Fundamentals of Programming

The assignments are Programming Exercises 1.1 on page 30, 2.3 on page 69, 3.13 on page 110, 4.23 on pages 156-157, 5.13 on page 196 of the textbook (Chapters 1-5). For your convenience, the exercises are shown below, where 3.13 is modified slightly. Please copy here your source codes, including your input and output screenshots. Please upload this document along with your source files (i.e., the .java files) on Blackboard by the due date.

1.1 (*Display three messages*) Write a program that displays Welcome to Java, Welcome to Computer Science, and Programming is fun.

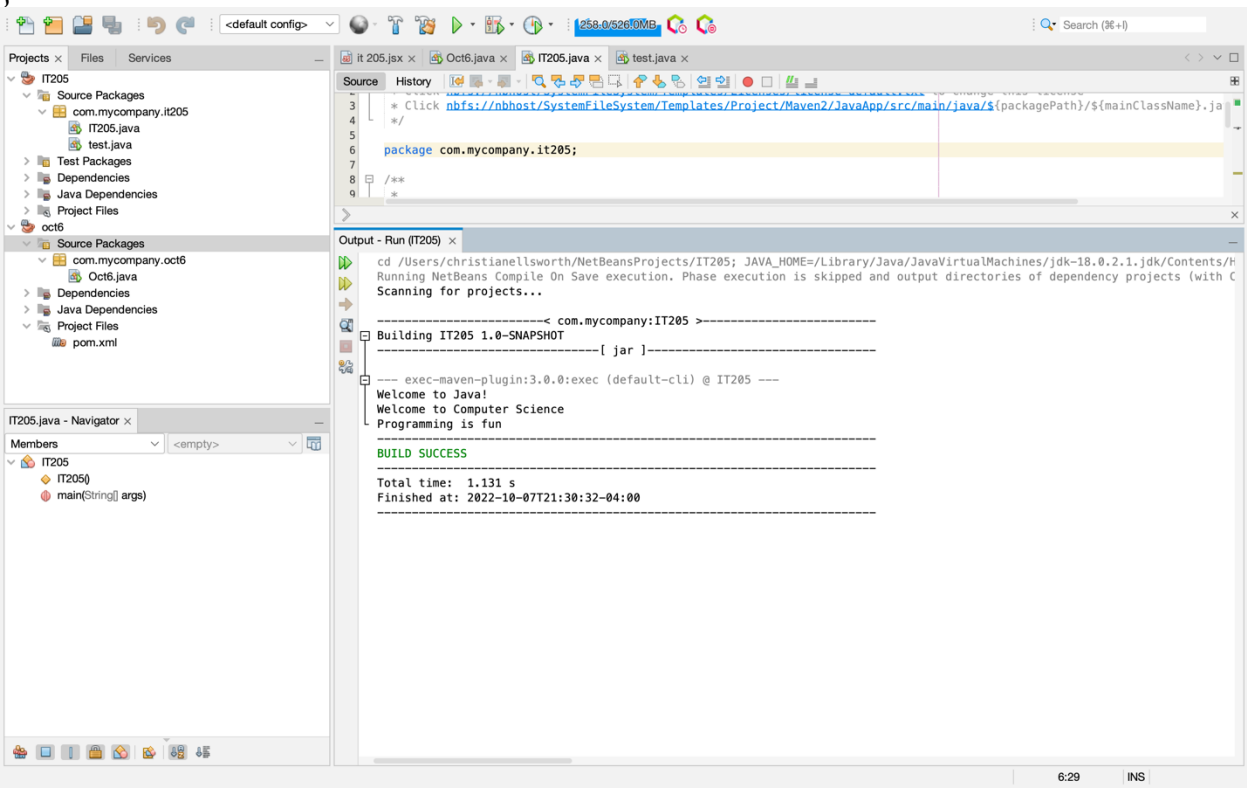
```
/**
 *
 * @author christianellsworth
 */
public class IT205 {


    public static void main(String[] args) {
        System.out.println("Welcome to Java!");


        System.out.println("Welcome to Computer Science");


        System.out.println("Programming is fun");
    }
```

}



2.3 (*Convert feet into meters*) Write a program that reads a number in feet, converts it to meters, and displays the result. One foot is 0.305 meter. Here is a sample run:

Enter a value for feet: 16.5

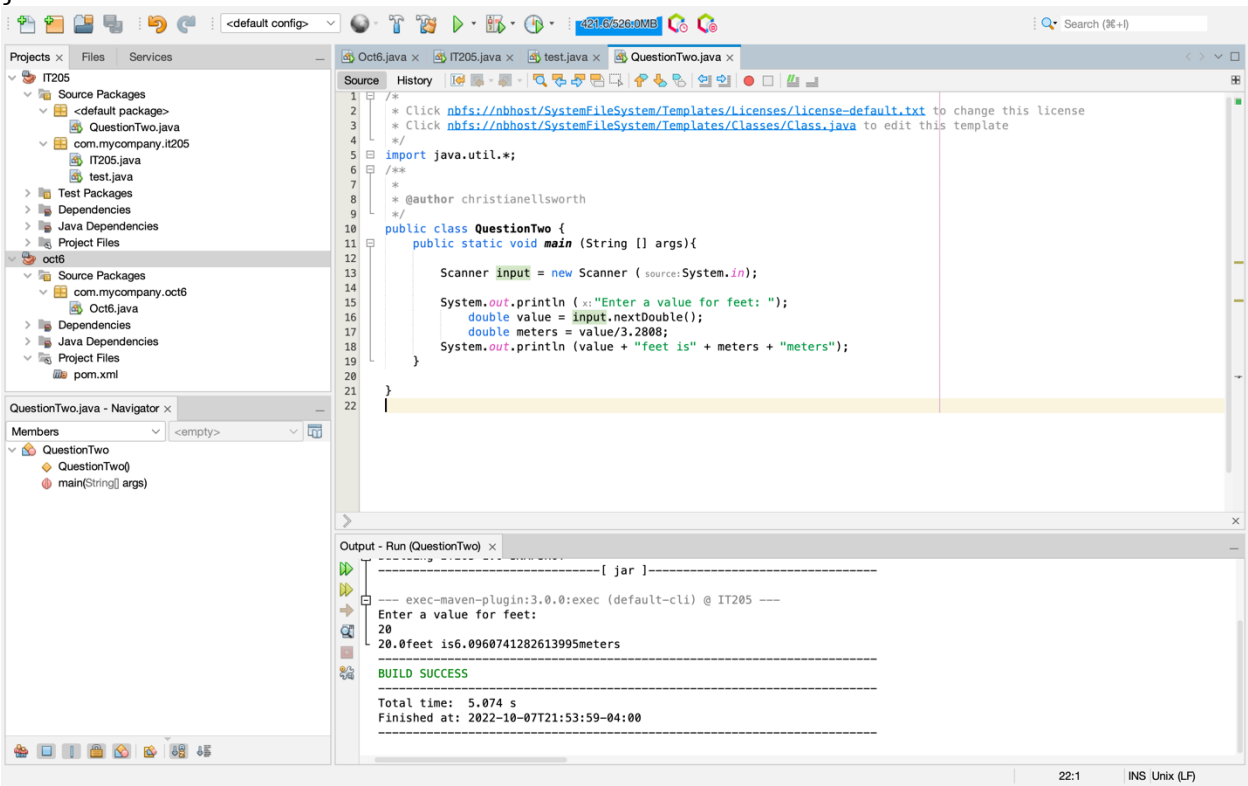
16.5 feet is 5.0325 meters

```
import java.util.*;
/**
 *
 * @author christianellsworth
 */
public class QuestionTwo {
    public static void main (String [] args){

        Scanner input = new Scanner (System.in);

        System.out.println ("Enter a value for feet: ");
            double value = input.nextDouble();
```

```
        double meters = value/3.2808;
    System.out.println (value + "feet is" + meters + "meters");
  }
}
```



3.13 (*Financial application: compute taxes*) The U.S. federal personal income tax is calculated based on filing status and taxable income. There are four filing statuses: single filers, married filing jointly or qualified widow(er), married filing separately, and head of household. The tax rates vary every year. The following table shows the rates for 2020.

With a marginal tax rate, you pay that rate only for the amount of your income that falls into a certain range. To understand how marginal rates work, consider the bottom tax rate of 10%. For single filers, all income between $0 and $9,700 is subject to a 10% tax rate. If you have $10,000 in taxable income, the first $9,700 is subject to the 10% rate, and the remaining $300 is subject to the tax rate of the next bracket (12%). Check the following table to see what your top marginal tax rate is.

Tax Rate	Single	Married Filing Jointly	Married Filing Separately	Head of Household
10%	$0 – $9,875	$0 – $19,750	$0 – $9,875	$0 – $14,100
12%	$9,876 – $40,125	$19,751 – $80,250	$9,876 – $40,125	$14,101 – $53,700
22%	$40,126 – $85,525	$80,251 – $171,050	$40,126 – $85,525	$53,701 – $85,500
24%	$85,526 – $163,300	$171,051 – $326,600	$85,526 – $163,300	$85,501 – $163,300

32%	$163,301 – $207,350	$326,601 – $414,700	$163,301 – $207,350	$163,301 – $207,350
35%	$207,351 – $518,400	$414,701 – $622,050	$207,351 – $518,400	$207,351 – $518,400
37%	$518,401+	$622,051+	$518,401+	$518,401+

Using the above table, write a program to compute your personal income tax for the last calendar year. Your program should prompt the user to enter the filing status and taxable income, and compute the tax. Enter 0 for single filers, 1 for married filing jointly or qualified widow(er), 2 for married filing separately, and 3 for head of household. If you need any hints, refer to Listing 3.5 ComputeTax.java, which gives the source code to compute taxes for single filers.

```
import java.util.Scanner;
/**
 *
 * @author christianellsworth
 */
public class question3 {
      public static void main (String [] args) {

      Scanner input = new Scanner(System.in);

   System.out.print ("0-single filer, 1-married jointly or qualifying widow(er),"
         + "\n2-married seperately, 3-head of household)\n"
         + "Enter the filing status: ");
   int status =  input.nextInt();

   System.out.print("Enter the taxable income: ");
   double income = input.nextDouble();

   double tax = 0;
   switch (status)
   {
      case 0:
               tax += (income <= 8350) ? income * 0.10 : 8350 * 0.10;
               if (income > 8350)
                  tax += (income <= 8350) ? (income - 8350) * 0.15 :
                        25600 * 0.15;
```

```
        if (income > 33950)
           tax += (income <= 82250) ? (income - 33950) * 0.25 :
                 48300 * 0.25;
        if (income > 82250)
           tax += (income <= 171500) ?  (income - 82250) * 0.28 :
                 89300 * 0.28;
        if (income > 171500)
           tax += (income <= 372950) ?  (income - 171500) * 0.33 :
                 201400 * 0.33;
        if (income > 372950)
           tax += (income - 372950) * 0.35;
        break;

case 1:
        tax += (income <= 16700) ? income * 0.10 : 16700 * 0.10;
        if (income > 16700)
           tax += (income <= 67900) ?  (income - 16700) * 0.15 :
                 (67900 -16700) * 0.15;
        if (income > 67900)
           tax += (income <= 137050) ?  (income - 67900) * 0.25 :
                 (137050 -67900) * 0.25;
        if (income > 137050)
           tax += (income <= 208850) ?  (income - 137050) * 0.28 :
                 (208850 - 137050) * 0.28;
        if (income > 208850)
           tax += (income <= 372950) ?  (income - 208850) * 0.33 :
                 (372950 - 208850) * 0.33;
        if (income > 372950)
           tax += (income - 372950) *0.35;
        break;

case 2:
        tax+= (income <= 8350) ? income * 0.10 : 8350 * 0.10;
```

```
        if (income > 8350)
           tax += (income <= 33950) ?  (income - 8350) * 0.15 :
                  (33950 - 8350) * 0.15;
        if (income > 33950)
           tax += (income <= 68525) ?  (income - 33950) * 0.25 :
                  (68525 - 33950) * 0.25;
        if (income > 68525)
           tax += (income <= 104425) ?  (income - 68525) * 0.28 :
                  (104425 - 68525) * 0.28;
        if (income > 104425)
           tax += (income <= 186475) ?  (income - 104425) * 0.33 :
                  (186475 - 104425) * 0.33;
        if (income > 186475)
           tax+= (income - 186475) * 0.35;
        break;

case 3:
        tax += (income <= 11950) ? income * 0.10 : 11950 * 0.10;
        if (income > 11950)
           tax += (income <= 45500) ?  (income - 11950) * 0.15 :
                  (45500 - 11950) * 0.15;
        if (income > 45500)
           tax += (income <= 117450) ?  (income - 45500) * 0.25 :
                  (117450 - 45500) * 0.25;
        if (income > 117450)
           tax += (income <= 190200) ?  (income - 117450) * 0.28 :
                  (190200 - 117450) * 0.28;
        if (income > 190200)
           tax += (income <= 372950) ?  (income - 190200) * 0.33 :
                  (372950 - 190200) * 0.33;
        if (income > 372950)
           tax += (income - 372950) * 0.35;
        break;
```

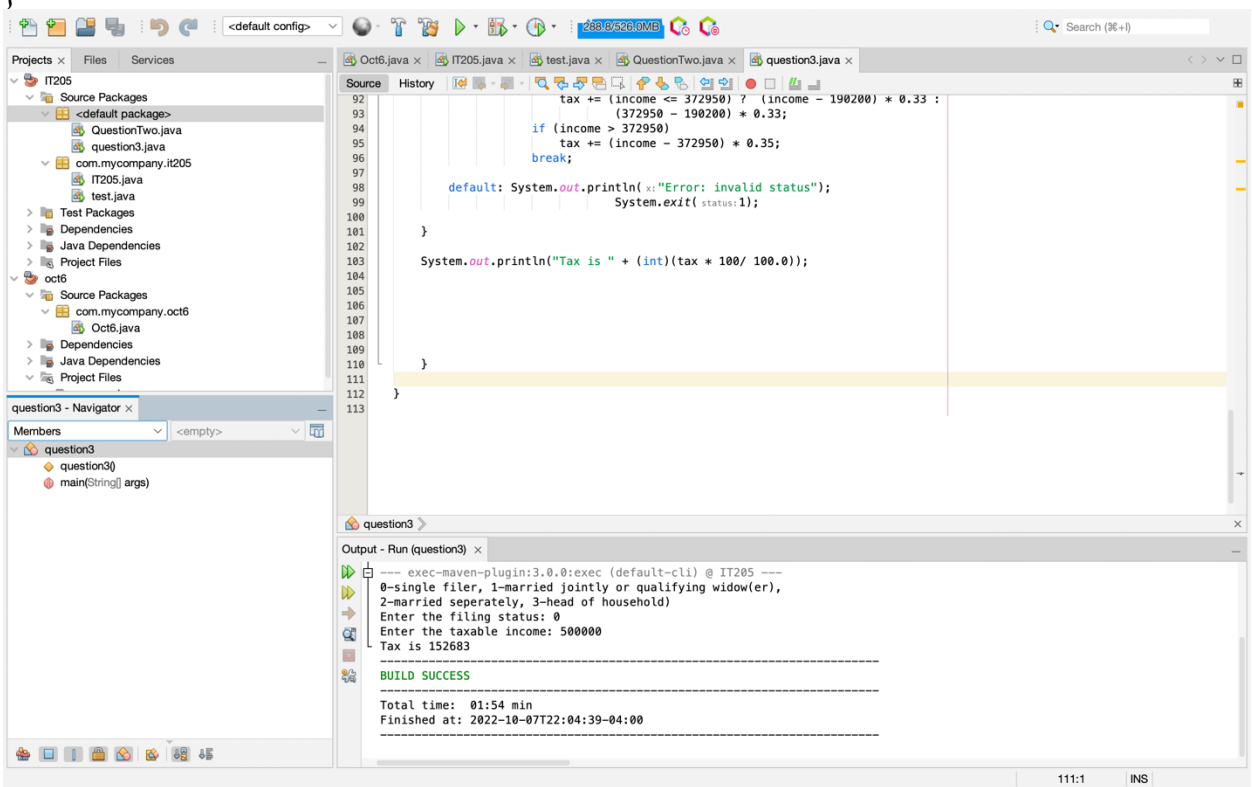
```
        default: System.out.println("Error: invalid status");
                        System.exit(1);

    }

   System.out.println("Tax is " + (int)(tax * 100/ 100.0));



    }
}
```



4.23 (*Financial application: payroll*) Write a program that reads the following information and prints a payroll statement:

Employee's name (e.g., Smith)

Number of hours worked in a week (e.g., 10)

Hourly pay rate (e.g., 9.75)

Federal tax withholding rate (e.g., 20%)

State tax withholding rate (e.g., 9%)

A sample run is as follows:

```
Enter employee's name: Smith
Enter number of hours worked in a week: 10
Enter hourly pay rate: 9.75
Enter federal tax withholding rate: 0.20
Enter state tax withholding rate: 0.09

Employee Name: Smith
Hours worked: 10.0
Pay Rate: $9.75
Gross Pay: $97.5
Deductions:
  Federal Withholding (20.0%): $19.5
  State Withholding (9.0%): $8.77
  Total Deduction: $28.27
Net Pay: $69.22
```

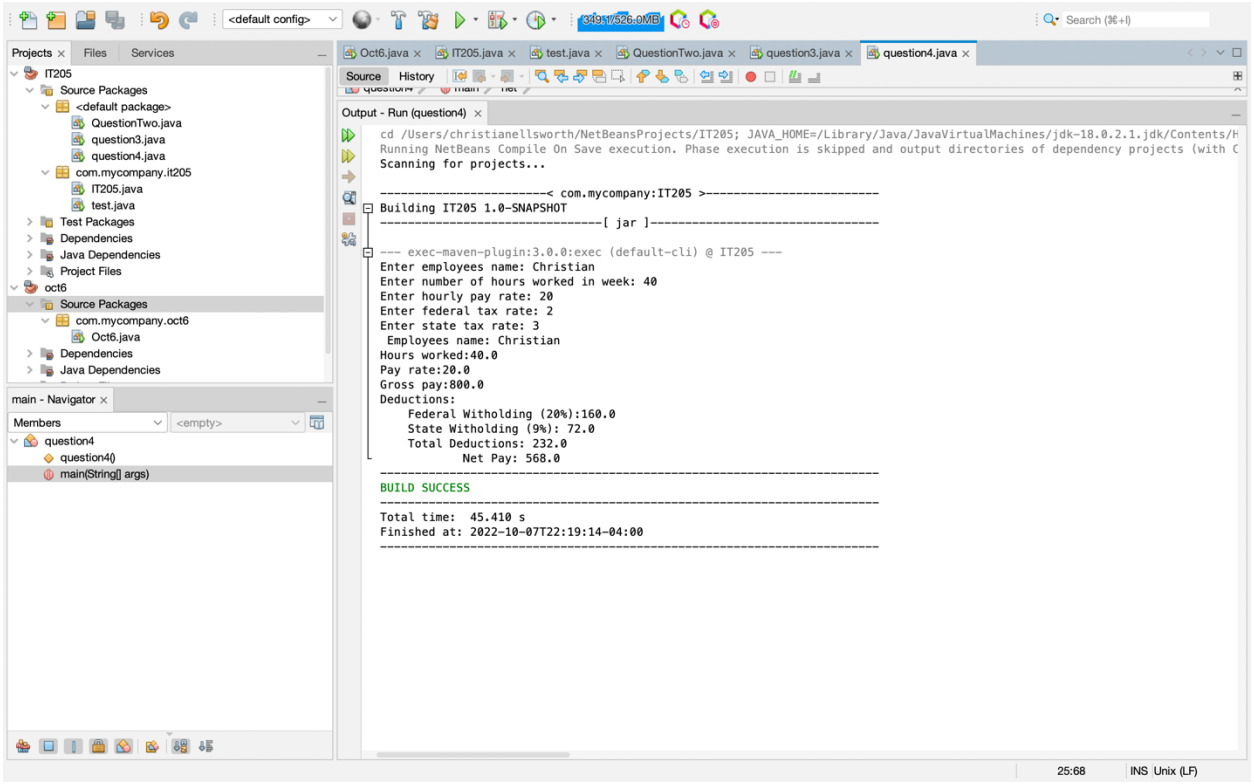
CODE:

```
  import java.util.Scanner;
/**
 *
 * @author christianellsworth
 */
public class question4 {
      public static void main(String[] args) {
      Scanner input = new Scanner(System.in);
      System.out.print("Enter employees name: ");
```

```
        String name = input.next();
    System.out.print("Enter number of hours worked in week: ");
        double  hoursWorked = input.nextDouble();
    System.out.print("Enter hourly pay rate: ");
        double rate = input.nextDouble();
    System.out.print("Enter federal tax rate: ");
        double federalTaxRate = input.nextDouble();
    System.out.print("Enter state tax rate: ");
        double stateTaxRate = input.nextDouble();
        double total = ((hoursWorked * rate ) * (0.20))
                + ((hoursWorked * rate) * (0.09));
        double net = ((hoursWorked * rate) - (((hoursWorked * rate) * (0.20))
                +((hoursWorked * rate) * (0.09))));

        System.out.println(" Employees name: " + name);
        System.out.println("Hours worked:" + hoursWorked);
        System.out.println("Pay rate:" + rate);
        System.out.println("Gross pay:" + hoursWorked * rate);
        System.out.println("Deductions: ");
        System.out.println("    Federal Witholding (20%):" + (hoursWorked * rate) * (.20));
        System.out.println("    State Witholding (9%): " + (hoursWorked * rate) * (0.09));
        System.out.println("    Total Deductions: " + total);
        System.out.println("           Net Pay: " + net);
}


}
```



5.13 (*Find the largest n such that* $n^3 < 12{,}000$) Use a while loop to find the largest integer n such that n^3 is less than 12,000.

```
**
 *
 * @author christianellsworth
 */
public class Question5 {
        public static void main(String[] args)
      {

        int largest=0;
        int count=1;
        while(count*count*count<12000)
        {
           if(count>largest)
                largest=count;
```

```
            count++ ;
        }
        System.out.println("The largest N suchthat N cube<12000 is "+largest);
    }

}
```