windows***, ***Linux***, ***Other***};

**protected** **static** OSType *detectedOS*;

**public** **static** Scanner *keyboard* = **new** Scanner(java.lang.System.***in***);

// the constructor

**public** SystemTestGUI(String model, String make, **int** speed) {

**this**.*make* = make;

**this**.*model* = model;

**this**.*speed* = speed;

}

// methods to return attribute values

**public** String getMake()

{

**return** *make*;

}

**public** String getModel()

{

**return** *model*;

}

**public** **int** getSpeed() {

**return** *speed*;

}

// methods to set the corresponding attributes

**public** **static** **int** setHardDisk1() {

{

**do** {

**try** {

*hardDisk* = Integer.*parseInt*(JOptionPane.*showInputDialog*("Enter Hard Disk Size (GB):"));

**if** (**true**)

**break**;

} **catch** (Exception e) {

JOptionPane.*showMessageDialog*(**null**, "Invalid input" + "\nSystem Exit");

java.lang.System.*exit*(0);

}

} **while** (**false**);

}

**return** *hardDisk*;

}

**public** **static** **int** setMemory1() {

{

**do** {

**try** {

*memorySize* = Integer.*parseInt*(JOptionPane.*showInputDialog*("Enter Memory Size (MB):"));

**if** (**true**)

**break**;

} **catch** (Exception e) {

JOptionPane.*showMessageDialog*(**null**, "Invalid input" + "\nSystem Exit");

java.lang.System.*exit*(0);

}

} **while** (**false**);

}

**return** *memorySize*;

}

//Check if the memory size is below 128MB and print corresponding messages

**public** **static** **boolean** goodMemorySize1()

{

**if** (*memorySize* > 128)

*goodMemorySize1* = **true**;

**else** **if** (*MemorySize1* < 128)

*goodMemorySize1* = **false**;

**return** *goodMemorySize1*;

}

//check if the hard disk size is below 2GB and print corresponding messages

**public** **static** String checkHDStatus1() {

System System = **new** System("Ezpad 6 pro", "Shenzhen Jumper Technology Co., Ltd.", 1600000000);

**if** (*hardDisk* < 2)

**return** *status* =("low");

**else** **if** (*hardDisk* > 2);

**return** *status* = ("OK");

}

// Part D of Algorithms and Data Structures Assessment 2

// method to display graphical menu based on structure of textual menu.

// upgrades Part A to gui.

**public** **static** **void** GraphicalMenu() {

JFrame J = **new** JFrame("Menu Options");

JMenuBar menubar = **new** JMenuBar();

JMenu Menu1 = **new** JMenu ();

// An image was added in order to make the graphical menu look better.

// Items added to MenuBar

menubar.add(*item1*);

menubar.add(*item2*);

menubar.add(*item3*);

menubar.add(*item4*);

menubar.add(*item5*);

menubar.add(Menu1);

// Size set

J.setJMenuBar(menubar);

J.setSize(1500, 70);

J.setVisible(**true**);

// ActionListeners used for jbuttons.

// When activated showMessageDialog displays attributes.

*item1*.addActionListener(e -> {

{

SystemTestGUI System = **new** SystemTestGUI("Ezpad 6 pro", "Shenzhen Jumper Technology Co., Ltd.", 1600000000);

JOptionPane.*showMessageDialog*( **null**, "Computer Details:\n\n" + "System Make =" + SystemTestGUI.*make* + "\nSystem Model =" + SystemTestGUI.*model* + "\nSystem Speed =" + SystemTestGUI.*speed* + " hz" + "\nSystem Purchase Cost=" + "£" + SystemTestGUI.*purchaseCost* + "\n\n");

}

});

*item2*.addActionListener(e -> {

JOptionPane.*showMessageDialog*( **null**, "System Properties:\n" + "\nOs name =" + java.lang.System.*getProperty*("os.name") + "\nOs version = " + java.lang.System.*getProperty*("os.version") + "\nOs architecture = " + java.lang.System.*getProperty*("os.arch") + "\nUser name = " + java.lang.System.*getProperty*("user.name") + "\nJava version = " + java.lang.System.*getProperty*("java.version") + "\n\n");

// if else and else if statements used to display positive message if detectedOS = Windows 10, negative message for Linux and neutral for any other OS.

**if** (*detectedOS* == **null**) {

String OS = java.lang.System.*getProperty*("os.name", "generic");

**if** ((OS.indexOf("Windows 10") >= 0)) {

*detectedOS* = OSType.***Windows***;

JOptionPane.*showMessageDialog*( **null**, "Thumbs up! Your operating system is Windows 10");

}

**else** **if** (OS.indexOf("nux") >= 0) {

*detectedOS* = OSType.***Linux***;

JOptionPane.*showMessageDialog*( **null**, "Thumbs down! Your operating system is Linux");

}

**else** {

*detectedOS* = OSType.***Other***;

JOptionPane.*showMessageDialog*( **null**, "Your choice of operating system is ok");

}

}

});

*item3*.addActionListener(e -> {

JOptionPane.*showMessageDialog*(**null**, "System Diagnosis\n" + "\nHard disk size =" + *checkHDStatus1*() + "\nMemory Size OK =" + *goodMemorySize1*() + "\n\n");

});

*item4*.addActionListener(e -> {

*setHardDisk1*();

*setMemory1*();

});

*item5*.addActionListener(e -> {

java.lang.System.*exit*(0);

});

}

// Methods are called into main.

// User is prompted to choose between graphical or text-based menu to start.

// The program can be run from the command prompt using Java SystemTestGUI.

// The former is meant to simplify ease of use.

**public** **static** **void** main(String[] args) {

java.lang.System.***out***.println("");

java.lang.System.***out***.println("Welcome, press t for text based menu or g for graphics based menu.");

*answer2* = *keyboard*.next();

**if** (*answer2*.equals("t"))

{System.*TextualMenu*(); }

**if** (*answer2*.equals("g"))

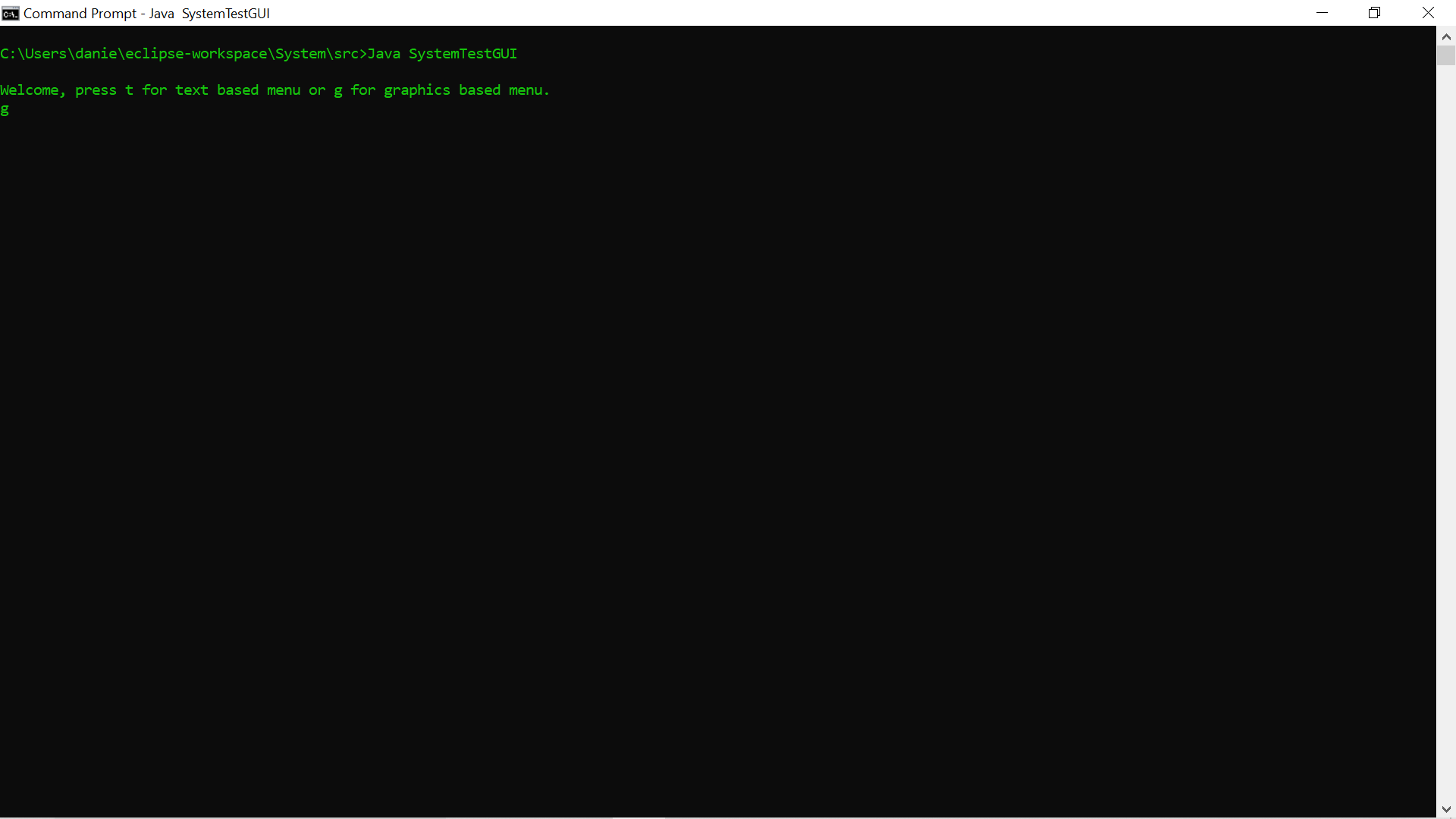
{*GraphicalMenu*(); }

}

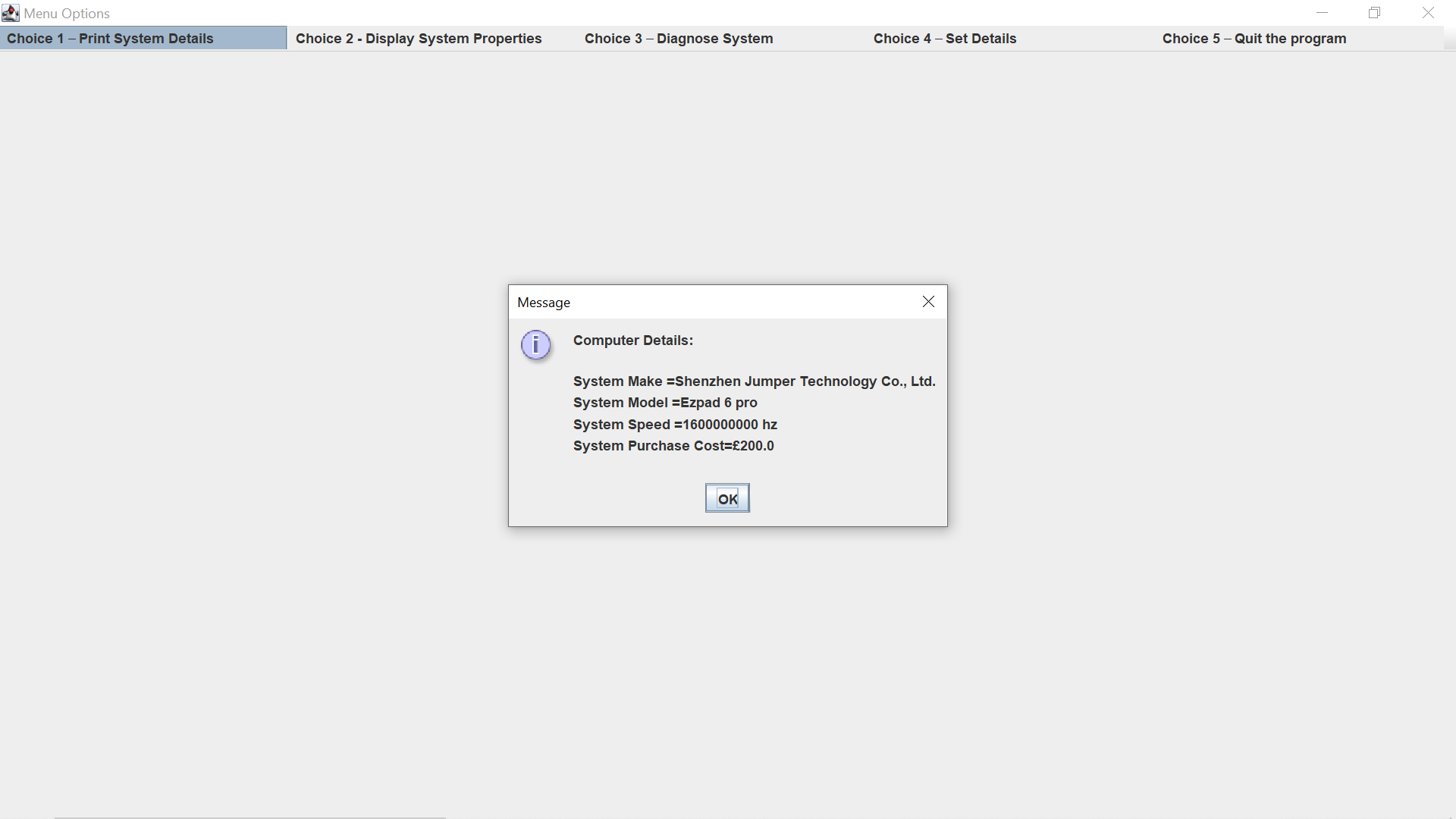
}

SystemTestGUI Class List of Outputs:

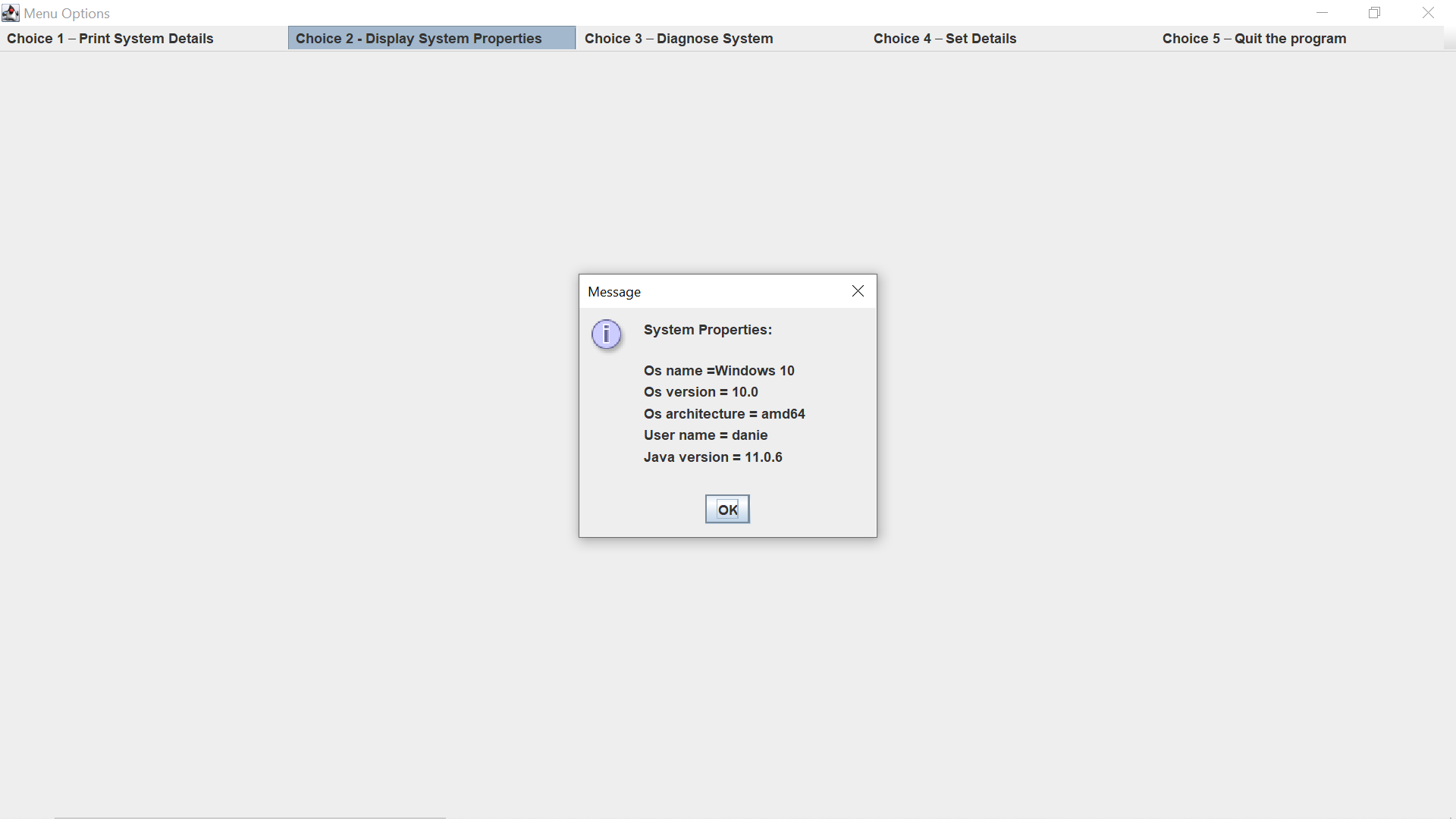
1)



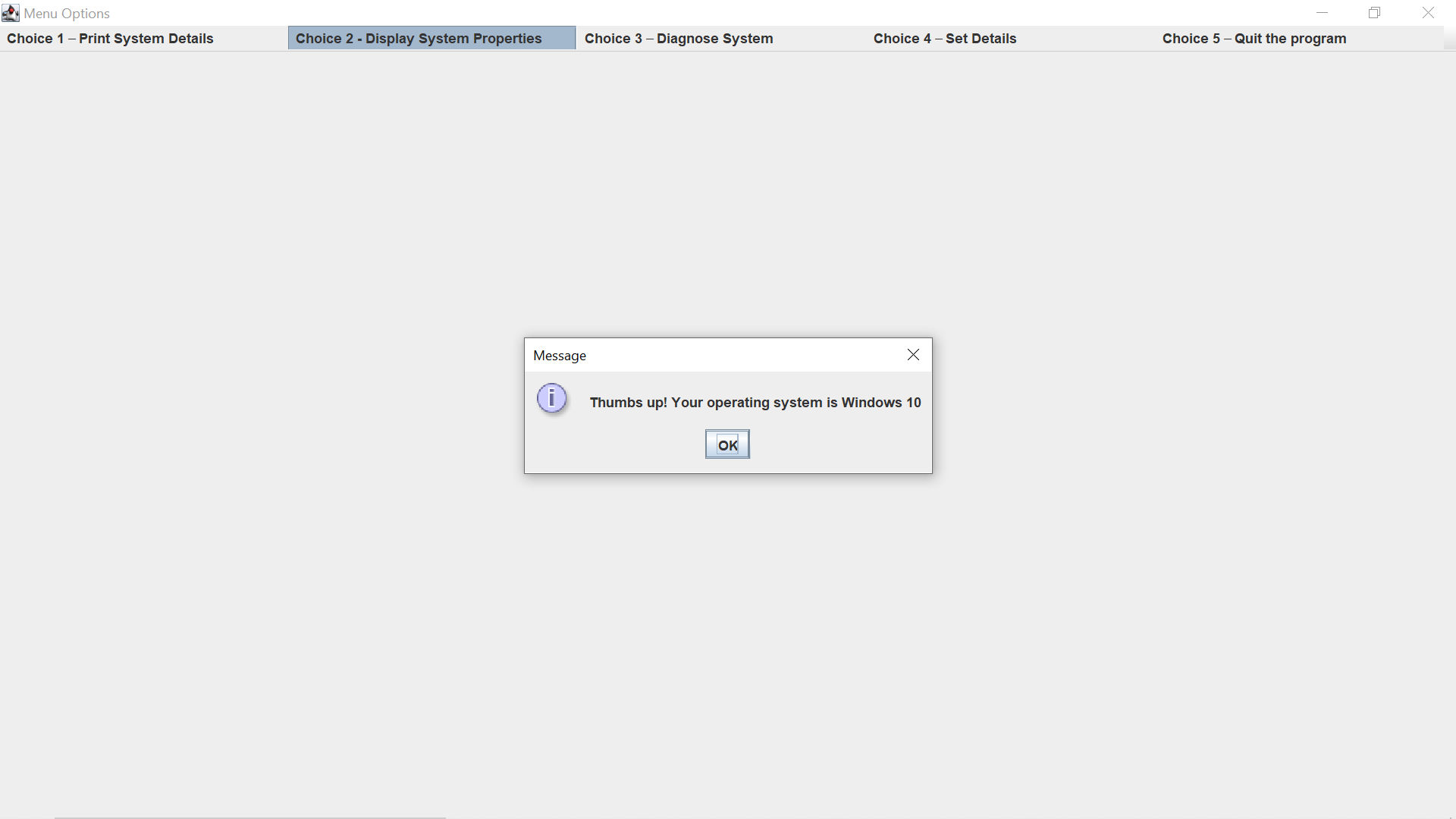
2)



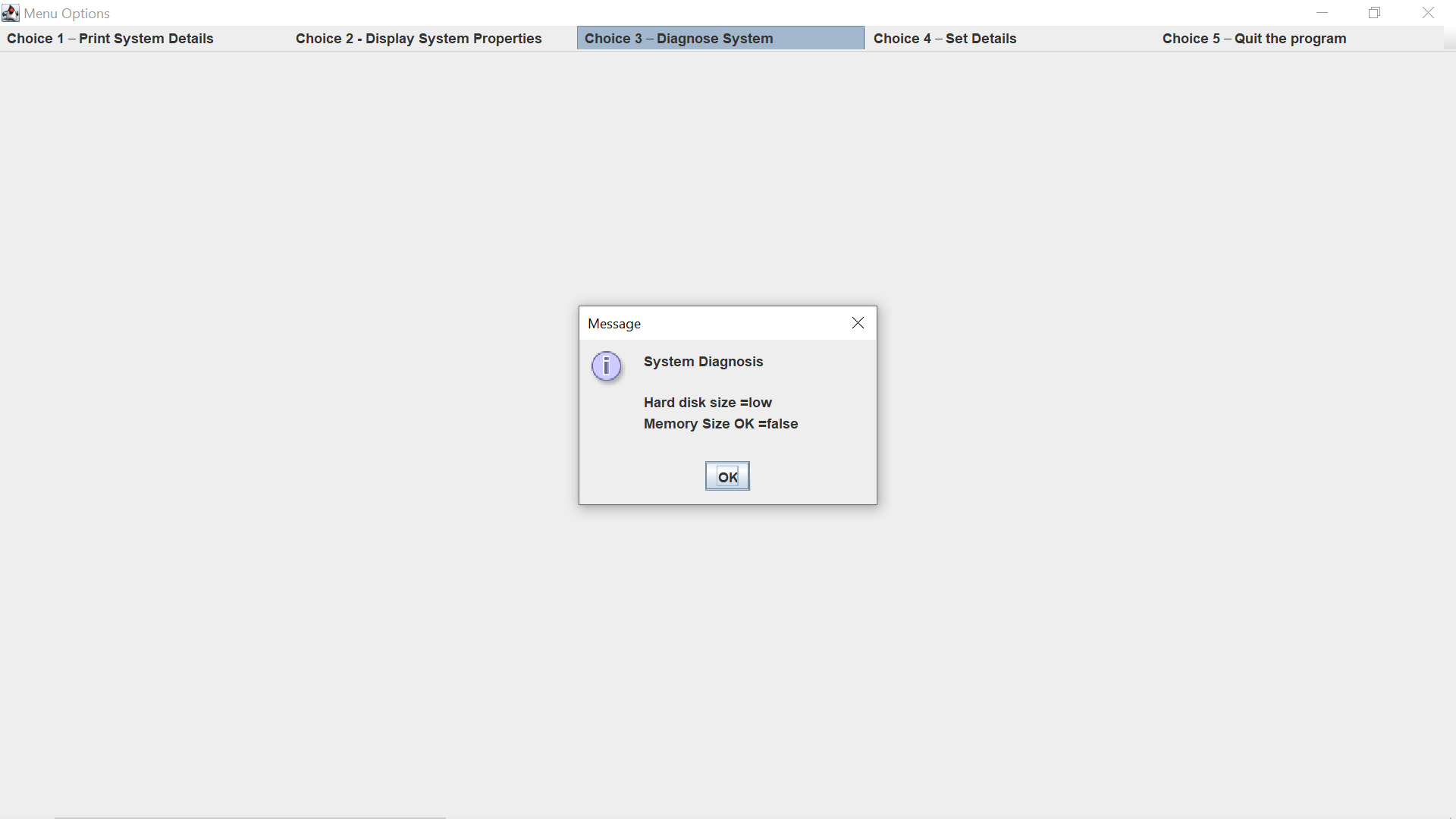
3)



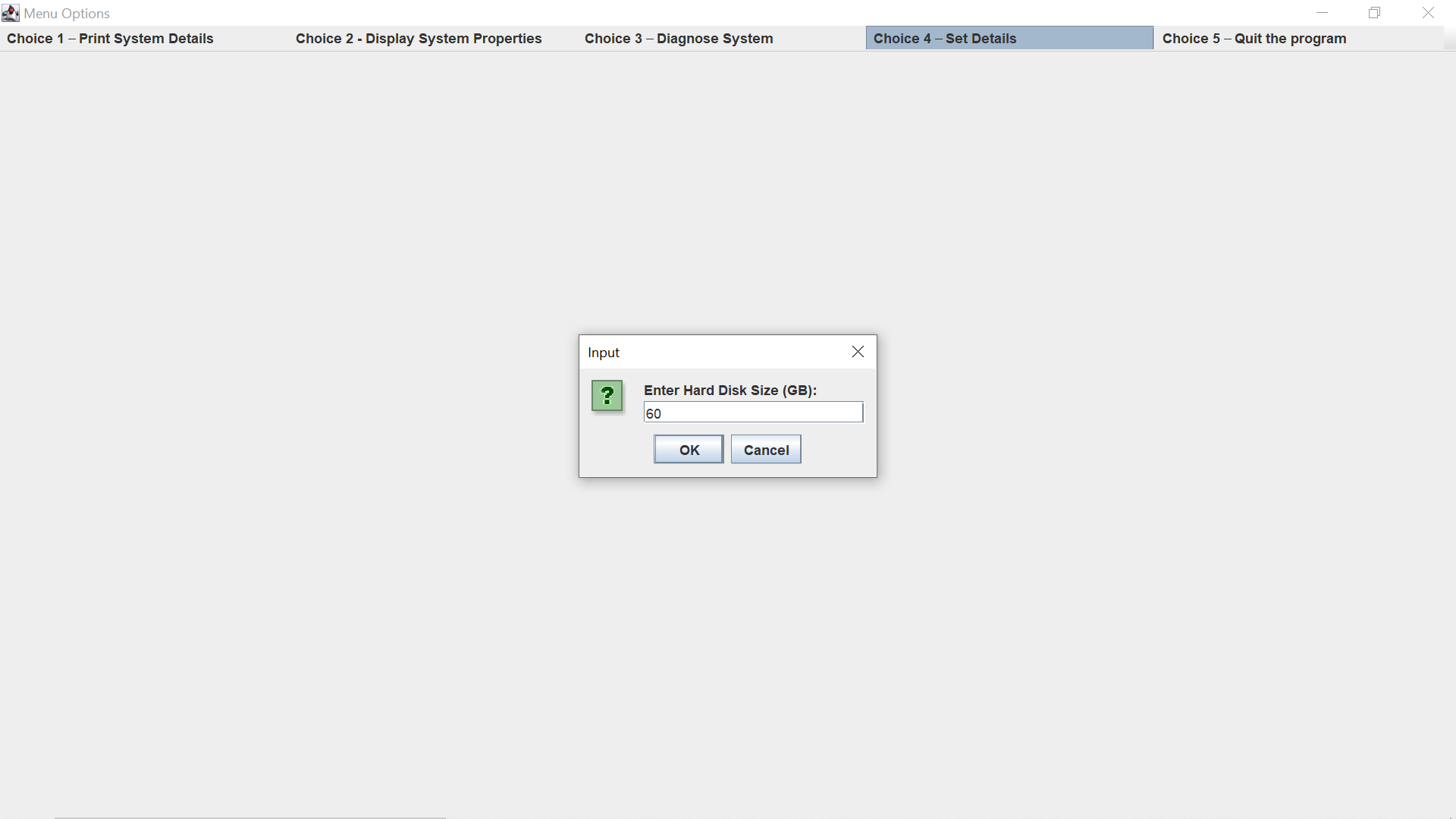
4)



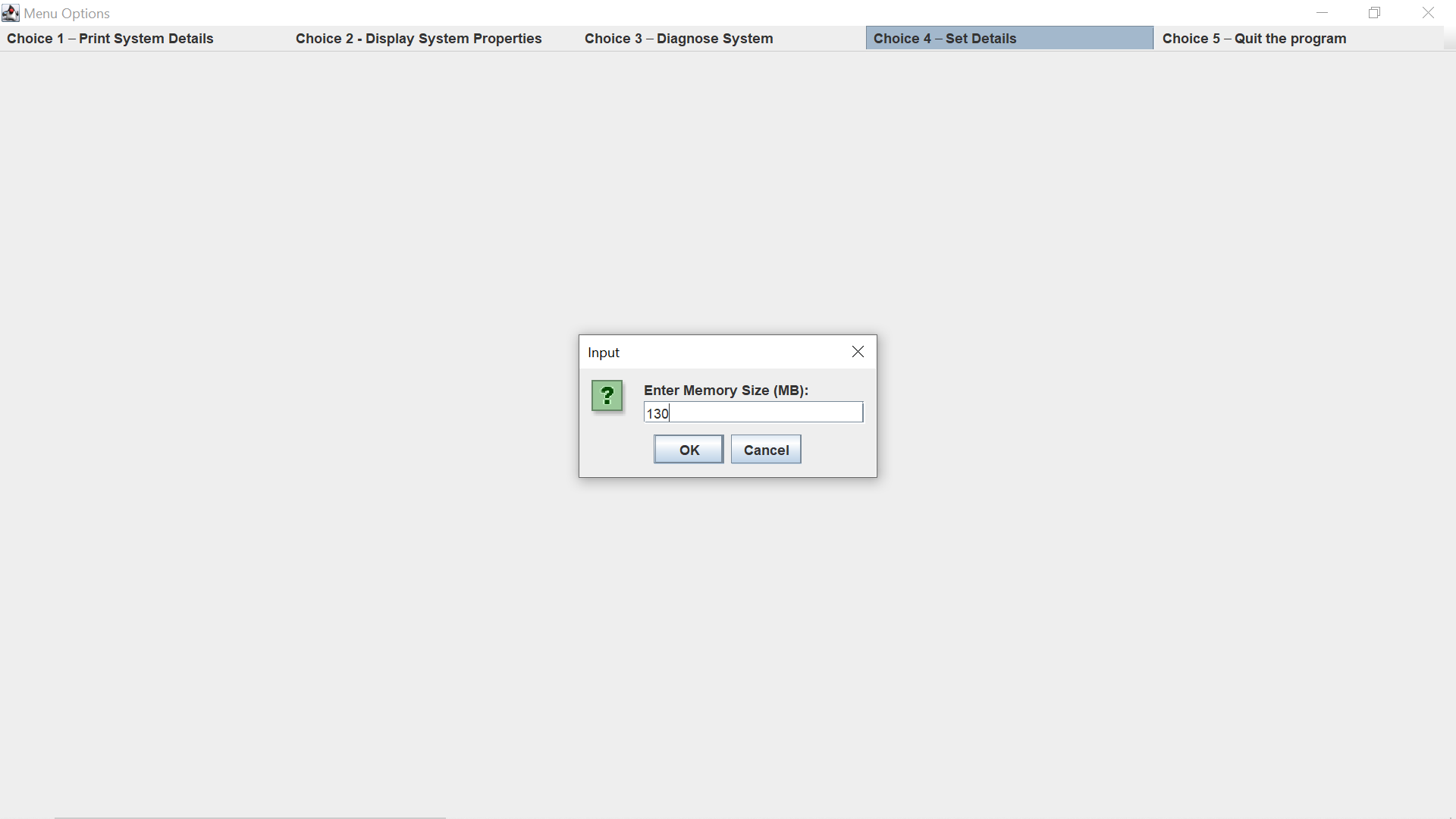
5)



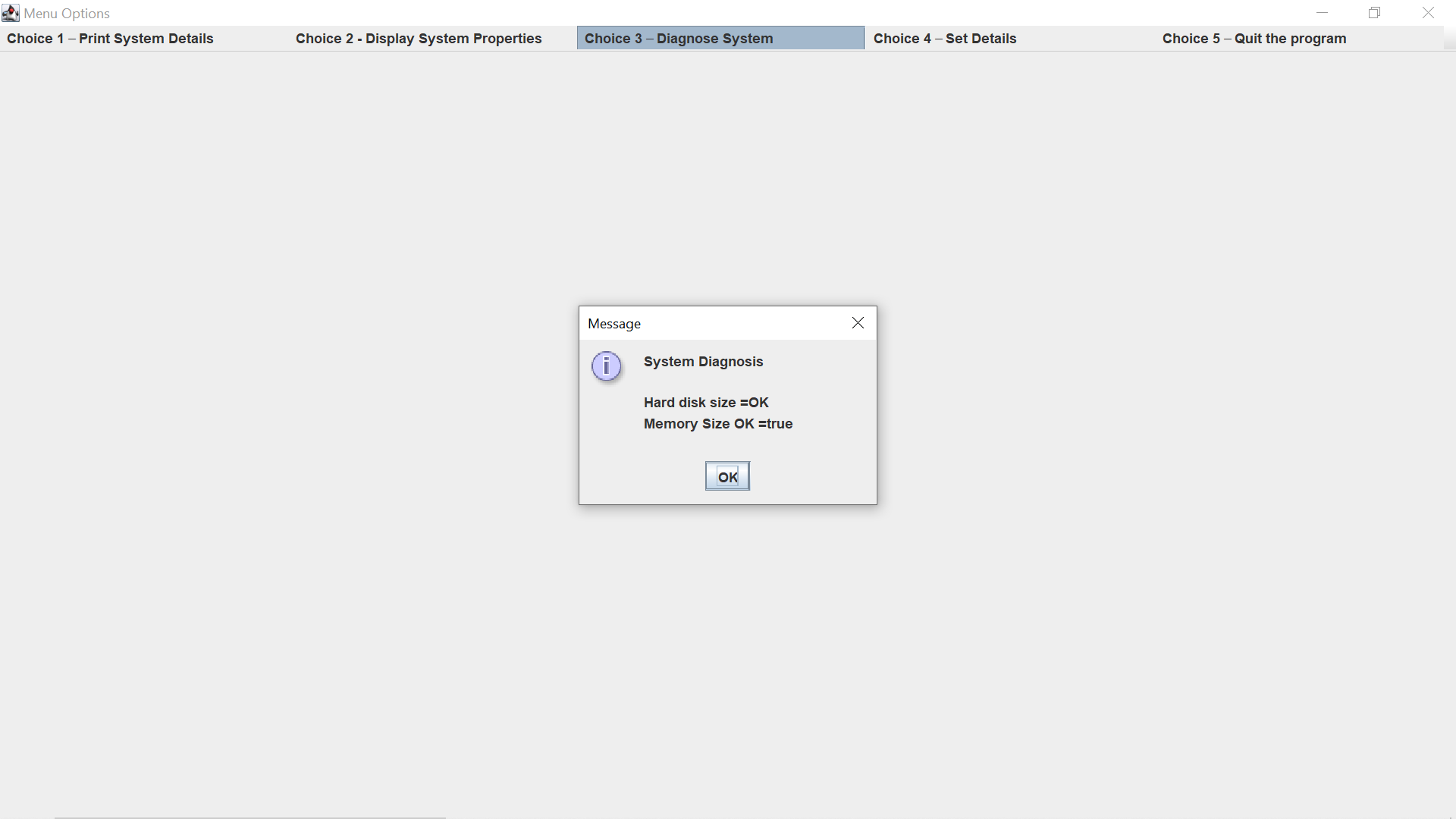
6)



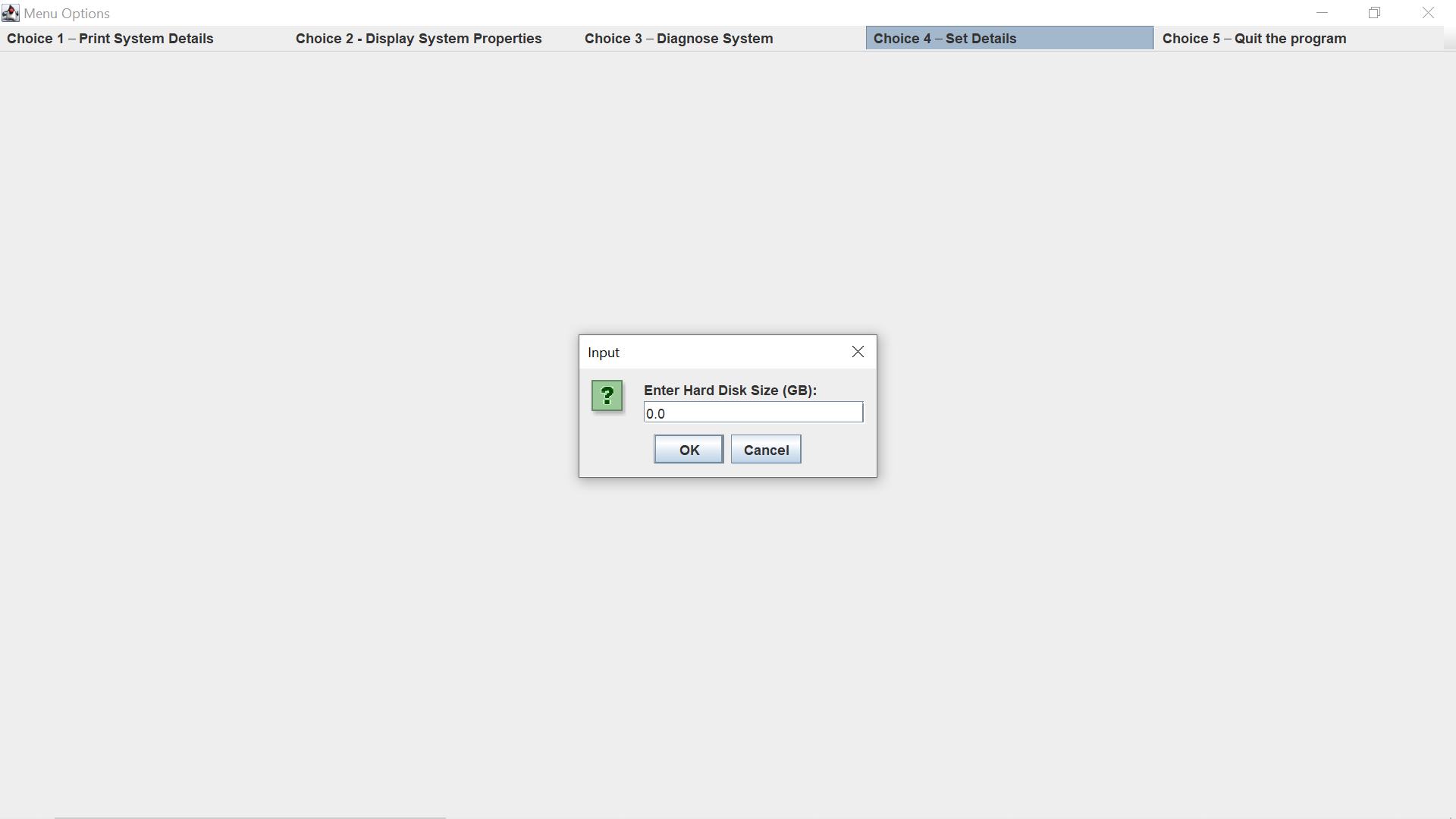
7)



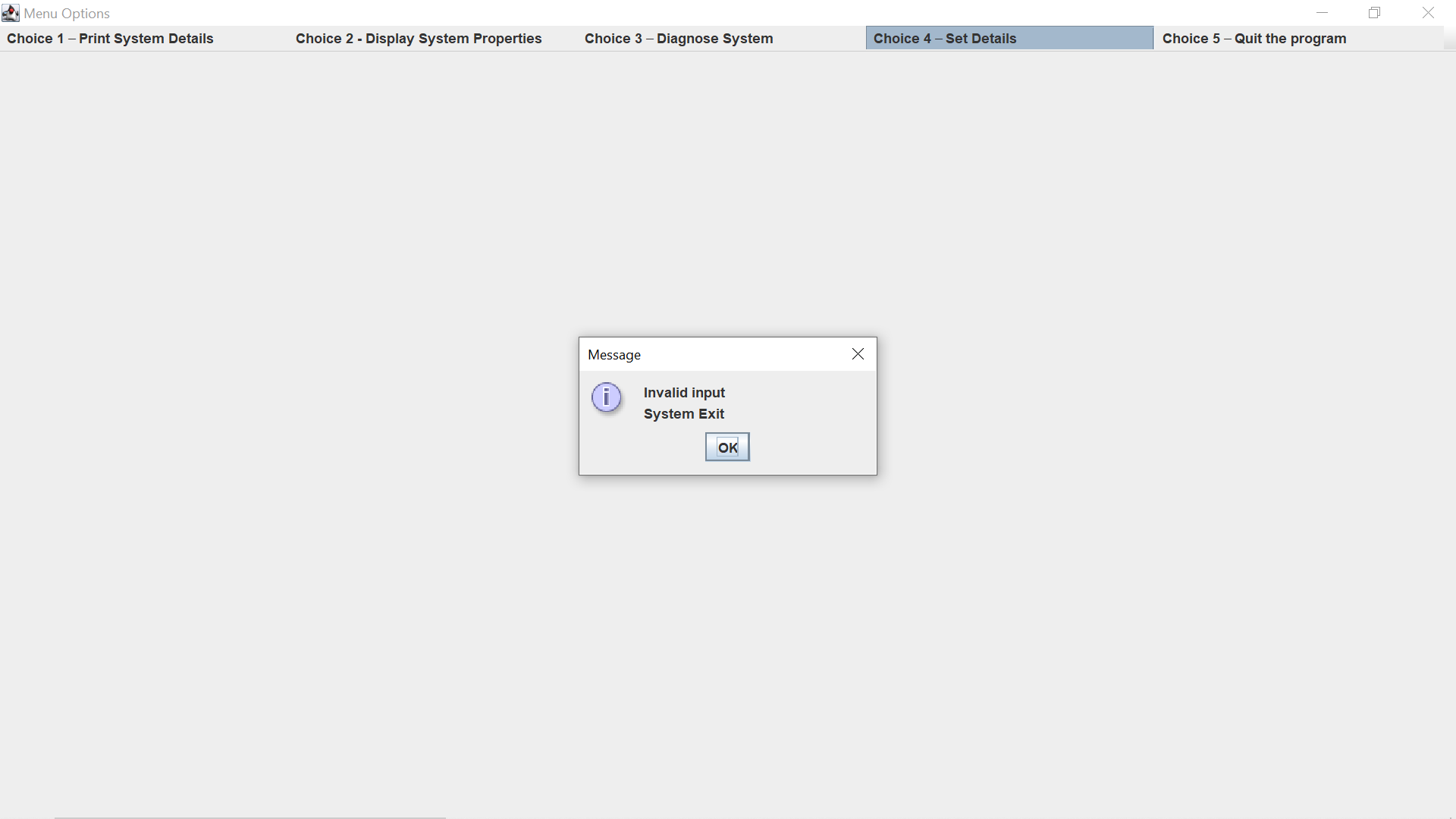
8)



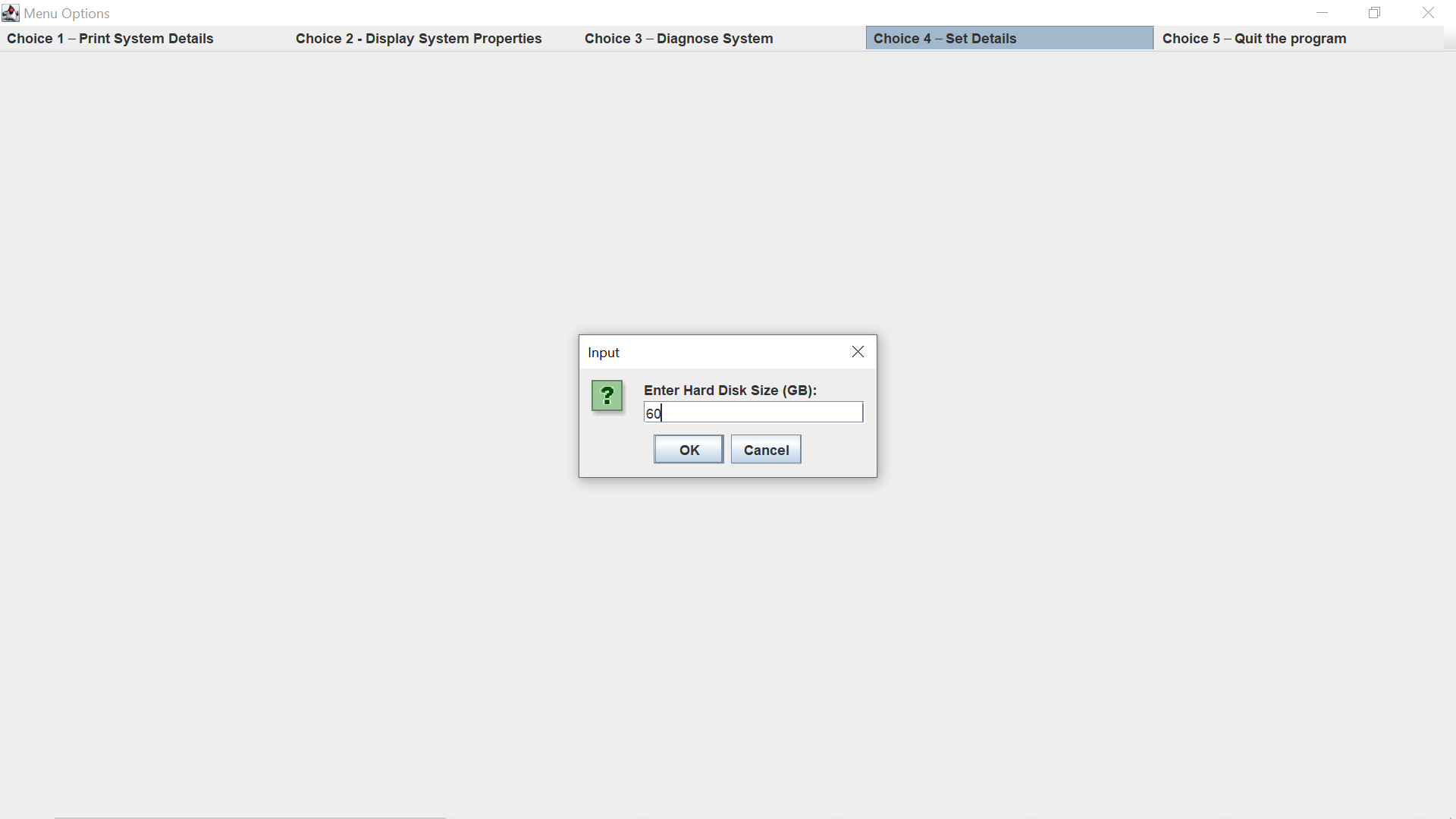
9)



10)



11)



12)

