PIC 20A Principles of Java Language with Applications

Final
Instructor: Shu Liu
Quarter: 2023 Fall

Instructions:

- You have **180 minutes** to complete the exam.
- You may **not** use any books or notes.
- Write your solutions in the space **below**, **next to** or **included with** the questions.
- If you need more space, use **scratch paper**.
- If you write important work on the scratch paper, **indicate** that you have done so **next to the relevant question**.
- Do not forget to write your name, UID, and discussion in the space below.
- **Good luck!**

Name: ______________________________
Student ID number: ________________________
Discussion: ____________________________

Question	Points	Score
1	17	
2	14	
3	19	
4	10	
5	10	
6	10	
Total:	80	

Problem 1. *17pts.*

Read the following code (saved in a file called `Question1.java`), and answer the following questions (a)-(e).

```
class Question1 {
    public static void main(String[] args){
        byte minByte = -128;
        byte maxByte = 127;
        short minShort = -32768;
        short maxShort = 32767;
        int minInt = -2147483648;
        int maxInt = 2147483647;

        short s1 = 512;
        int i = s1;
        // System.out.println(s1 == ((byte) i));


        // short s2 = maxByte + maxByte;
        // System.out.println(s2);


        double[] arr1 = new double[] {1.5, 2.5, 3.5, 4.5};
        // double[] arr2 = arr1;
        // for (int k = 0; k < arr2.length; k++){arr2[k] = arr2[k] * 2;}
        // for (int i = 0; i<arr1.length; i++) {System.out.print(arr1[k]+" ");}


        // int[] arr3 = new int[3];
        // f(arr3);
        // System.out.print(arr3[0] + " " + arr3[arr3.length - 1]);


        // int num = 24;
        // g(num);


        int[] arr4 = new int[] {1, 4, 9, 16, 25};
        // arr4 = h(arr4);
        // arr4 = h(arr4);
        // System.out.print(java.util.Arrays.toString(arr4));

    }


    public static void f(int[] array){
        array[0] = 888;
        array = new int[5];
    }

    public static void g(int num){
        int d = 2;
        while(num > 1){
            while(num % d == 0){
                System.out.print(d+" ");
                num = num / d;
            }
            d = d + 1;
        }
    }

    public static int[] h(int[] array){
        int[] newarray = new int[array.length - 1];
        for(int k = 0; k < array.length - 1; k++){
            newarray[k] = array[k+1] - array[k];
        }
        return newarray;
    }
}
```

For the lines in the `main` method that are commented, consider what happens when you uncomment them while **leaving the other lines commented**.

- **EITHER explain** why uncommenting the line gives a **compile-time error**;
- **OR write down the output** the line produces upon uncommenting it.

(a) (3 pts) `// System.out.println(s1 == ((byte) i));`

(b) (2 pts)

```
// short s2 = maxByte + maxByte;
// System.out.println(s2);
```

(c) (3 pts)

```
// double[] arr2 = arr1;
// for (int k = 0; k < arr2.length; k++){arr2[k] = arr2[k] * 2;}
// for (int i = 0; i<arr1.length; i++) {System.out.print(arr1[k]+" ");}
```

(d) (3 pts)

```
// int[] arr3 = new int[3];
// f(arr3);
// System.out.print(arr3[0] + " " + arr3[arr3.length - 1]);
```

(e) (3 pts)

```
// int num = 24;
// g(num);
```

(f) (3 pts)

```
// arr4 = h(arr4);
// arr4 = h(arr4);
// System.out.print(java.util.Arrays.toString(arr4));
```

Hint: The `toString()` method of `Array` has the format: `[*, *, *,...]`, where `*` indicates the value of each entry of the array.

Problem 2. *14pts.*

The following code implements the `BankAccount` class.

```
BankAccount.java
```

```java
class BankAccount{
    public static final String bankName = "Java Bank";
    private static double bankMoney = 0;
    private String accountName;
    private double accountMoney;

    public BankAccount(String name, double initialDeposit){
        accountName = name; accountMoney = initialDeposit;
        bankMoney = bankMoney + initialDeposit;
    }
    public BankAccount(){this("Default Account", 0.0);}
    public BankAccount(String name){this(name, 0.0);}
    public static BankAccount createAccountViaName(BankAccount a){
        return new BankAccount(a.accountName);
    }

    public static void printBankMoney(){System.out.println(bankMoney);}

    private static void changeBankMoney(double change, boolean isDeposit){
        if(isDeposit){ bankMoney = bankMoney + change; }
        else{ bankMoney = bankMoney - change; }
    }

    public double getAccountMoney(){return accountMoney;}

    public void deposit(double depositMoney){
        accountMoney = accountMoney + depositMoney;
        changeBankMoney(depositMoney, true);
    }
    public void withdraw(double withdrawMoney){
        if(accountMoney >= withdrawMoney){
            accountMoney = accountMoney - withdrawMoney;
            changeBankMoney(withdrawMoney, false);
        }
        else{System.out.println("Insufficient funds.");}
    }

    public String toString(){
        return "Name: " + accountName + ", " + "Balance: " + accountMoney;
    }
}
```

The following code is saved in a file called `Question2.java`.

```
Question2.java
```

```
class Question2{
    public static void main(String[] args){

        // System.out.println(BankAccount.bankName);
        // BankAccount.printBankMoney();

        BankAccount account1 = new BankAccount();
        BankAccount account2 = new BankAccount("Michael");
        BankAccount account3 = BankAccount.createAccountViaName(account2);
        BankAccount account4 = new BankAccount("Lara", 100);

        // System.out.println(account1);
        // System.out.println(account2);
        // System.out.println(account3);
        // System.out.println(account4);
        // BankAccount.printBankMoney();

        // account4.withdraw(20);
        // account3.deposit(40);
        // System.out.println(account3);
        // System.out.println(account4);
        // account1.withdraw(50);
        // System.out.println(account1);
        // BankAccount.printBankMoney();

        // BankAccount.changeBankMoney(1000);
        // BankAccount.printBankMoney();
    }
}
```

Suppose we place both `BankAccount.java` and `Question2.java` in the same folder. Then `Question2.java` compiles and runs properly. For the lines in the `main` method that are commented, consider what happens when you uncomment them while **leaving the other lines commented**.

- **EITHER explain** why uncommenting the line gives a **compile-time error**;
- **OR write down the output** the line produces upon uncommenting it.

(a) (2 pts)

```
// System.out.println(BankAccount.bankName);
// BankAccount.printBankMoney();
```

(b) (5 pts)

```
// System.out.println(account1);
// System.out.println(account2);
// System.out.println(account3);
// System.out.println(account4);
// BankAccount.printBankMoney();
```

(c) (5 pts)

```
// account4.withdraw(20);
// account3.deposit(40);
// System.out.println(account3);
// System.out.println(account4);
// account1.withdraw(50);
// System.out.println(account1);
// BankAccount.printBankMoney();
```

(d) (2 pts)

```
// BankAccount.changeBankMoney(1000);
// BankAccount.printBankMoney();
```

Problem 3. *19pts.*

Suppose we have written some Java files and arrange them as follows

```
package1 --- Animal.java
          |_ Pet.java
          |_ Dog.java
          |_ Cat.java
          |_ Samoyed.java
          |_ PersianCat.java
          |_ Question3.java


package2 --- Husky.java
          |_ Shorthair.java
```

Where `package1` and `package2` are two separate packages.

The Java files in `package1` are listed below.

Animal.java

```java
package package1;

public interface Animal{
    public void run();
    public void sleep();
    public void sound();
}
```

Pet.java

```java
package package1;

public interface Pet{
    public String favoritePetToy();
    public void feature();
}
```

Dog.java

```java
package package1;

public abstract class Dog implements Animal, Pet{
    public String name;

    public Dog(String name){this.name = name;}

    public void run(){ System.out.println("A dog is chasing a frisbee."); }

    public void sleep(){
       System.out.println("Dogs sleep between 12 and 14 hours a day.");
    }

    public final void sound(){ System.out.println("Woof"); }

    public String favoritePetToy(){ return "frisbee"; }

    protected void nameOrigin(){ System.out.println("Unknown."); }

    void whereFrom(){ System.out.println("All over the world."); }
}
```

Cat.java

```java
package package1;

public abstract class Cat implements Animal, Pet{
    public String name;

    public Cat(String name){this.name = name;}

    public void run(){ System.out.println("A cat is chasing a mouse."); }

    public void sleep(){
       System.out.println("Cats sleep between 12 and 18 hours a day.");
    }

    public final void sound(){System.out.println("Meow");}

    public String favoritePetToy(){return "yarn ball";}

    protected void whereFrom(){System.out.println("All over the world.");}
}
```

Samoyed.java

```java
package package1;

public final class Samoyed extends Dog implements Pet{

    public int age;

    public Samoyed(int age, String name){ super(name); this.age = age; }

    public String toString(){
       return "(Samoyed) Name: " + name + ", Age: " + age;
    }

    @ Override
    public void whereFrom(){System.out.println("Siberia.");}

    @ Override
    public void run(){System.out.println("Samoyed pulls the sled.");}

    public void feature(){System.out.println("Always smiling.");}
}
```

PersianCat.java

```java
package package1;
public final class PersianCat extends Cat implements Pet{

    public int age;

    public PersianCat(int age, String name){ super(name); this.age = age; }

    public String toString(){
       return "(Persian cat) Name: " + name + ", Age: " + age;
    }

    @ Override
    public void whereFrom(){System.out.println("Khorasan");}

    @ Override
    public void sleep(){
       System.out.println("Persian cat sleeps 14 hours a day.");
    }

    public void feature(){System.out.println("Long-haired");}
}
```

Question3.java

```java
package package1;

import package2.Husky;

public class Question3{
    public static void main(String[] args){

        PersianCat myPersianCat = new PersianCat(2, "Kitten");
        Samoyed mySamoyed = new Samoyed(4, "Puppy");

        // System.out.println(myPersianCat);
        // System.out.println(mySamoyed);


        Cat cat = myPersianCat;
        Dog dog = mySamoyed;

        // System.out.println(dog.age);

        // cat.sleep();
        // dog.run();
        // dog.nameOrigin();


        Animal[] listOfAnimals = new Animal[] {myPersianCat, mySamoyed};
        Pet[] listOfPets = new Pet[] {myPersianCat, mySamoyed};

        // listOfAnimals[0].whereFrom();

        // ((Dog) listOfPets[1]).sound();


        // Cat newCat = new Cat("Kitty");


        Husky myHusky = new Husky("pup");
        dog = myHusky;
        // dog.whereFrom();
        // dog.nameOrigin();

    }

}
```

The Java files in `package2` are listed below.

Husky.java

```
package package2;

import package1.Dog;
import package1.Pet;

public final class Husky extends Dog implements Pet{
    public Husky(String name){
        super(name);
    }

    public void whereFrom(){ System.out.println("North artic."); }

    public void feature(){ System.out.println("Energetic and fast."); }

    public void nameOrigin(){
       System.out.println("Named after the Eskimos.");
    }
}
```

Shorthair.java

```
package package2;

import package1.Cat;

public final class Shorthair extends Cat implements Pet{

    public Shorthair(String name){ super(); this.name = name; }

    public void sound(){ System.out.println("Meoooow."); }

    private void whereFrom(){
       System.out.println("Europe and North America.");
    }

    public void feature(){ System.out.println("Has short hair."); }
}
```

The file `Question3.java` compiles and runs properly.

(a) (3 pts) The following lines run properly if you uncomment them, please write down the output.

```
// System.out.println(myPersianCat);
// System.out.println(mySamoyed);
```

(b) (1 pts) The following line will lead to a compile-time error if you uncomment it, please specify the reason.

```
// System.out.println(dog.age);
```

(c) (3 pts) The following lines run properly if you uncomment them, please write down the output.

```
// cat.sleep();
// dog.run();
// dog.nameOrigin();
```

(d) (1.5 pts) The following line will lead to a compile-time error if you uncomment it, please specify the reason.

```
// listOfAnimals[0].whereFrom();
```

(e) (1.5 pts) The following line runs properly if you uncomment it, please write down the output.

```
// ((Dog) listOfPets[1]).sound();
```

(f) (1.5 pts) The following line will lead to a compile-time error if you uncomment it, please specify the reason.

```
// Cat newCat = new Cat("Kitty");
```

(g) (3 pts) The following lines run properly if you uncomment them, please write down the output.

```
// dog.whereFrom();
// dog.nameOrigin();
```

(h) Now let us take a close look at the file `Shorthair.java`, this file contains several different kinds of mistakes. For each of the following lines, please specify why there is a mistake.

- (1.5 pts)

  ```
  public Shorthair(String name){ super(); this.name = name; }
  ```

- (1.5 pts)

  ```
  public void sound(){ System.out.println("Meoooow."); }
  ```

- (1.5 pts)

  ```
  private void whereFrom(){
     System.out.println("Europe and North America.");
  }
  ```

Problem 4. *10pts.*

Suppose you are given the following Java code `Question4.java` on Java GUI.

```
import javax.swing.JFrame;
import javax.swing.JPanel;
import java.awt.Dimension;
import java.awt.Graphics;
import java.awt.Color;


class Question4 {
    public static void main(String[] args){
        JFrame frame1 = new JFrame("frame 1");
        frame1.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        Canvas1 canvas1 = new Canvas1();
        frame1.add(canvas1);
        frame1.getContentPane().setPreferredSize(new Dimension(400, 400));
        frame1.pack();
        frame1.setVisible(true);

        JFrame frame2 = new JFrame("frame 2");
        frame2.setDefaultCloseOperation(JFrame.DISPOSE_ON_CLOSE);
        Canvas2 canvas2 = new Canvas2(200, 100);
        frame2.add(canvas2);
        frame2.pack();
        frame2.setVisible(true);
    }
}

class Canvas1 extends JPanel {
    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.BLACK);
        // public abstract void fillRect(int x, int y, int width, int height)
        // Fills the specified rectangle.
        // The left and right edges of the rectangle are at x and x + width - 1.
        // The top and bottom edges are at y and y + height - 1.
        // The resulting rectangle covers an area width pixels by height pixels.
        g.fillRect(0, 0, 100, 200);
    }
}
```

```
class Canvas2 extends JPanel {
    int width; int height;

    Canvas2(int width, int height){
        super();
        this.width = width; this.height = height;
        this.setPreferredSize(new Dimension(width, height));
    }

    protected void paintComponent(Graphics g) {
        super.paintComponent(g);

        g.setColor(Color.BLACK);
        // public abstract void drawLine(int x1, int y1, int x2, int y2)
        // Draws a line, using the current color,
        // between the points (x1, y1) and (x2, y2)
        // in this graphics context's coordinate system.
        g.drawLine(0, 0, width, height);
    }
}
```

This file (`Question4.java`) compiles and runs properly. In the following questions, you will be asked to do some sketching. **You do not need to be very accurate in sketching, a proper sketching that reflects the correct understanding of the code will be awarded full credits.**

(a) (6 pts) When you compile and run `Question4.java`, you will get the following two frames. Please sketch the graphics in both frames in Figure 1 and Figure 2.

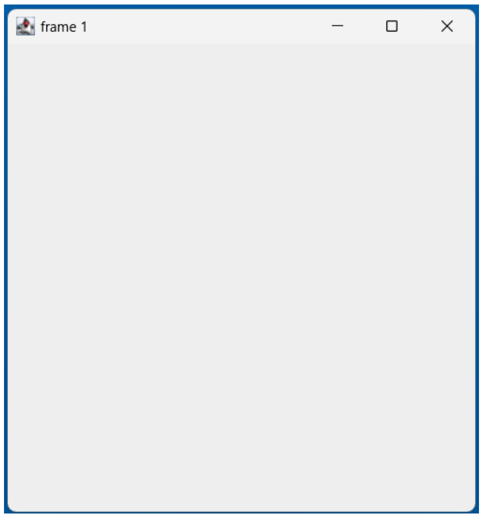

Figure 1: `frame1`

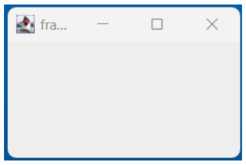

Figure 2: `frame2`

(b) (2 pts) What will happen if we close `frame1` while remaining `frame2` untouched?

A. Both `frame1` and `frame2` will disappear. The program will be terminated.
B. Both `frame1` and `frame2` will disappear. The program will not be terminated.
C. Only `frame1` will disappear. The program will not be terminated.
D. Only `frame2` will disappear. The program will not be terminated.

(c) (2 pts) What will happen if we close `frame2` while remaining `frame1` untouched?

A. Both `frame1` and `frame2` will disappear. The program will be terminated.
B. Both `frame1` and `frame2` will disappear. The program will not be terminated.
C. Only `frame1` will disappear. The program will not be terminated.
D. Only `frame2` will disappear. The program will not be terminated.

Problem 5. *10pts.*

Suppose you are given the following Java code `Question5.java` on Java GUI.

```
import javax.swing.JFrame;
import javax.swing.JButton;
import java.awt.GridLayout;
import java.awt.event.ActionListener;
import java.awt.event.ActionEvent;


public class Question5 {
    public static void main(String[] args) {
        JFrame frame = new JFrame("Listening for action");
        frame.setDefaultCloseOperation(JFrame.EXIT_ON_CLOSE);
        // public GridLayout(int rows, int cols)
        // Creates a grid layout with the specified number of rows and columns.
        // All components in the layout are given equal size.
        frame.setLayout(new GridLayout(2, 2));

        JButton button1 = new JButton("Button 1");
        JButton button2 = new JButton("Button 2");
        JButton button3 = new JButton("Button 3");
        JButton button4 = new JButton("Button 4");

        ActionListener reporter1 = new ReportToConsole(555);
        ActionListener reporter2 = new ReportToConsole(444);
        ActionListener reporter3 = new ReportToConsole(777);
        ActionListener reporter4 = new ReportToConsole(999);

        button1.addActionListener(reporter1);
        button2.addActionListener(reporter2);
        button3.addActionListener(reporter3);
        button4.addActionListener(reporter4);

        frame.add(button1);
        frame.add(button2);
        frame.add(button3);
        frame.add(button4);

        frame.setSize( 400,  400);
        frame.setVisible(true);
    }
}
```

```
class ReportToConsole implements ActionListener {
    int ID;

    ReportToConsole(int ID) {this.ID = ID;}

    public void actionPerformed(ActionEvent e) {
        System.out.println("Reporter ID: " + ID);
        Object source = e.getSource();
        if (source instanceof JButton) {
            JButton sourceButton = (JButton) source;
            if (ID % 2 == 0){
                System.out.println("Listened to: " + sourceButton.getText());
            }
        }
    }
}
```

This file (`Question5.java`) compiles and runs properly. In the following questions, you will be asked to do some sketching. **You do not need to be very accurate in sketching, a proper sketching that reflects the correct understanding of the code will be awarded full credits.**

(a) (4 pts) When you compile and run `Question5.java`, you will get the following frame. Please sketch the buttons on this frame in Figure 3.

Reminder: Do not forget the text on the buttons.

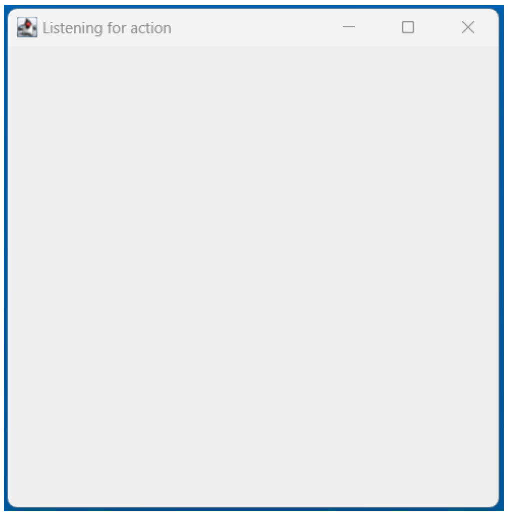

Figure 3: `frame`

(b) (1.5 pts) Suppose we click on `Button 1`. Write down the output in the console in the following box.

(c) (1.5 pts) Suppose we click on `Button 2`. Write down the output in the console in the following box (NO need to include the previous output.)

(d) (1.5 pts) Suppose we click on `Button 3`. Write down the output in the console in the following box (NO need to include the previous output.)

(e) (1.5 pts) Suppose we click on `Button 4`. Write down the output in the console in the following box (NO need to include the previous output.)

Problem 6. *10pts.*

Question a The following Java code compiles and runs properly.

```
class SquareRootOfNegativeNumberException extends ArithmeticException {
    SquareRootOfNegativeNumberException(double x) {
        super("Cannot take the square root of negative number " + x);
    }
}

class Question6a{
    public static double squareRoot(double x){
        if(x<0){ throw new SquareRootOfNegativeNumberException(x); }
        else{ return Math.sqrt(x); }
    }

    public static void main(String[] args){
        try{
            try{
                System.out.println("1");
                byte[] byteArray = new byte[-1];
            }
            catch(IndexOutOfBoundsException e){
                System.out.println("2");
                int[] intArray = new int[3];
                intArray[3] = 1;
            }
            catch(NegativeArraySizeException e){
                System.out.println("3");
                double y = squareRoot(-8.0);
            }
            catch(ArithmeticException e){
                System.out.println("4");
                double z = squareRoot(4.0);
            }
        }
        catch(IndexOutOfBoundsException e){
            System.out.println("5");
        }
        catch(NegativeArraySizeException e){
            System.out.println("6");
        }
        catch(SquareRootOfNegativeNumberException e){
            System.out.println("7");
        }
        finally{
            System.out.println("8");
        }
    }
}
```

(6 pts) Please write down the output below.

Question b The following Java code compiles and runs properly.

```
import java.io.IOException;
import java.io.FileNotFoundException;

class Question6b{

    // public static void f() { throw new FileNotFoundException(); }

    public static void g() throws IOException{
       System.out.println("But no IOExceptions are thrown out.");
    }

    // public static void h(){ g(); }
}
```

For the lines in the `Question6b` class that are commented, consider what happens when you uncomment them while **leaving the other lines commented**.

- Either tell that uncommenting the line will **lead** to a compile-time error;
- Or tell that uncommenting the line will **not lead** to any compile-time error;

You do **NOT** need to explain any reasons.

(a) (2 pts)

```
// public static void f() { throw new FileNotFoundException(); }
```

(b) (2 pts)

```
// public static void h(){ g(); }
```

(END OF EXAM)

BLANK PAGE

BLANK PAGE