# Directions

1. Complete the following programs.
2. Screenshot the running programs. Include enough output to show the program works in it’s entirety.
3. Submit screenshots/copies of the code.
   1. Partial credit can be had if you made a valiant effort
4. Submit to Brightspace

Part 1: Complete Chapter 8 Programming Exercises starting on page 321; provide a snippet of the code and of enough output to show the program works in its entirety:

1. public class NineInts{  
 public static void main(String[] args) {  
  
 int[]numbers ={1,2,3,4,5,6,7,8,9};  
  
 for(int x =numbers.length-1;x>=0;x--)  
 {  
 System.*out*.println(numbers[x]);  
 }  
  
 }  
}

Text

Description automatically generated

2. import java.util.Scanner;  
public class DistanceFromAverage {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 double entry=0.0;  
 double[]numbers = new double[15];  
 double total =0.0;  
 double average =0.0;  
 boolean checking =true;  
  
 for(int x =0; x<15;x++)  
 {  
 System.*out*.println("Enter a number, Enter 99999 if you would like to quit entering numbers");  
 entry = sc.nextDouble();  
  
 if(entry==99999 && numbers[0]==0)  
 {  
 checking=false;  
 break;  
 }  
 else if(entry==99999)  
 {  
 break;  
 }  
  
 numbers[x]=entry;  
 total+=entry;  
 average = total/(x);  
 }  
  
 if(checking==false)  
 {  
 System.*out*.println("Error: No numbers entered");  
 }  
 else {  
  
 System.*out*.println(total);  
 System.*out*.println(average);  
  
 for (int y = 0; y < numbers.length; y++) {  
 if (numbers[y] > 0) {  
 System.*out*.println("Number: " + (y + 1) + " is " + numbers[y]);  
 System.*out*.println("Distance from average: " + (average - numbers[y]));  
 }  
 }  
 }  
  
 sc.close();  
 }  
}

Text

Description automatically generated

4. import java.util.\*;  
  
public class ArrayMethodDemo {  
 public static void main(String[] args) {  
 Scanner sc = new Scanner(System.*in*);  
 int[]numbers = new int[5];  
 int entry = 0;  
  
 for(int x =0; x<numbers.length;x++)  
 {  
 System.*out*.println("Enter an integer value");  
 entry = sc.nextInt();  
 numbers[x]= entry;  
 }  
  
// displayInt(numbers);  
// System.out.println();  
// displayIntReverse(numbers);  
// System.out.println();  
// displaySum(numbers);  
// System.out.println();  
// displayLessTwelve(numbers);  
// System.out.println();  
 *displayAboveAverage*(numbers);  
 }  
  
 public static void displayInt(int[]numbers)  
 {  
 for(int x =0;x<numbers.length;x++)  
 {  
 System.*out*.println("Number:"+ (x+1)+ " "+ numbers[x]);  
 }  
 }  
  
 public static void displayIntReverse(int[]numbers)  
 {  
 for(int x =numbers.length-1;x>=0;x--)  
 {  
 System.*out*.println("Number:"+ (x+1)+ " "+ numbers[x]);  
 }  
 }  
  
 public static void displaySum(int[]numbers)  
 {  
 int sum =0;  
 for(int x =numbers.length-1;x>=0;x--)  
 {  
 sum+=numbers[x];  
 }  
  
 System.*out*.println("Sum: "+ sum);  
 }  
  
 public static void displayLessTwelve(int[]numbers)  
 {  
 boolean check = true;  
 for(int x =numbers.length-1;x>=0;x--)  
 {  
 if(numbers[x]<12) {  
 System.*out*.println("Number: " + numbers[x]);  
 check = false;  
 }  
 }  
 if(check==true)  
 System.*out*.println("No numbers under 12");  
 }  
  
 public static void displayAboveAverage(int[]numbers)  
 {  
 int average = 0;  
 int total =0;  
  
 for(int x =0; x<numbers.length; x++)  
 {  
 total+=numbers[x];  
 }  
  
 average=total/numbers.length;  
  
 for(int y=0; y<numbers.length;y++)  
 {  
 if(numbers[y]>average)  
 System.*out*.println(numbers[y] +" is greater than "+ average);  
 }  
 }  
  
}

Text

Description automatically generated with medium confidence

6.

A) public class CollegeCourse {  
 private String classId;  
 private int hours;  
  
 @Override  
 public String toString() {  
 return "CollegeCourse{" +  
 "classId='" + classId + '\'' +  
 ", hours=" + hours +  
 ", grade=" + grade +  
 '}';  
 }  
  
 private char grade;  
  
 public CollegeCourse(String classId,int hours,char grade)  
 {  
 this.classId = classId;  
 this.hours = hours;  
 this.grade = grade;  
 }  
  
 public String getClassId() {  
 return classId;  
 }  
  
 public void setId(String classId) {  
 this.classId = classId;  
 }  
  
 public int getHours() {  
 return hours;  
 }  
  
 public void setHours(int hours) {  
 this.hours = hours;  
 }  
  
 public char getGrade() {  
 return grade;  
 }  
  
 public void setGrade(char grade) {  
 this.grade = grade;  
 }  
}

B) public class Student {  
 private String idNum;  
 private CollegeCourse[] courses = new CollegeCourse[5];  
  
 @Override  
 public String toString() {  
 return "Student{" +  
 "idNum='" + idNum + '\'' +  
 "},";  
 }  
  
 public Student(String idNum, CollegeCourse[] courses)  
 {  
 this.idNum = idNum;  
 this.courses = courses;  
 }  
  
 public String getIdNum() {  
 return idNum;  
 }  
  
 public void setIdNum(String idNum) {  
 this.idNum = idNum;  
 }  
  
 public CollegeCourse getCourses(int num) {  
 return courses[num];  
 }  
  
 public void setCourses(CollegeCourse c, int num)  
 {  
 this.courses[num] = c;  
 }  
  
  
  
}

C) public class InputGrades {  
 public static void main(String[] args) {  
  
  
 char grade1 = 'A';  
 int hours1 = 1;  
 String cId1 = "c1";  
 char grade2 = 'B';  
 int hours2 = 2;  
 String cId2 = "c2";  
 char grade3 = 'C';  
 int hours3 = 3;  
 String cId3 = "c3";  
 char grade4 = 'D';  
 int hours4 = 4;  
 String cId4 = "c4";  
 char grade5 = 'F';  
 int hours5 = 5;  
 String cId5 = "c5";  
 String sId1 = "abc123";  
 String sId2 = "def456";  
 CollegeCourse[] classes = new CollegeCourse[5];  
 CollegeCourse[] classes2 = new CollegeCourse[5];  
  
 CollegeCourse c1 = new CollegeCourse(cId1, hours1, grade1);  
 CollegeCourse c2 = new CollegeCourse(cId2, hours2, grade2);  
 CollegeCourse c3 = new CollegeCourse(cId3, hours3, grade3);  
 CollegeCourse c4 = new CollegeCourse(cId4, hours4, grade4);  
 CollegeCourse c5 = new CollegeCourse(cId5, hours5, grade5);  
  
  
 Student s1 = new Student(sId1, classes);  
 s1.setCourses(c1,0);  
 s1.setCourses(c2,1);  
 s1.setCourses(c3,2);  
 s1.setCourses(c4,3);  
 s1.setCourses(c5,4);  
  
 Student s2 = new Student(sId2,classes2);  
 s2.setCourses(c5,0);  
 s2.setCourses(c4,1);  
 s2.setCourses(c3,2);  
 s2.setCourses(c2,3);  
 s2.setCourses(c1,4);  
  
 System.*out*.println(s1.toString() + " " + s1.getCourses(0).toString());  
 System.*out*.println(s2.toString() + " " + s2.getCourses(0).toString());  
 }  
}

