Question 1

1a)

import javax.swing.\*;

import java.util.Objects;

public class DialogBox {

public String inputCourseCode() {

String courseCode;

courseCode = JOptionPane.showInputDialog(null,

"Input the course code of this course:");

if (JOptionPane.showConfirmDialog(null,

"The course code input is: " + courseCode + ", is it correct?", null,

JOptionPane.YES\_NO\_CANCEL\_OPTION) != JOptionPane.YES\_OPTION) {

courseCode = null;

}

return courseCode;

}

public void checkCourseCode(){

String courseCode = null;

final String correctCourseCode = "COMPS203F";

do {

try {

courseCode = inputCourseCode();

if (!Objects.equals(courseCode, correctCourseCode))

throw new IllegalArgumentException();

} catch (IllegalArgumentException e) {

JOptionPane.showMessageDialog(null,

"The course code \"" + courseCode + "\" is not correct");

}

} while (!Objects.equals(courseCode, correctCourseCode));

}

public static void main(String[] args) {

DialogBox dialogBox = new DialogBox();

dialogBox.checkCourseCode();

}

}

1b)

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

public class LengthConversion {

public LengthConversion(){

JFrame frame = new JFrame("");

frame.setLayout(new GridLayout(2,2));

JTextField textField = new JTextField();

JLabel label = new JLabel();

JButton btnToInches = new JButton("cm to inches");

btnToInches.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try {

label.setText(Double.toString((double) Math.round(Double.parseDouble(textField.getText()) / 2.54 \* 100) / 100) + " inches");

}

catch (NumberFormatException ignored){}

}

});

JButton btnToCm = new JButton("inches to cm");

btnToCm.addActionListener(new ActionListener() {

@Override

public void actionPerformed(ActionEvent e) {

try{

label.setText(Double.toString((double)Math.round(Double.parseDouble(textField.getText())\*2.54\*100)/100)+" cm");

}

catch (NumberFormatException ignored){}

}

});

frame.add(textField);

frame.add(label);

frame.add(btnToInches);

frame.add(btnToCm);

frame.setDefaultCloseOperation(JFrame.EXIT\_ON\_CLOSE);

frame.setLocationRelativeTo(null);

frame.pack();

frame.setVisible(true);

}

public static void main(String[] args) {

LengthConversion lengthConversion = new LengthConversion();

}

}

Question 2 (Complete view of class HexEditor is at the last two pages)

a)

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import [java.io](http://java.io).\*;

public class HexEditor extends JFrame implements ActionListener{

public HexEditor() {

JTextArea textArea = new JTextArea(null,"", 5, 10);

add(textArea);

JMenuBar menuBar = new JMenuBar();

JMenu menu = new JMenu("File");

JMenuItem menuItem = new JMenuItem(“Load");

menuItem.addActionListener(this);

menu.add(menuItem);

menuBar.add(menu);

setJMenuBar(menuBar);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

pack();

setVisible(true);

}

}

b)

public class TestHexEditor {

public static void main(String[] args) {

HexEditor anEditor = new HexEditor();

anEditor.setVisible(true);

}

}

c)

private String activeFile;

private JTextArea textArea;

public HexEditor() {

textArea = new JTextArea(null,"", 5, 10);

}

public void actionPerformed(ActionEvent e){

if (e.getSource().equals(menuItem)) {

try {

textArea.setText("");

activeFile = JOptionPane.showInputDialog(null, "Input file name", "/Users/herbert/asm.txt");

FileInputStream fin = new FileInputStream(activeFile);

while (fin.available() > 0)

textArea.append("" + (char) fin.read());

} catch (IOException | NullPointerException ex) {

JOptionPane.showMessageDialog(null,"No such file");

activeFile = null;

}

}

}

d)

textArea = new JTextArea(null,"", 5, 10);

add(textArea,BorderLayout.WEST);

JLabel gap = new JLabel();

gap.setPreferredSize(new Dimension(5,gap.getHeight()));

gap.setBackground(Color.gray);

add(gap);

JTextArea inputArea = new JTextArea();

inputArea.setColumns(20);

add(inputArea, BorderLayout.EAST);

JButton btn = new JButton("Update hex”);

btn.addActionListener(this);

add(btn, BorderLayout.SOUTH);

e)

private JMenuItem menuItem;

private JButton btn;

private JTextArea hexArea;

public HexEditor() {

menuItem = new JMenuItem(“Load");

hexArea = new JTextArea();

btn = new JButton("Update hex”);

}

public void actionPerformed(ActionEvent e){

if (e.getSource().equals(btn)){

hexArea.setText("");

try {

FileInputStream fin = new FileInputStream(activeFile);

while (fin.available() > 0)

hexArea.append(Integer.toHexString(fin.read())+" ");

} catch (IOException | NullPointerException ignored) {

}

}

}

f)

private JMenuItem save;

public HexEditor() {

save = new JMenuItem("Save");

save.addActionListener(this);

menu.add(save);

}

public void actionPerformed(ActionEvent e){

if(e.getSource().equals(save)){//

File file = null;

FileWriter out=null;

try {

if(activeFile!=null){

file = new File(activeFile);

out = new FileWriter(file);

} else{

String newFile = JOptionPane.showInputDialog(null, "Enter NEW file name", "\"/Users/herbert/asm.txt\"");

file = new File(newFile);

out = new FileWriter(file);

}

out.write(textArea.getText());

out.close();

}

catch (IOException | NullPointerException ignored){}

}

}  
  
Question 3

a)

public static void recursiveReversePrint(String aString) {

if ((null == aString) || (aString.length() <= 1)) {

System.out.print(aString);

}

else {

recursiveReversePrint(aString.substring(1));

System.out.print(aString.charAt(0));

}

}

b)

i)

public class Staff {

private String staffID, name;

private int salary;

public Staff(String anID, String aName, int aSalary){

staffID = anID;

name = aName;

salary = aSalary;

}

public String getStaffID() {

return staffID;

}

public String getName() {

return name;

}

public int getSalary() {

return salary;

}

@Override

public String toString() {

return "StaffID: " + staffID + ", Name: " + name + ", Salary: " + (double) salary;

}

}

ii)

import java.util.HashMap;

public class Company {

private HashMap<String, Staff> staffMap = new HashMap<String, Staff>();

public void addStaff(Staff oneStaff){

staffMap.put(oneStaff.getStaffID(), oneStaff);

}

}

iii)

public class TestCompany {

public static void main(String[] args) {

Company smallCompany = new Company();

smallCompany.addStaff(new Staff("A007", "James", 10001));

smallCompany.addStaff(new Staff("A009", "James", 999));

smallCompany.addStaff(new Staff("A012", "Moon", 5000));

}

}

iv)

In class TestCompany:

public static void main(String[] args) {

smallCompany.showStaff();

}

In class Company:

public void showStaff(){

for(Object key:staffMap.keySet()){

System.out.println(staffMap.get(key));

}

}

v)

In class TestCompany:

System.out.println(smallCompany.nameSet());

In class Company:

import java.util.Set;

public class Company {

public Set<String> nameSet(){

Set<String> names = new HashSet<>();

for(Object key:staffMap.keySet()){

names.add(staffMap.get(key).getName());

}

return names;

}

}

vi)

In class TestCompany:

System.out.println(smallCompany.salaryList());

In class Company:

import java.util.List;

public class Company {

public List<Integer> salaryList(){

List<Integer> salaries = new ArrayList<>();

for(Object key:staffMap.keySet()){

salaries.add(staffMap.get(key).getSalary());

}

return salaries;

}

}

vii)

In main of class TestCompany:

smallCompany.removeStaff(“A009");

smallCompany.showStaff();

In class Company:

public Staff removeStaff(String staffID){

Staff staff;

if ((staff = staffMap.get(staffID)) == null){

return null;

}

staffMap.remove(staffID);

return staff;

}

Complete view of class HexEditor

import javax.swing.\*;

import java.awt.\*;

import java.awt.event.ActionEvent;

import java.awt.event.ActionListener;

import java.io.\*;

public class HexEditor extends JFrame implements ActionListener{

private JTextArea textArea;

private JMenuItem menuItem;

private JButton btn;

private JTextArea hexArea;

private String activeFile;

private JMenuItem save;

public HexEditor() {

textArea = new JTextArea(null,"", 5, 10);

add(textArea,BorderLayout.WEST);

JMenuBar menuBar = new JMenuBar();

JMenu menu = new JMenu("File");

menuItem = new JMenuItem("Load");

menuItem.addActionListener(this);

menu.add(menuItem);

menuBar.add(menu);

setJMenuBar(menuBar);

JLabel gap = new JLabel();

gap.setPreferredSize(new Dimension(5,gap.getHeight()));

gap.setBackground(Color.gray);

add(gap);

hexArea = new JTextArea();

hexArea.setColumns(20);

add(hexArea, BorderLayout.EAST);

btn = new JButton("Update hex");

btn.addActionListener(this);

add(btn, BorderLayout.SOUTH);

save = new JMenuItem("Save");

save.addActionListener(this);

menu.add(save);

setDefaultCloseOperation(EXIT\_ON\_CLOSE);

setLocationRelativeTo(null);

pack();

setVisible(true);

}

public void actionPerformed(ActionEvent e){

if (e.getSource().equals(menuItem)) {

try {

textArea.setText("");

activeFile = JOptionPane.showInputDialog(null, "Input file name", "/Users/herbert/asm.txt");

FileInputStream fin = new FileInputStream(activeFile);

while (fin.available() > 0)

textArea.append("" + (char) fin.read());

} catch (IOException | NullPointerException ex) {

JOptionPane.showMessageDialog(null,"No such file");

activeFile = null;

}

}

if (e.getSource().equals(btn)){

hexArea.setText("");

try {

FileInputStream fin = new FileInputStream(activeFile);

while (fin.available() > 0)

hexArea.append(Integer.toHexString(fin.read())+" ");

} catch (IOException | NullPointerException ignored) {

}

}

if(e.getSource().equals(save)){//

File file = null;

FileWriter out=null;

try {

if(activeFile!=null){

file = new File(activeFile);

out = new FileWriter(file);

} else{

String newFile = JOptionPane.showInputDialog(null, "Enter NEW file name", "\"/Users/herbert/asm.txt\"");

file = new File(newFile);

out = new FileWriter(file);

}

out.write(textArea.getText());

out.close();

}

catch (IOException | NullPointerException ignored){}

}

}

public static void main(String[] args) {

HexEditor hexEditor = new HexEditor();

}

}