ICS 102 Problem Set 07

Note: In addition to these questions, you are required to solve all lab task questions on static methods, classes, static members, and arrays of references.

1. What is the output of the following program?

class Example1a {

public static void main(String[] args) {

double x = 10.0;

System.out.println("before: x = " + x);

halveIt(x);

System.out.println("after: x = " + x);

}

public static void halveIt(double y) {

y /= 2.0; // divide y by two

System.out.println("halved: y = " + y);

}

}

1. What is the output of the following program?

class Example1b {

public static void main(String[] args) {

int x = 10, y = 20;

System.out.println("before swap: x = " + x + ", y = " + y);

swap(x, y);

System.out.println("after swap: x = " + x + ", y = " + y);

}

public static void swap(int x, int y) {

int temp = x;

x = y;

y = temp;

System.out.println("inside swap: x = " + x + ", y = " + y);

}

}

1. What is the output of the following program?

public class PassString {

public static void main(String[] args){

String str1 = "Dhahran";

System.out.println(str1);

modify(str1);

System.out.println(str1);

}

public static void modify(String str2){

System.out.println(str2);

str2 = "Dammam";

System.out.println(str2);

}

}

1. What is the output of the following program?

public class PassString2 {

public static void main(String[] args){

String str1 = "Dhahran", str2 = "Riyadh";

System.out.println("Before swap: str1 = " + str1 + ", str2 = " + str2);

swap(str1, str2);

System.out.println("After swap: str1 = " + str1 + ", str2 = " + str2);

System.out.println(str1);

}

public static void swap(String str1, String str2){

String temp = str1;

str1 = str2;

str2 = temp;

System.out.println("Inside swap: str1 = " + str1 + ", str2 = " + str2);

}

}

1. What is the output of the following program?

class Person{

private String name;

private int age;

public Person(String name, int age){

this.name = name;

this.age = age;

}

public String getName(){

return name;

}

public void setName(String name){

this.name = name;

}

public String toString(){

return "[name = " + name + ", age = " + age + "]";

}

}

public class CallExample1 {

public static void main(String[] args) {

Person p = new Person("Ahmad", 15);

System.out.println("Object before: " + p);

System.out.println("Name before: " + p.getName());

modify(p);

System.out.println("Name after: " + p.getName());

System.out.println("Object after: " + p);

}

public static void modify(Person p){

p.setName("Yusuf");

p = null;

}

}

1. What is the output of the following program?

class Car{

private String name;

public void setName(String name){

this.name = name;

}

public String getName(){

return name;

}

}

public class CallExample02 {

public static void main(String[] args) {

Car a = new Car();

a.setName("Mercedes");

changer(a);

System.out.println(a.getName());

}

static void changer(Car b){

b.setName("BMW");

}

}

1. What is the output of the following program?

class Car{

private String name;

public void setName(String name){

this.name = name;

}

public String getName(){

return name;

}

}

public class CallExample02 {

public static void main(String[] args) {

Car a = new Car();

a.setName("Mercedes");

changer(a);

System.out.println(a.getName());

}

static void changer(Car b){

b = new Car();

b.setName("BMW");

}

}

1. What is the output of the following program?

public class CallTest05 {

public static void main(String[] args) {

int[] x = {1, 2, 3, 4, 5};

System.out.println("Before modification: ");

System.out.println("x[1] = " + x[1] + ", x[4] = " + x[4]);

modify(x);

System.out.println("After modification: ");

System.out.println("x[1] = " + x[1] + ", x[4] = " + x[4]);

}

public static void modify(int[] array){

array[1] = 35;

array[4] = 22;

}

}

1. What is the output of the following program?

class Dog{

private int size;

private String name ;

public String color ;

Dog(String color){

this.color = color;

}

public int getSize(){

return size;

}

public String getName(){

return name;

}

}

public class DefaultValues {

public static void main(String[] args) {

Dog dog1 = new Dog("brown");

System.out.println("Dog size is " + dog1.getSize());

System.out.println("Dog name is " + dog1.getName());

System.out.println("Dog color is " + dog1.color);

}

}

1. What is the output of the following program?

class Point{

private int x;

private int y;

public String toString(){

return "[x = " + x + " , y = " + y + "]";

}

public int getX(){

return x;

}

public int getY(){

return y;

}

public void setX(int x){

this.x = x;

}

public void setY(int y){

this.y = y;

}

}

public class NoArgumentConstructor01 {

public static void main(String[] args) {

Point p1 = new Point(); // Using the default constructor

System.out.println(p1);

p1.setX(3);

p1.setY(5);

System.out.println(p1);

}

}

1. What is the output of the following program?

class Point{

private int x;

private int y;

public Point(int x, int y){

this.x = x;

this.y = y;

}

public String toString(){

return "[x = " + x + " , y = " + y + "]";

}

public int getX(){

return x;

}

public int getY(){

return y;

}

public void setX(int x){

this.x = x;

}

public void setY(int y){

this.y = y;

}

}

public class NoArgumentConstructor02 {

public static void main(String[] args) {

Point p1 = new Point();

System.out.println(p1);

p1.setX(3);

p1.setY(5);

System.out.println(p1);

}

}

1. What is the output of the following program?

class Point{

private int x;

private int y;

public Point(){

}

public Point(int x, int y){

this.x = x;

this.y = y;

}

public String toString(){

return "[x = " + x + " , y = " + y + "]";

}

public int getX(){

return x;

}

public int getY(){

return y;

}

public void setX(int x){

this.x = x;

}

public void setY(int y){

this.y = y;

}

}

public class NoArgumentConstructor03 {

public static void main(String[] args) {

Point p1 = new Point(); // Using the No-argument constructor

System.out.println(p1);

p1.setX(3);

p1.setY(5);

System.out.println(p1);

Point p2 = new Point(12, 9);

System.out.println(p2);

}

}

1. Consider a class Rectangle with the following instance variables:

private double length;

private double width;

* Write a no-argument constructor for the class.
* Write a two-argument constructor for the class. The constructor must throw IllegalArgumentException.
* Write a copy constructor for the class.

Write the following methods for the class:

* getArea – return the area of this rectangle (the calling object).
* hasSameArea – takes another rectangle object and return true if its area is the same as this rectangle (the calling object).
* isSquare – checks if a given rectangle (as a parameter) is square or not.
* toString
* equals

1. What is the output of the following program?

class MyClass {

private static int x = 5;

public int y;

public MyClass() {}

public MyClass(int n) { y = n; }

public int getX() { return x; }

public int getY() { return y; }

public String toString() { return "(" + x + "," + y + ")"; }

public void adjustX() { x++; }

public void adjustY() { y++; }

public void adjustY(int n) { y += n; }

public static void main(String [] args){

MyClass a = new MyClass();

System.out.println(a.getX());

System.out.println(a.getY());

MyClass b = new MyClass(20);

System.out.println(b);

int n = 10;

a.adjustY(n);

b.y = a.y;

b.adjustX();

b.adjustY();

System.out.println(a + " " + b);

}

}

1. Write a *Temperature* class that has two instance variables: a temperature value (a floating-point number) and a character for the scale, either 'C' for Celsius or 'F' for Fahrenheit.

The class should have four constructors:

* one for each instance variable (assume zero degrees if no value is specified and Celsius if no scale is specified),
* one with two parameters for the two instance variables, and
* a no-argument constructor (set to zero degrees Celsius).

Include two accessor methods to return the temperature:

*getTempCelsius*: to return the degrees Celsius,

*getTempFahrenheit*: to return the degrees Fahrenheit

Note: use the following formulas :

* degreesC = 5(degreesF - 32)/9
* degreesF = (9(degreesC)/5) + 32

Include three mutator methods,

* *setValue* to set the value,
* *setScale* to set the scale ('F' or 'C'), and
* *setValueAndScale* to set both;

Include a suitable *toString* method.

Include an equals method

Then write a test class called *TestTemperature* that tests all the methods. Be sure to use each of the constructors.

1. Write a class Employee with:

* private static variable totalSalary that is used to store the total salary of all created Employee objects.
* static getTotalSalary method.
* Instance variables id, name, salary, numberOfDependents.
* A three-argument constructor. The constructor must throw IllegalArgumentException if salary and numberOfDependents are invalid i.e., < 0
* raise(double amount): method to raise salary by amount. The method must throw IllegalArgumentException if amount < 0
* addDependents(int d): method to increase number of dependent by d. The method must throw IllegalArgumentException if d < 0
* equals and toString methods. The equality of objects is only based on id.

Write an appropriate EmployeeDriver class to test the Employee class methods. Your class must create at least three Employee objects.