7.1 -JAVA

1. Create a simple class Shape that will represent a 2-dimensional shape with line segments for edges. It should have the following instance variables: numSides (int), regular (boolean). Create at least two constructors and getter and setter methods.

Code:

public class Shape {

private int numSides;

private boolean regular;

public Shape() {

this.numSides = 0;

this.regular = false;

}

public Shape(int numSides, boolean regular) {

this.numSides = numSides;

this.regular = regular;

}

public int getNumSides() {

return numSides;

}

public void setNumSides(int numSides) {

this.numSides = numSides;

}

public boolean isRegular() {

return regular;

}

public void setRegular(boolean regular) {

this.regular = regular;

}

@Override

public String toString() {

return "Shape with " + numSides + " sides, regular: " + regular;

}

}

2. Identify the key parts of the Java Class below. Put asterisks next to all the instance variables. Place a box around each constructor. Circle the signature of methods other than the constructor method. Place triangles around the parameters. Underline the return types of methods.

Code:

public class Animal {

\*int weight, height;

\*double speed;

[Animal() {

weight = 50;

height = 4;

speed = 2; // miles per hour

}]

[Animal(int w, int h, int s) {

weight = w;

height = h;

speed = s;

}]

⭘public double getTime(⮕double miles⮕) { // gets the number of hours to go these miles

return miles / speed;

}

⭘public int getWeight() {

return weight;

}

⭘public int getHeight() {

return height;

}

⭘public double getSpeed() {

return speed;

}

}

3. Write code to create two instances of the Animal class template listed in problem #2. Be sure to use each of the two constructors provided. Then add Java code that will print the following: a. Animal #1 has a speed of \_\_\_. b. Animal #2 has a speed of \_\_\_. Be sure that the blanks are automatically filled in with the actual speeds. Use the methods provided to access the speeds.

CODE:

public class Main {

public static void main(String[] args) {

Animal animal1 = new Animal();

Animal animal2 = new Animal(60, 5, 10);

System.out.println("Animal #1 has a speed of " + animal1.getSpeed() + ".");

System.out.println("Animal #2 has a speed of " + animal2.getSpeed() + ".");

}

}

4. Write a class Student. It should have the following instance variables for the name, credits, grade point average (GPA), and quality Points. Create a constructor method. Create two other methods as follows: a. A method that will return the current grade point average which will be the quality points divided by the credits. b. A method that will take in the credits for a class or semester along with the quality points. It should update the credits, the quality points, and the GPA.

import java.util.Scanner;

public class Student {

private String name;

private int credits;

private double gpa;

private int qualityPoints;

public Student(String name, int credits, double gpa, int qualityPoints) {

this.name = name;

this.credits = credits;

this.gpa = gpa;

this.qualityPoints = qualityPoints;

}

public double getCurrentGPA() {

return (double) qualityPoints / credits;

}

public void updateCreditsAndQualityPoints(int newCredits, int newQualityPoints) {

this.credits += newCredits;

this.qualityPoints += newQualityPoints;

this.gpa = (double) qualityPoints / credits;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter student name: ");

String name = scanner.nextLine();

System.out.print("Enter current credits: ");

int credits = scanner.nextInt();

System.out.print("Enter current quality points: ");

int qualityPoints = scanner.nextInt();

Student student = new Student(name, credits, 0.0, qualityPoints);

System.out.println("Current GPA: " + student.getCurrentGPA());

System.out.print("Enter new credits for update: ");

int newCredits = scanner.nextInt();

System.out.print("Enter new quality points for update: ");

int newQualityPoints = scanner.nextInt();

student.updateCreditsAndQualityPoints(newCredits, newQualityPoints);

System.out.println("Updated GPA: " + student.getCurrentGPA());

scanner.close();

}

}

5. Using the class you created in #4, create three instances of the Student Class from the table below:

class Student {

String name;

int credits;

int qualityPoints;

public Student(String name, int credits, int qualityPoints) {

this.name = name;

this.credits = credits;

this.qualityPoints = qualityPoints;

}

public void displayInfo() {

System.out.println("Name: " + name);

System.out.println("Credits: " + credits);

System.out.println("Quality Points: " + qualityPoints);

System.out.println();

}

}

public class Main {

public static void main(String[] args) {

Student maryJones = new Student("Mary Jones", 14, 46);

Student johnStiner = new Student("John Stiner", 60, 173);

Student ariSamala = new Student("Ari Samala", 31, 69);

maryJones.displayInfo();

johnStiner.displayInfo();

ariSamala.displayInfo();

}

}

6. Using the instance variables created in #5, add 13 credits and 52 quality points to the student “Ari Samala”

Code:

class Student {

String name;

int credits;

int qualityPoints;

public Student(String name, int credits, int qualityPoints) {

this.name = name;

this.credits = credits;

this.qualityPoints = qualityPoints;

}

public void displayInfo() {

System.out.println("Name: " + name);

System.out.println("Credits: " + credits);

System.out.println("Quality Points: " + qualityPoints);

System.out.println();

}

}

public class Main {

public static void main(String[] args) {

Student maryJones = new Student("Mary Jones", 14, 46);

Student johnStiner = new Student("John Stiner", 60, 173);

Student ariSamala = new Student("Ari Samala", 31, 69);

maryJones.displayInfo();

johnStiner.displayInfo();

ariSamala.displayInfo();

}

}

7.

Code:

public class Card {

String suit, name;

int points;

Card(int n1, int n2) {

suit = getSuit(n1);

name = getName(n2);

points = getPoints(name);

}

public String toString() {

return "The " + name + " of " + suit;

}

public String getName(int i) {

switch (i) {

case 1: return "Ace";

case 2: return "Two";

case 3: return "Three";

case 4: return "Four";

case 5: return "Five";

case 6: return "Six";

case 7: return "Seven";

case 8: return "Eight";

case 9: return "Nine";

case 10: return "Ten";

case 11: return "Jack";

case 12: return "Queen";

case 13: return "King";

default: return "error";

}

}

public int getPoints(String n) {

switch (n) {

case "Jack":

case "Queen":

case "King":

case "Ten":

return 10;

case "Two": return 2;

case "Three": return 3;

case "Four": return 4;

case "Five": return 5;

case "Six": return 6;

case "Seven": return 7;

case "Eight": return 8;

case "Nine": return 9;

case "Ace": return 1;

default: return -1;

}

}

public String getSuit(int i) {

switch (i) {

case 1: return "Diamonds";

case 2: return "Clubs";

case 3: return "Spades";

case 4: return "Hearts";

default: return "error";

}

}

public static void main(String[] args) {

Card card1 = new Card((int)(Math.random() \* 4) + 1, (int)(Math.random() \* 13) + 1);

Card card2 = new Card((int)(Math.random() \* 4) + 1, (int)(Math.random() \* 13) + 1);

System.out.println("Card 1: " + card1);

System.out.println("Card 2: " + card2);

}

}

8. Add code to the Main class in exercise #7 to the following: a. Display the total point value for the two random cards. b. Ask the user if they would like another card. If they say yes display the new card and the points for all 3 cards in their “Hand”. c. Loop to allow the user to continue to add cards to the hand until the number of points goes over 21 or the user decides not to add any more cards or the total number of cards is 5

Code:

import java.util.ArrayList;

import java.util.Scanner;

public class Card {

String suit, name;

int points;

Card(int n1, int n2) {

suit = getSuit(n1);

name = getName(n2);

points = getPoints(name);

}

public String toString() {

return "The " + name + " of " + suit + " with " + points + " points.";

}

public String getName(int i) {

switch (i) {

case 1: return "Ace";

case 2: return "Two";

case 3: return "Three";

case 4: return "Four";

case 5: return "Five";

case 6: return "Six";

case 7: return "Seven";

case 8: return "Eight";

case 9: return "Nine";

case 10: return "Ten";

case 11: return "Jack";

case 12: return "Queen";

case 13: return "King";

default: return "error";

}

}

public int getPoints(String n) {

switch (n) {

case "Jack":

case "Queen":

case "King":

case "Ten": return 10;

case "Two": return 2;

case "Three": return 3;

case "Four": return 4;

case "Five": return 5;

case "Six": return 6;

case "Seven": return 7;

case "Eight": return 8;

case "Nine": return 9;

case "Ace": return 1;

default: return -1;

}

}

public String getSuit(int i) {

switch (i) {

case 1: return "Diamonds";

case 2: return "Clubs";

case 3: return "Spades";

case 4: return "Hearts";

default: return "error";

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

ArrayList<Card> hand = new ArrayList<>();

int totalPoints = 0;

// Generate and display the first two cards

for (int i = 0; i < 2; i++) {

int suitNumber = (int) (Math.random() \* 4.0 + 1);

int faceNumber = (int) (Math.random() \* 13.0 + 1);

Card newCard = new Card(suitNumber, faceNumber);

hand.add(newCard);

totalPoints += newCard.points;

System.out.println(newCard);

}

System.out.println("Total points: " + totalPoints);

while (totalPoints <= 21 && hand.size() < 5) {

System.out.print("Would you like another card? (yes/no): ");

String response = scanner.nextLine();

if (!response.equalsIgnoreCase("yes")) {

break;

}

int suitNumber = (int) (Math.random() \* 4.0 + 1);

int faceNumber = (int) (Math.random() \* 13.0 + 1);

Card newCard = new Card(suitNumber, faceNumber);

hand.add(newCard);

totalPoints += newCard.points;

System.out.println(newCard);

System.out.println("Total points: " + totalPoints);

}

if (totalPoints > 21) {

System.out.println("You've gone over 21 points!");

} else {

System.out.println("Final hand:");

for (Card card : hand) {

System.out.println(card);

}

System.out.println("Total points: " + totalPoints);

}

scanner.close();

}

}