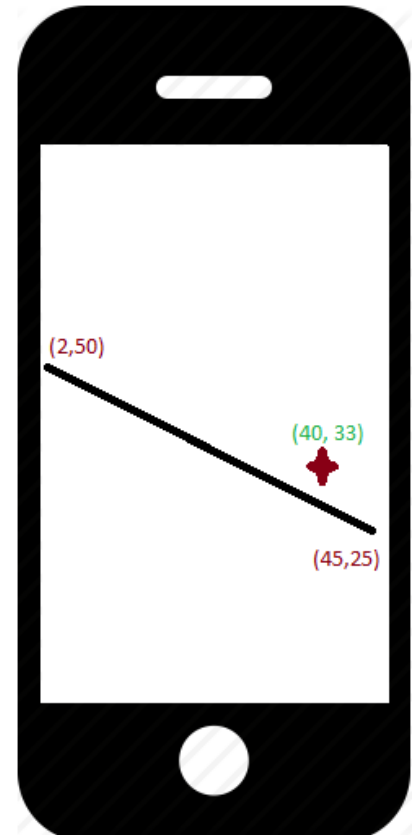


Computer Science Department
Object Oriented Programming Course
Practical Exercises

Sheet # 1

- 1- ** Write a java program to ask the user to enter his/her name and print welcome message. For example if the user enters 'Ahmed' as his name the program should print "Hi Ahmed, Welcome to JAVA".
- 2- ** In a mobile game the player should throw a boll above a certain area. Suppose that margin line coordinates are (2, 50) and (45, 25), write a program that takes the ball position and determine if the ball above the line or not. For example if the user enters (40, 33) as the ball coordinates the program should print that the ball is above the line.
- 3- *** Write a program to teach kids addition. Your program should show to the kid a simple question and wait the answer. You program should check if the answer is correct or not and print the correct answer if kid's answer is wrong. (use random number between 1 to 10)
- 4- *** Using logical operators print out the truth table for the following expressions
 - a. A and B
 - b. A or B
 - c. A Xor B
- 5- * Write a Java program to print the ASCII code table form character 'A' to character 'H'
- 6- *** Write a program that asks the user to enter three integers and prints them in ascending order.



- * Self-study problem
- ** Assignment problem
- ***Lab work problem

Sheet # 2

- 7- Take integer inputs from user until he/she presses q (Ask to press q to quit after every integer input). Print average and product of all numbers.
- 8- *** Write a java program to print the first 10 numbers in Fibonacci series. By definition, the first two numbers in the Fibonacci sequence are 0 and 1, and each subsequent number is the sum of the previous two.
- 9- * Write a java program to compute *sin* and *cos* of an angle. The program should ask the user to enter angle value in degrees and print the sin and cos of this angle. (don't use Math.sin or Math.cos)
- 10- *** Write a program that contains a function which takes two numbers x, and y and the function swap these numbers.
- 11- ** Write a program which uses your own sin and cos functions to convert from polar to Cartesian coordinates.
- 12- * Write a function that takes two integers x and y then it computes x power y.
- 13- Print the following patterns using loop :
 - a.
*
**

 - b.
*

*
 - c.
1010101
10101
101
1

- 14- ** Write a function that takes an integer L as a parameter and print a square of '*' with length L. for example if L= 4, your function should output the following shape.

```
* * * *  
* * * *  
* * * *  
* * * *
```

Sheet # 3

- 15- ** Write a java program to calculate multiplication of two matrices. The program should ask the user to enter matrix dimensions for matrix A and matrix B. The program fill matrix A and matrix B with random integers form 0 to 100. The program should calculate (A x B) and print the result to the user in matrix form.
- 16- * Write a function that takes an integer array as input and returns the same array after sorting it.
- 17- *** Write a program to show how to use dynamic array (ArrayList) of String. The program should accept to enter students' names and managing an ordered list of student names. The program should allow adding, deleting and showing students list.
- 18- * Write a program that uses the function in problem 2 to sort an array of integers before searching this array using binary search algorithm. The program should ask the user to enter the length of the array and the array elements (*dynamic array*). The user also enters an element and the program prints the index of this element or -1 if the element doesn't exist in the array.
- 19- ** Write a program to count number of ordered items. The ordered items are stored in 2-dimension array. The first column stores the item id and the second column stores the number of ordered items.
- 20- *** Write a function that takes two parameters, a sorted array of integers and an integer x. the function should return the array after inserting the integer x in the right position.

Sheet # 4

Write a java class named Student which contains the following attributes: name as a string, age as double, id as an int, level as an int, and score as a double. Also the class contains the following methods:

- 21- * Default constructor that initialize a student attributes; name "Test", age 20, id 0, level 2, and score 3.0.
- 22- ** Parameterized constructor that initialize a student attributes to a given values.
- 23- ** Print to print the student attributes.
- 24- *** Accessor and mutator methods to every attribute.
- 25- *** Write a program to create an array of ten students, and print the student with the highest score.
- 26- **Write a program to print the names of students by creating a Student class. If no name is passed while creating an object of Student class, then the name should be "Unknown", otherwise the name should be equal to the String value passed while creating object of Student class.

Sheet # 5

- 27- ** Write a Java program that takes a Boolean expression as input and prints the truth table of this expression. For example if the user enters this expression (A&B|C), the program should printout the following table

A	B	C	A&B C
T	T	T	T
F	T	T	T
T	F	T	T
F	F	T	T
T	T	F	T
F	T	F	F
T	F	F	F
F	F	F	F

- 28- *** Write a Java program to analysis an input text. The program should prints the following information:

Number of lines

Number of words

A table shows each word and its count.

For example if the input text is:

In this course, master the essentials of object-oriented programming on the Java platform, including fundamental Java syntax.

The output should be:

Number of lines = 1

Number of words = 18

Word	Count	Word	Count
In	1	This	1
course	1	master	1
the	2	essential	1
of	1	object	1
oriented	1	programming	1
on	1	Java	2
platform	1	including	1
fundamental	1	syntax	1

Sheet # 6

- 29- A boy has his money deposited \$1000, \$1500 and \$2000 in banks-Bank A, Bank B and Bank C respectively. We have to print the money deposited by him in a particular bank. Create a class 'Bank' with a method 'getBalance' which returns 0. Make its three subclasses named 'BankA', 'BankB' and 'BankC' with a method with the same name 'getBalance' which returns the amount deposited in that particular bank. Call the method 'getBalance' by the object of each of the three banks.
- 30- A class has an integer data member 'i' and a method named 'printNum' to print the value of 'i'. Its subclass also has an integer data member 'j' and a method named 'printNum' to print the value of 'j'. Make an object of the subclass and use it to assign a value to 'i' and to 'j'. Now call the method 'printNum' by this object.
- 31- Suppose a class 'A' has a static method to print "Parent". Its subclass 'B' also has a static method with the same name to print "Child". Now call this method by the objects of the two classes. Also, call this method by an object of the parent class referring to the child class i.e. A obj = new B()
- 32- All the banks operating in India are controlled by RBI. RBI has set a well defined guideline (e.g. minimum interest rate, minimum balance allowed, maximum withdrawal limit etc) which all banks must follow. For example, suppose RBI has set minimum interest rate applicable to a saving bank account to be 4% annually; however, banks are free to use 4% interest rate or to set any rates above it.

Write a JAVA program to implement bank functionality in the above scenario and demonstrate the dynamic polymorphism concept. Note: Create few classes namely Customer, Account, RBI (Base Class) and few derived classes (SBI, ICICI, PNB etc). Assume and implement required member variables and functions in each class.

Hint:

Class Customer

```
{
```

```
//Personal Details ...
```



```
// Few functions ...
}
Class Account
{
// Account Detail ...
// Few functions ...
}
Class RBI
{
Customer c; //hasA relationship
Account a; //hasA relationship
..
Public double GetInterestRate() { }
Public double GetWithdrawalLimit() { }
}
Class SBI: public RBI
{
//Use RBI functionality or define own functionality.
}
Class ICICI: public RBI
{
//Use RBI functionality or define own functionality.
}
```

Sheet # 7

33- Create a class named 'Member' having the following members:

Data members

1 - Name

2 - Age

3 - Phone number

4 - Address

5 - Salary

It also has a method named 'printSalary' which prints the salary of the members.

Two classes 'Employee' and 'Manager' inherits the 'Member' class. The 'Employee' and 'Manager' classes have data members 'specialization' and 'department' respectively. Now, assign name, age, phone number, address and salary to an employee and a manager by making an object of both of these classes and print the same.

34- Create a class named 'Shape' with a method to print "This is This is shape".

Then create two other classes named 'Rectangle', 'Circle' inheriting the Shape class, both having a method to print "This is rectangular shape" and "This is circular shape" respectively. Create a subclass 'Square' of 'Rectangle' having a method to print "Square is a rectangle". Now call the method of 'Shape' and 'Rectangle' class by the object of 'Square' class.

Sheet # 8

35- What is the difference between Checked and Unchecked Exceptions in Java?

36- What is the output of the following code snippets?

a-

```
class Example{
    public static void main(String args[]){
        int x,y,z;
        try{
            x = 5;
            y = 0;
            z = x / y;
            System.out.println("The output for z cannot be executed.")
        }
        catch(ArithmeticException e)
        {
            System.out.println("Expression throws divide by 0 exceptions");
        }
    }
}
```

b-

```
class MultiCatch{
    public static void main(String args[]){
        try{
            int a[] = new int a[];
            a[6] = 15/0;
        }
        catch(ArithmeticException e){
            System.out.println("Arithmetic Exception Handled");
        }
        catch(ArrayIndexOutOfBoundsException e1)
        {
            System.out.println("Array index out of bound!");
        }
    }
}
```

c-

```
class NestedException
{
    public static void main(String[] args) {
        try {
            int arr[]={7,0,12,8};
            try {
                int x=arr[2]/arr[1];
            }
            catch(ArithmeticException e) {
                System.out.println("Divide by Zero Exception");
            }
            arr[4]=3;
        }
    }
}
```

```

    }
    catch(ArrayIndexOutOfBoundsException e1) {
        System.out.println("Array index is out of bound");
    }
}
}

```

d-

```

class Example{
public static void main(String args[]){
    int x,y,z;
    try {
        x = 5;
        y = 0;
        z = x / y;
        System.out.println("The output for z cannot be
executed.")
    }
    catch(ArithmeticException e)
    {
        System.out.println("Expression throws divide by 0
exception");
    }
    finally
    {
        System.out.println("You are in the finally block");
    }
}
}

```

e-

```

class MyNewException extends Exception
{
    String s1;

    MyNewException(String s2) {
        s1=s2;
    }
    public String toString() {
        return ("MyNewException defines: "+s1) ;
    }
    public static void main(String args[]){
        try {
            throw new MyNewException("E - R - R - O - R ");
        }
        catch(MyNewException e1) {
            System.out.println(e1) ;
        }
    }
}

```

- 37- Create a custom exception called `MyException`. The new exception should give the message “A test Exception message: your code have a logical error”. Write a test program that throws `MyException` when the user enter a number as input for name.
- 38- Create a class named `'Rectangle'` with two data members `'length'` and `'breadth'` and two methods to print the area and perimeter of the rectangle respectively. Its constructor having parameters for length and breadth is used to initialize length and breadth of the rectangle. Let class `'Square'` inherit the `'Rectangle'` class with its constructor having a parameter for its side (suppose `s`) calling the constructor of its parent class as `'super(s,s)'`. Print the area and perimeter of a rectangle and a square.
- 39- Now repeat the above example to print the area of 10 squares.
Hint-Use array of objects

Sheet # 9

- 40- Create an abstract class `'Parent'` with a method `'message'`. It has two subclasses each having a method with the same name `'message'` that prints "This is first subclass" and "This is second subclass" respectively. Call the methods `'message'` by creating an object for each subclass.
- 41- An abstract class has a constructor which prints "This is constructor of abstract class", an abstract method named `'a_method'` and a non-abstract method which prints "This is a normal method of abstract class". A class `'SubClass'` inherits the abstract class and has a method named `'a_method'` which prints "This is abstract method". Now create an object of `'SubClass'` and call the abstract method and the non-abstract method. (Analyse the result)
- 42- Create an abstract class `'Animals'` with two abstract methods `'cats'` and `'dogs'`. Now create a class `'Cats'` with a method `'cats'` which prints "Cats meow" and a class `'Dogs'` with a method `'dogs'` which prints "Dogs bark", both inheriting the

class 'Animals'. Now create an object for each of the subclasses and call their respective methods.

43- Create an abstract class 'Bank' with an abstract method 'getBalance'. \$100, \$150 and \$200 are deposited in banks A, B and C respectively. 'BankA', 'BankB' and 'BankC' are subclasses of class 'Bank', each having a method named 'getBalance'. Call this method by creating an object of each of the three classes.

44- We have to calculate the area of a rectangle, a square and a circle. Create an abstract class 'Shape' with three abstract methods namely 'RectangleArea' taking two parameters, 'SquareArea' and 'CircleArea' taking one parameter each. The parameters of 'RectangleArea' are its length and breadth, that of 'SquareArea' is its side and that of 'CircleArea' is its radius. Now create another class 'Area' containing all the three methods 'RectangleArea', 'SquareArea' and 'CircleArea' for printing the area of rectangle, square and circle respectively. Create an object of class 'Area' and call all the three methods.

Sample Exams

Question 1 (12 pts) The government supports new companies with allowance according to the number of employees in the company. A special allowance is supported to IT companies. Solve the following points.

- Write a class named Company which has numOfEmp and name. In addition to the no argument constructor, it has a constructor that initialize the numOfEmp with a given int value. It also contains a method calcAllowance returns numOfEmp * 500.
- Write a class for IT Company that inherits from Company class. The allowance of IT companies is calculated as number of employee * 1000.
- Using polymorphism, write the main method that defines an array of companies. The array contains 2 company objects and one IT company. Calculate the total allowance for the three companies.

Question 2 What is the output of the following code snippet? (13 pts)

<pre>public class Test { public static int doStuff(double x, double y) { return (int)(x / y); } public static void main(String[] args) { float x = 6.0f; int y = 11; x = Test.doStuff(y,x); System.out.println("x="+x+", y="+y); while (x < y) { System.out.println(x); x += 4; } } }}</pre>	<pre>class GeometricObject { public String toString() { return "this is a Geometric Shape "; } } class Circle extends GeometricObject { double radius; public String toString() { return super.toString() + "\nradius is " + radius; } } public class Test{ public static void main(String[] args){ GeometricObject g = new Circle(); System.out.println(g); }}</pre>
<pre>class A{ static int value; public A(){ public A(int v){ value = v; } } } class B extends A{ public B(){ super(2); value = 4; } } public class Test{</pre>	<pre>class A { public void f(A a) { System.out.println("fa(A)"); } public void f(B b) { System.out.println("fa(B)"); } } class B extends A { public void f(A a) { System.out.println("fb(A)"); } public void f(B b) { System.out.println("fb(B)"); } } public class Test {</pre>

<pre> public static void main(String[] a){ A b = new B(); System.out.println(b.value); } } </pre>	<pre> public static void main(String[] args) { A a = new A(); B b = new B(); A ba=(A)b; // Write output next to each of the following: a.f(a); a.f(b); b.f(a); b.f(b); ba.f(a); ba.f(b); a.f(ba); b.f(ba); ba.f(ba); }} </pre>

Question 3 Detect the errors (if any) in the following code and rewrite it correctly (9 pts)

<pre> //this program should write a text on a file import java.io.File; import java.io.PrintWriter; public class Test { public Test() { } public static void main(String[] args){ File f = new File("test.txt"); f.println("Test"); } } </pre>	<pre> import java.io.File; import java.io.PrintWriter; public class Test { public Test() { } public static void main(String[] args){ try{ File f1 = new File("test.txt"); PrintWriter pw = new PrintWriter(f1); pw.println("Test"); pw.close(); } catch(Exception e){ } } } </pre>
<pre> Public interface testable{ void printStr(){ System.out.println("This is a test"); } } abstract class Test implement testable{ public abstract void void doTest() } class Test2 extends Test{ int a =4; </pre>	<pre> interface testable{ void printStr(); } abstract class Test implements testable{ public abstract void doTest(); } class Test2 extends Test{ int a =4; public void doTest(){ } } </pre>

}	public void printStr(){ } }
Rewrite the following code segment using if-else statements. Assume that grade has been declared as a type char. switch (grade){ case 'A': System.out.println("Excellent"); case 'B': System.out.println("Good"); case 'C': System.out.println("OK"); default: System.out.println("Let's talk"); }	

Question 4 write a program (16 pts)

*A company pays its employees on a weekly basis. The employees are of three types: **Salaried employees** are paid a fixed weekly salary regardless of the number of hours worked, **hourly employees** are paid by the hour and receive overtime pay for all hours worked in excess of 40 hours, and **commission employees** are paid a percentage of their. The company wants to implement a Java application that performs its payroll calculations using polymorphism. The program should take the employees data and calculates the salaries. Employee data depends on the employee type. Salaries employee has only name and salary, hourly employee has name, wage per hour, and commission employee has name sales amount and percentage. The over time for hourly employee is calculated for hours over 40 and the wage will be 1.5 wage. The commission employee salary is calculated as a percentage of his sales amount.*

Question 1 (8 pts)

- Write an abstract class named item which has price and name. In addition it has an abstract method calcNetPrcie which should calculate the item price after a given discount.
- Write biscuits class that inherits from item class in the previous question. The discount on biscuits is 7%.
- Suppose that employment law stated that the lower limit of the salaries is 1000. Create a user-defined exception to be thrown in case of salary is below the lower limit.
- Create a simple test function to use the new created exception.

Question 2 What is the output of the following code snippet? (6 pts)

```
class Person {
    public Person() {
        System.out.println("a person object is
created");
    }
    public int getSalary(int a){
        return a *2;
    }
    public String getAddress(){
        return "Assiut";
    } }
class Student extends Person{
    public Student() {
        System.out.println("a student object is
created");
    }
    public int getSalary(int a){
        return a *3;
    }
    public String getLevel(){
        return "3";
    }
}
public class Test {
    public static void main(String[] args){
        Person p = new Person();
        Person p2 = new Student();
        System.out.println(((Person)p).getAddress());
        System.out.println(((Person)p).getSalary(6));

        System.out.println(((Student)p2).getAddress()
);
```

<pre> System.out.println(((Student)p2).getSalary(6)) ; System.out.println(((Student)p2).getLevel()); } } </pre>	
<pre> public class Test { public static void main(String[] args) { A a = new A(); a.p(10); a.p(10.0); } } class B { public void p(double i) { System.out.println(i * 2); } } class A extends B { public void p(double i) { System.out.println(i); } } </pre>	<pre> public class Test { public static void main(String[] args) { int[][] m = new int [3][3]; for(int i=0; i<3; i++) for(int j=0; j<3; j++) m[i][j] = i+2*j; for(int i=0; i<3; i++){ for(int j=0; j<3; j++) System.out.print(m[i][j]+" "); System.out.println(); } } } </pre>

Question 3 write a program (11 pts)

An online trading company gives discount to its customer. It classifies the customer to three categories. The first is registered customer who make a registration on its website. The second category is golden customer who is a registered customer with total purchase amount last year is above 30000. The last category is premium customer who is a registered customer with purchased amount last year is above 100000. The company usually offers discounts to the customer in some occasions. Currently, the company offer 7% discount for registered customer. For golden customer it offers more 4% discount after registered customer discount. For premium users, the company offers 10% after registered customer discount.

Help the company to calculate the total amount of purchasing. Assume that there are 4 customers buying with these amounts 2000, 1000, 2500, 5600. Assume that the categories of customers are premium, registered, golden, and premium respectively.

The following code is trying to solve the problem

```

public class Company {
    public static void main(String[] args){
        int p1 = 2000;
        int p2 = 1000;
        int p3 = 2500;
        int p4 = 5600;
        double
                                total
                                =
calcDiscount3(p1)+calcDiscount1(p2)+calcDiscount2(p3)+calcDiscount3(p4);

```

```
        System.out.println(total);
    }
    static double calcDiscount1(double price){
        price -= price*0.07;
        return price;
    }
    static double calcDiscount2(double price){
        price -= price*0.07;
        price -= price*0.04;
        return price;
    }
    static double calcDiscount3(double price){
        price -= price*0.07;
        price -= price*0.1;
        return price;
    }
}
```

Rewrite the code in object oriented fashion and try to improve the code to be maintainable.

Question 1 (20 pts) Write java code snippets to do the following:

- Write an abstract class named item which has price and name. In addition it has an abstract method calcNetPrcie which should calculate the item price after a given discount.
- Write biscuits class that inherits from item class in the previous question. The discount on biscuits is 7%.
- Suppose that employment law stated that the lower limit of the salaries is 1000. Create a user-defined exception to be thrown in case of salary is below the lower limit.
- Create a simple test function to use the new created exception.

Question 2 What is the output of the following code snippet? (34 pts)

<pre> public class A { public static int doStuff(double x, double y) { return (int)(x / y); } public static void main() { float x = 6.0; int y = 11; x = A.doStuff(y,x); System.out.print("x="+x+", y="+y); } } </pre>	<pre> public class GeometricObject { public String toString() { return "this is a Geometric Shape "; } } public class Circle extends GeometricObject { double radius; public String toString() { return super.toString() + "\nradius is " + radius; } } public class Test{ public static void main(String args){ GeometricObject g = new Circle(); System.out.println(g); }} </pre>
<pre> public class C { public String toString() { return "c"; } public void method1() { System.out.print("c 1 "); } public void method2() { System.out.print("c 2 "); } } public class D extends B { public void method1() { </pre>	<pre> public class B extends C { public String toString() { return "b"; } public void method2() { System.out.print("b 2 "); super.method2(); } } public class A extends B { public void method2() { System.out.print("a 2 "); method1(); } } </pre>

<pre> System.out.print("d 1 "); method2(); } } </pre>	<pre> } } </pre>
<p>Given the classes above, what output is produced by the following code?</p> <pre> C[] elements = {new A(), new B(), new C(), new D()}; for (int i = 0; i < elements.length; i++) { System.out.println(elements[i]); elements[i].method1(); System.out.println(); elements[i].method2(); System.out.println(); System.out.println(); } </pre>	
<p>Choose the correct answer</p> <pre> public class A extends Exception {...} public class B extends A {...} public class C extends B {...} public void doStuff() throws A,B,C </pre> <p>The following code does not compile. Why?</p> <pre> try { doStuff(); } catch(A a) { a.printStackTrace(); } catch(B b) { b.printStackTrace(); } catch(C c) { c.printStackTrace(); } finally { System.out.println("I love exceptions!"); } </pre> <p>(a) The catch blocks for exceptions of type B and C are unreachable. ***** (b) A finally block cannot be used with multiple catch blocks. (c) B and C are not exception classes since they do not extend class Exception and therefore cannot be caught. (d) No one loves exceptions and therefore the finally block fails to compile.</p>	
<pre> int num = 1, max = 20; while (num < max) { System.out.println(num); num += 4; } </pre>	<pre> public class Test { public static void main(String[] args) { A a = new A(); a.p(10); a.p(10.0); } </pre>

	<pre> } class B { public void p(double i) { System.out.println(i * 2); } } class A extends B { public void p(double i) { System.out.println(i); } } </pre>
<pre> class Base { public void Print() { System.out.println("Base"); } } class Derived extends Base { public void Print() { System.out.println("Derived"); } } class Main{ public static void DoPrint(Base o) { o.Print(); } public static void main(String[] args) { Base x = new Base(); Base y = new Derived(); Derived z = new Derived(); DoPrint(x); DoPrint(y); DoPrint(z); } } </pre>	<pre> class A{ static int value; public A(int v){ value = v; } } public class B extends A{ public B(){ value = 4; } } public class C{ public static void main(){ B b = new B(); System.out.println(b.value); } } </pre>

Question 3 write a program (16 pts)

A trading company deals with different suppliers. Each supplier has name, address and phone. In general, each item has name, price, and one or more suppliers. There are two main items in the company: biscuits and pens. The most important property for biscuits is expire date. However, the important property for pens is manufacturer country. Design and implement a Java program to help the company to manage buying process. (Use UML to show your design) write a test class to test your program. In test program, suppose that there are two suppliers TSA, and ERA. The company is buying the item *biscoo* (biscuit) from TSA and ERA, while *CLARO* pens from ERA only. The program should print the data of items and their suppliers. Use constructors to initialize data members of classes.

- No. of pages: 2 - No. of questions: 2

answer all questions

Question 1 (9 pts)

Faculty of computers and Information (FCI) organizes some trips every year. A trip is identified by name and place. There are two types of trips: scientific trip and entertainment trip. The scientific trip has start date, end date, name, and place. The entertainment trip is a one-day trip it has date, name, and place. There is an abstract method *getPeriod* which returns number of day for a trip. This year FCI will organize 2 scientific trips and 3 entertainment trips. Write a java program to keep the list of trips this year and show how we can calculate the total number of days for all trips in this list.

What is the output of the following code snippet? (6 pts)

<pre> class Base { public void Print() { System.out.println("Base"); } } class Derived extends Base { public void Print() { System.out.println("Derived"); } } class Main{ public static void DoPrint(Base o) { o.Print(); } public static void main(String[] args) { Base x = new Base(); Base y = new Derived(); Derived z = new Derived(); DoPrint(x); DoPrint(y); DoPrint(z); } } </pre>	<p>Determine the statements that will be executed if an exception (type is <code>Exception2</code>) occurs in statment3</p> <pre> try { statement1; statement2; statement3; } catch(Exception1 ex) { statement4; } catch(Exception2 ex) { statement5; throw ex; } finally { statement6; } statement7; </pre>

<pre> public class Faculty extends Employee { public static void main(String[] args) { new Faculty(); } public Faculty() { System.out.println("(4) Faculty's no-arg constructor "); } } class Employee extends Person { public Employee() { this("(2) Invoke Employee's overloaded constructor"); System.out.println("(3) Employee's no-arg constructor "); } public Employee(String s) { System.out.println(s); } } class Person { public Person() { System.out.println("(1) Person's no-arg constructor"); } } </pre>	<pre> public class GeometricObject { public String toString() { return "this is a Geometric Shape "; } } public class Circle extends GeometricObject { double radius; public String toString() { return super.toString() + "\nradius is " + radius; } } public class Test{ public static void main(String args){ GeometricObject g = new Circle(); System.out.println(g); }} </pre>

Object Oriented Programming course

Code: CS

Practical exam (sample)

- 1) Write a java class named Employee contains (name, salary) and a method called calcTax that calculate tax according to the following table.

salary	Tax
≤ 5000	5%
$5000 < \text{salary} \leq 10000$	7%
> 10000	10%

- 2) In addition write a class named Engineer which inherits from Employee class.
The new class override the method calcTax. The new implementation adds 5% in addition to the normal employee tax.
- 3) Write a main class to test your classes.