Question 1:

myObject1.printX();

Value of x : 5

myObject1.incrementCount();

count =1;

MyClass.incrementCount();

count=1;

myObject1.printCount();

Value of count : 1

myObject2.printCount();

Value of count : 0

myObject2.printX();

Value of x : 7

myObject1.setX(14);

Within Object1, x is set to 14.

myObject1.incrementCount();

count=3;

myObject1.printX();

Value of x : 14

myObject1.printCount();

Value of count : 3

myObject2.printCount();

Value of count : 0

Question 2:

Clock.java

public class Clock {  
 private int hours;  
 private int minutes;  
 private int seconds;  
  
 public Clock(){  
 hours=12;  
 minutes=0;  
 seconds=0;  
 }  
 public Clock(int hours, int minutes, int seconds){  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public Clock(int seconds){  
 int hours, minutes, remainingSeconds;  
 hours=seconds/3600;  
 System.*out*.println(hours);  
 //there are 3600 seconds in an hour  
 remainingSeconds=seconds%3600;  
 //there are 60 seconds in 1 minute  
 minutes=remainingSeconds/60;  
 System.*out*.println(minutes);  
 remainingSeconds=remainingSeconds%60;  
 seconds=remainingSeconds;  
 System.*out*.println(seconds);  
  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public void setClock(int seconds){  
 int hours, minutes, remainingSeconds;  
 hours=seconds/3600;  
 //there are 3600 seconds in an hour  
 remainingSeconds=seconds%3600;  
 //there are 60 seconds in 1 minute  
 minutes=remainingSeconds/60;  
 remainingSeconds=remainingSeconds%60;  
 seconds=remainingSeconds;  
  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public int getHours(){  
 return hours;  
 }  
 public int getMinutes(){  
 return minutes;  
 }  
 public int getSeconds(){  
 return seconds;  
 }  
 public void setHours(int newHours){  
 hours=newHours;  
  
 }  
 public void setMinutes(int newMinutes){  
 minutes=newMinutes;  
 }  
 public void setSeconds(int newSeconds){  
 seconds=newSeconds;  
 }  
 public void tick(){  
 seconds++;  
 }  
 public void tickDown(){  
 if(seconds>0) seconds--;  
 //seconds could end up being negative  
 }  
 public void addClock(Clock clock){  
 int clockHours=clock.getHours();  
 int clockMinutes=clock.getMinutes();  
 int clockSeconds=clock.getSeconds();  
  
 int totalHours= hours+ clockHours;  
 int totalMinutes= minutes +clockMinutes;  
 int totalSeconds=seconds +clockSeconds;  
  
 //figure out the seconds  
 if(totalSeconds<60 && totalSeconds>=0){  
 seconds=totalSeconds;  
 }else{  
 //total seconds is greater than or equal to 60  
 int newMinutes=totalSeconds/60;  
 seconds=totalSeconds%60;  
 //seconds=0;  
 totalMinutes+=newMinutes;  
 }  
 //figure out the minutes  
 if(totalMinutes>=0 && totalMinutes<60){  
 minutes=totalMinutes;  
 }else{  
 //minutes are greater than or equal to an hour  
 int newHours=totalMinutes/60;  
 minutes=totalMinutes%60;  
 totalHours+=newHours;  
 }  
 //figure out hours  
 if(totalHours>=0 && totalHours<=23){  
 hours=totalHours;  
 }else{  
 //hours are greater than 23, must reset the clock  
 hours= totalHours-24;  
 }  
 }  
 public Clock subtractClock(Clock clock){  
 Clock newClock;  
 int clockHours=clock.getHours();  
 int clockMinutes=clock.getMinutes();  
 int clockSeconds=clock.getSeconds();  
  
 int totalHours= Math.*abs*(hours- clockHours);  
 int totalMinutes= Math.*abs*(minutes-clockMinutes);  
 int totalSeconds=Math.*abs*(seconds-clockSeconds);  
  
 int tempSeconds, tempMinutes, tempHours;  
  
 //figure out the seconds  
 if(totalSeconds<60 && totalSeconds>=0){  
 //seconds=totalSeconds;  
  
 tempSeconds=totalSeconds;  
 }else{  
 //total seconds is greater than or equal to 60  
 int newMinutes=totalSeconds/60;  
 //seconds=totalSeconds%60;  
  
 tempSeconds=totalSeconds%60;  
 //seconds=0;  
 totalMinutes+=newMinutes;  
 }  
 //figure out the minutes  
 if(totalMinutes>=0 && totalMinutes<60){  
 tempMinutes=totalMinutes;  
 //minutes=totalMinutes;  
 }else{  
 //minutes are greater than or equal to an hour  
 int newHours=totalMinutes/60;  
 tempMinutes=totalMinutes%60;  
 // minutes=totalMinutes%60;  
 totalHours+=newHours;  
 }  
 //figure out hours  
 if(totalHours>=0 && totalHours<=23){  
 tempHours=totalHours;  
 //hours=totalHours;  
 }else{  
 //hours are greater than 23, must reset the clock  
 tempHours=totalHours-24;  
 //hours= totalHours-24;  
 }  
 newClock = new Clock(tempSeconds, tempMinutes, tempHours);  
 return newClock;  
  
  
 }  
 public String toString(){  
 String valHours=String.*valueOf*(hours);  
 String valMinutes=String.*valueOf*(minutes);  
 String valueSeconds=String.*valueOf*(seconds);  
 if(valHours.length()<2){  
 valHours = "0"+valHours;  
 }  
 if(valMinutes.length()<2){  
 valMinutes="0"+ valMinutes;  
 }  
 if(valueSeconds.length()<2){  
 valueSeconds="0"+valueSeconds;  
 }  
 return "(" +valHours + ":" + valMinutes + ":"+  
 valueSeconds+ ")";  
 }  
  
  
}

ClockDemo.java

public class Clock {  
 private int hours;  
 private int minutes;  
 private int seconds;  
  
 public Clock(){  
 hours=12;  
 minutes=0;  
 seconds=0;  
 }  
 public Clock(int hours, int minutes, int seconds){  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public Clock(int seconds){  
 int hours, minutes, remainingSeconds;  
 hours=seconds/3600;  
 System.*out*.println(hours);  
 //there are 3600 seconds in an hour  
 remainingSeconds=seconds%3600;  
 //there are 60 seconds in 1 minute  
 minutes=remainingSeconds/60;  
 System.*out*.println(minutes);  
 remainingSeconds=remainingSeconds%60;  
 seconds=remainingSeconds;  
 System.*out*.println(seconds);  
  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public void setClock(int seconds){  
 int hours, minutes, remainingSeconds;  
 hours=seconds/3600;  
 //there are 3600 seconds in an hour  
 remainingSeconds=seconds%3600;  
 //there are 60 seconds in 1 minute  
 minutes=remainingSeconds/60;  
 remainingSeconds=remainingSeconds%60;  
 seconds=remainingSeconds;  
  
 this.hours=hours;  
 this.minutes=minutes;  
 this.seconds=seconds;  
  
 }  
 public int getHours(){  
 return hours;  
 }  
 public int getMinutes(){  
 return minutes;  
 }  
 public int getSeconds(){  
 return seconds;  
 }  
 public void setHours(int newHours){  
 hours=newHours;  
  
 }  
 public void setMinutes(int newMinutes){  
 minutes=newMinutes;  
 }  
 public void setSeconds(int newSeconds){  
 seconds=newSeconds;  
 }  
 public void tick(){  
 seconds++;  
 }  
 public void tickDown(){  
 if(seconds>0) seconds--;  
 //seconds could end up being negative  
 }  
 public void addClock(Clock clock){  
 int clockHours=clock.getHours();  
 int clockMinutes=clock.getMinutes();  
 int clockSeconds=clock.getSeconds();  
  
 int totalHours= hours+ clockHours;  
 int totalMinutes= minutes +clockMinutes;  
 int totalSeconds=seconds +clockSeconds;  
  
 //figure out the seconds  
 if(totalSeconds<60 && totalSeconds>=0){  
 seconds=totalSeconds;  
 }else{  
 //total seconds is greater than or equal to 60  
 int newMinutes=totalSeconds/60;  
 seconds=totalSeconds%60;  
 //seconds=0;  
 totalMinutes+=newMinutes;  
 }  
 //figure out the minutes  
 if(totalMinutes>=0 && totalMinutes<60){  
 minutes=totalMinutes;  
 }else{  
 //minutes are greater than or equal to an hour  
 int newHours=totalMinutes/60;  
 minutes=totalMinutes%60;  
 totalHours+=newHours;  
 }  
 //figure out hours  
 if(totalHours>=0 && totalHours<=23){  
 hours=totalHours;  
 }else{  
 //hours are greater than 23, must reset the clock  
 hours= totalHours-24;  
 }  
 }  
 public Clock subtractClock(Clock clock){  
 Clock newClock;  
 int clockHours=clock.getHours();  
 int clockMinutes=clock.getMinutes();  
 int clockSeconds=clock.getSeconds();  
  
 int totalHours= Math.*abs*(hours- clockHours);  
 int totalMinutes= Math.*abs*(minutes-clockMinutes);  
 int totalSeconds=Math.*abs*(seconds-clockSeconds);  
  
 int tempSeconds, tempMinutes, tempHours;  
  
 //figure out the seconds  
 if(totalSeconds<60 && totalSeconds>=0){  
 //seconds=totalSeconds;  
  
 tempSeconds=totalSeconds;  
 }else{  
 //total seconds is greater than or equal to 60  
 int newMinutes=totalSeconds/60;  
 //seconds=totalSeconds%60;  
  
 tempSeconds=totalSeconds%60;  
 //seconds=0;  
 totalMinutes+=newMinutes;  
 }  
 //figure out the minutes  
 if(totalMinutes>=0 && totalMinutes<60){  
 tempMinutes=totalMinutes;  
 //minutes=totalMinutes;  
 }else{  
 //minutes are greater than or equal to an hour  
 int newHours=totalMinutes/60;  
 tempMinutes=totalMinutes%60;  
 // minutes=totalMinutes%60;  
 totalHours+=newHours;  
 }  
 //figure out hours  
 if(totalHours>=0 && totalHours<=23){  
 tempHours=totalHours;  
 //hours=totalHours;  
 }else{  
 //hours are greater than 23, must reset the clock  
 tempHours=totalHours-24;  
 //hours= totalHours-24;  
 }  
 newClock = new Clock(tempSeconds, tempMinutes, tempHours);  
 return newClock;  
  
  
 }  
 public String toString(){  
 String valHours=String.*valueOf*(hours);  
 String valMinutes=String.*valueOf*(minutes);  
 String valueSeconds=String.*valueOf*(seconds);  
 if(valHours.length()<2){  
 valHours = "0"+valHours;  
 }  
 if(valMinutes.length()<2){  
 valMinutes="0"+ valMinutes;  
 }  
 if(valueSeconds.length()<2){  
 valueSeconds="0"+valueSeconds;  
 }  
 return "(" +valHours + ":" + valMinutes + ":"+  
 valueSeconds+ ")";  
 }  
  
  
}

Please enter seconds greater than 0

15689

15689

4

21

29

30 31 32 33 34 35 36 37 38 39

Please enter hours within the range of 0 and 23

22

22

Please enter minutes within the range of 0 and 59

45

45

Please enter seconds within the range of 0 and 59

16

16

17 18 19 20 21 22 23 24 25 26 (03:07:05)

(22:45:26)

Process finished with exit code 0

Question 3

public class Complex {  
 private double realPart;  
 private double imaginaryPart;  
  
 public Complex(){  
 realPart=0.0;  
 imaginaryPart=0.0;  
 }  
 public Complex(double realPart, double imaginaryPart){  
 this.realPart=realPart;  
 this.imaginaryPart=imaginaryPart;  
 }  
 public Complex add(Complex otherNumber){  
 return new Complex(realPart+ otherNumber.realPart, imaginaryPart  
 + otherNumber.imaginaryPart);  
 }  
 public Complex subtract(Complex otherNumber){  
 return new Complex(realPart-otherNumber.realPart, imaginaryPart  
 - otherNumber.imaginaryPart);  
 }  
 public Complex multiply (Complex otherNumber){  
 return new Complex(realPart\*otherNumber.realPart, imaginaryPart  
 \* otherNumber.imaginaryPart);  
 }  
 public Complex divide (Complex otherNumber){  
 return new Complex(realPart/otherNumber.realPart, imaginaryPart  
 / otherNumber.imaginaryPart);  
  
 }  
 public void setRealPart (double realPart) {  
 this.realPart=realPart;  
 }  
 public void setImaginaryPart (double imaginaryPart) {  
 this.imaginaryPart=imaginaryPart;  
 }  
 public double getRealPart(){  
 return realPart;  
 }  
  
 public double getImaginaryPart(){  
 return imaginaryPart;  
 }  
  
 public String toString(){  
 return "The real part is " + realPart + " The imaginary part is " + imaginaryPart;  
  
 }  
  
}

import java.util.Scanner;  
public class ComplexDemo {  
 public static void main(String [] args){  
 int realPartOne, imaginaryPartOne, realPartTwo, imaginaryPartTwo;  
 Complex complexOne, complexTwo;  
 Scanner console=new Scanner(System.*in*);  
 System.*out*.println("Please enter a real number");  
 realPartOne=console.nextInt();  
  
 System.*out*.println("Please enter an imaginary number");  
 imaginaryPartOne=console.nextInt();  
  
 complexOne=new Complex(realPartOne, imaginaryPartOne);  
  
 System.*out*.println("Please enter a second real number");  
 realPartTwo=console.nextInt();  
  
 System.*out*.println("Please enter a second imaginary number");  
 imaginaryPartTwo= console.nextInt();  
  
 complexTwo=new Complex(realPartTwo, imaginaryPartTwo);  
  
 //Testing getters and setters  
 System.*out*.println(complexOne.getRealPart());  
 System.*out*.println(complexOne.getImaginaryPart());  
  
 //testing addition  
 Compl

ex temp= complexOne.add(complexTwo);  
 System.*out*.println(temp.toString());  
  
 //testing subtraction  
 temp=complexOne.subtract(complexTwo);  
 System.*out*.println(temp.toString());  
  
 //testing multiplication  
 temp=complexOne.multiply(complexTwo);  
 System.*out*.println(temp.toString());  
  
 //testing division  
 temp=complexOne.divide(complexTwo);  
 System.*out*.println(temp.toString());  
  
 }  
}

Please enter a real number

5

Please enter an imaginary number

6

Please enter a second real number

5

Please enter a second imaginary number

3

5.0

6.0

The real part is 10.0The imaginary part is 9.0

The real part is 0.0The imaginary part is 3.0

The real part is 25.0The imaginary part is 18.0

The real part is 1.0The imaginary part is 2.0

Process finished with exit code 0

Question 4

Author

public class Author {  
 private String firstName;  
 private String lastName;  
  
 public Author(String firstName, String lastName){  
 this.firstName=firstName;  
 this.lastName=lastName;  
 }  
 public void setFirstName (String firstName) {  
 this.firstName=firstName;  
 }  
  
 public void setLastName (String lastName){  
 this.lastName=lastName;  
  
 }  
 public String getFirstName(){  
 return firstName;  
 }  
 public String getLastName(){  
 return lastName;  
 }  
 public String toString() {  
 return "The author's name is " + firstName + " " + lastName + ".";  
 }  
}

Book

public class Book {  
 private String title;  
 private String author;  
 private double price;  
  
 public Book(String title, String author, double price){  
 this.title=title;  
 this.author=author;  
 this.price=price;  
 }  
 public void setTitle(String title){  
 this.title=title;  
  
 }  
 public void setAuthor(String author){  
 this.author=author;  
 }  
 public void setPrice(double price){  
 this.price=price;  
 }  
 public String getTitle(){  
 return title;  
 }  
 public String getAuthor(){  
 return author;  
  
 }  
 public String toString(){  
 return "The title of this book is " + title +  
 ", the author is " + author + ", and the price of this book is "+  
 price + ".";  
 }  
}

Book Demo

import java.util.Scanner;  
public class BookDemo {  
 public static void main (String[] args){  
 Scanner console= new Scanner(System.*in*);  
 Book example= new Book("Developing Java Software", "Russel Winderand", 79.75);  
 System.*out*.println(example.toString());  
 }  
}

The title of this book is Developing Java Software, the author is Russel Winderand, and the price of this book is 79.75.

Process finished with exit code 0