**Java Assignment**

**Lab 2: Language Fundamentals, Classes and Objects**

2.1 Write a java program to print person details in the format as shown below:

Person Details:

\_\_\_\_\_\_\_\_\_\_\_\_

First Name: Divya

Last Name: Bharathi

Gender: F

Age: 20

Weight: 85.55

A) **package** com.capg.assignments;

**public** **class** a1{

**public** **static** **void** main(String[] args) {

System.***out***.println("Person Details:");

System.***out***.println("\_\_\_\_\_\_\_\_\_\_\_\_\_");

System.***out***.println();

System.***out***.println("First Name: Divya");

System.***out***.println("Last Name: Bharathi");

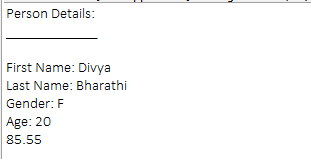
System.***out***.println("Gender: F");

System.***out***.println("Age: 20");

System.***out***.println("85.55");

}

}



2.2: Write a program to accept a number from user as a command line argument and check whether the given number is positive or negative number.

A) **package** com.capg.assignments;

**public** **class** a2{

**public** **static** **void** main(String[] args) {

**int** x=Integer.*parseInt*(args[0]);

**if**(x>0){

System.***out***.println("The number is positive");

}**else** **if**(x<0){

System.***out***.println("The number is negative");

}**else**{

System.***out***.println("Enter a valid number");

}

}

}



2.3: Refer the class diagram given below and create a person class.



Create default and parameterized constructor for Person class.

Also Create “PersonMain.java” program and write code for following operations:

1. Create an object of Person class and specify person details through constructor.
2. Display the details in the format given in Lab assignment 2.1

A) Person.java:

**package** com.capgemini.ass.java;

**public** **class** Person {

**private** String firstName;

**private** String lastName;

**private** **char** gender;

//The constructor method

**public** Person(String firstName, String lastName, **char** gender) {

**super**();

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.gender = gender;

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** **char** getGender() {

**return** gender;

}

**public** **void** setGender(**char** gender) {

**this**.gender = gender;

}

//Method to display the details of the person

**public** **void** displayPersonDetails(){

System.***out***.println("Name: "+firstName);

System.***out***.println("Surname: "+lastName);

System.***out***.println("Gender: "+gender);

}

}

PersonMain.java:

**package** com.capgemini.ass.java;

**public** **class** PersonMain {

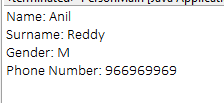
**public** **static** **void** main(String[] args) {

Person p = **new** Person("Anil","Reddy",'M');

p.displayPersonDetails();

}

}



2.4: Modify Lab assignment 2.3 to accept phone number of a person. Create a new method to implement the same and also define method for displaying person details.

Person.java:

**package** com.capgemini.ass.java;

**public** **class** Person {

**private** String firstName;

**private** String lastName;

**private** **char** gender;

**private** **int** phoneNumber;

//The constructor method

//Method to display the details of the person

**public** **void** displayPersonDetails(){

System.***out***.println("Name: "+firstName);

System.***out***.println("Surname: "+lastName);

System.***out***.println("Gender: "+gender);

System.***out***.println("Phone Number: "+phoneNumber);

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** **char** getGender() {

**return** gender;

}

**public** **void** setGender(**char** gender) {

**this**.gender = gender;

}

**public** **int** getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(**int** phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** Person(String firstName, String lastName, **char** gender,

**int** phoneNumber) {

**super**();

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.gender = gender;

**this**.phoneNumber = phoneNumber;

}

}

PersonMain.java:

**package** com.capgemini.ass.java;

**public** **class** PersonMain {

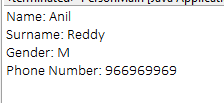
**public** **static** **void** main(String[] args) {

Person p = **new** Person("Anil","Reddy",'M',966969969);

p.displayPersonDetails();

}

}



2.5: Modify the above program, to accept only ‘M’ or ‘F’ as gender field values. Use Enumeration for implementing the same.

Person.java:

**package** com.capgemini.ass.java;

**public** **class** Person {

**private** String firstName;

**private** String lastName;

**private** Gender gender;

**private** **int** phoneNumber;

//The constructor method

//Method to display the details of the person

**public** **void** displayPersonDetails(){

System.***out***.println("Name: "+firstName);

System.***out***.println("Surname: "+lastName);

System.***out***.println("Gender: "+gender);

System.***out***.println("Phone Number: "+phoneNumber);

}

**public** String getFirstName() {

**return** firstName;

}

**public** **void** setFirstName(String firstName) {

**this**.firstName = firstName;

}

**public** String getLastName() {

**return** lastName;

}

**public** **void** setLastName(String lastName) {

**this**.lastName = lastName;

}

**public** Gender getGender() {

**return** gender;

}

**public** **void** setGender(Gender gender) {

**this**.gender = gender;

}

**public** **int** getPhoneNumber() {

**return** phoneNumber;

}

**public** **void** setPhoneNumber(**int** phoneNumber) {

**this**.phoneNumber = phoneNumber;

}

**public** Person(String firstName, String lastName, Gender gender,

**int** phoneNumber) {

**super**();

**this**.firstName = firstName;

**this**.lastName = lastName;

**this**.gender = gender;

**this**.phoneNumber = phoneNumber;

}

}

PersonMain.java:

**package** com.capgemini.ass.java;

**public** **class** PersonMain {

**public** **static** **void** main(String[] args) {

Person p = **new** Person("Anil","Reddy",Gender.***M***,966969969);

p.displayPersonDetails();

}

}

Gender.java:

**package** com.capgemini.ass.java;

**public** **enum** Gender {

***M***,***F***

}

**Lab 3: Exploring Basic Java Class Libraries**

3.1: Create a method which can perform a particular String operation based on the user’s choice. The method should accept the String object and the user’s choice and return the output of the operation.

Options are

* Add the String to itself
* Replace odd positions with #
* Remove duplicate characters in the String
* Change odd characters to upper case

**package** com.capgemini.ass.java;

**import** java.util.Scanner;

**public** **class** ThreePointOne {

**public** **static** **void** main(String[] args) {

@SuppressWarnings("resource")

Scanner input = **new** Scanner(System.***in***);

System.***out***.println("Type your name below:");

String name = input.next();

System.***out***.println(name+" "+name);

}

}

**package** com.capgemini.ass.java;

**public** **class** ThreePointOnePointOne {

**public** **static** **void** main(String[] args) {

String a = "capgeminiassignment";

**for** (**int** i=0; i < a.length(); i++){

**if** (i % 2 != 0){

a = a.substring(0,i-1) + "#" + a.substring(i, a.length());

}

}

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

}

}

**package** com.capgemini.ass.java;

**import** java.util.LinkedHashSet;

**import** java.util.Set;

**public** **class** ThreePointOnePointTwo {

**public** **static** **void** main(String[] args) {

String string = "camilacabello";

**char**[] chars = string.toCharArray();

Set<Character> charSet = **new** LinkedHashSet<Character>();

**for** (**char** c : chars) {

charSet.add(c);

}

StringBuilder sb = **new** StringBuilder();

**for** (Character character : charSet) {

sb.append(character);

}

System.***out***.println(sb.toString());

}

}

**package** com.capgemini.ass.java;

**public** **class** StringFunctionTester {

**public** **static** **void** main(String[] args) {

StringFunctions str=**new** StringFunctions();

String a="hello ";

System.***out***.println(str.add(a));

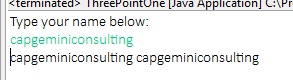
System.***out***.println(str.removeDuplicate(a));

System.***out***.println(str.replace(a));

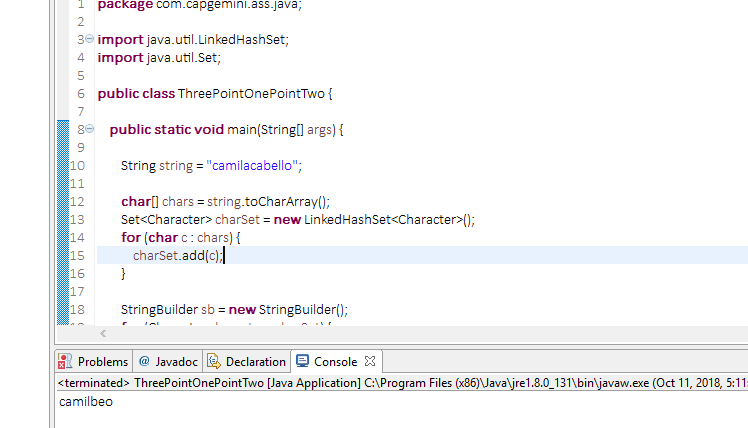
System.***out***.println(str.oddUpper(a));

}

}







3.2: Create a method that accepts a String and checks if it is a positive string. A string is considered a positive string, if on moving from left to right each character in the String comes after the previous characters in the Alphabetical order. For Example: ANT is a positive String (Since T comes after N and N comes after A). The method should return true if the entered string is positive.

A)

**StringChecker.java**

**package** com.capgemini.ass.java;

**public** **class** StringChecker {

**public** String isPositive(String string) {

**int** count=0;

**char**[] chars = string.toCharArray();

**for** (**int** i = 0; i < chars.length-1; i++) {

**if**(chars[i]<chars[i+1]){

count++;

}

}

**if**(count==chars.length-1){

**return** "Positive";

}

**return** "Negative";

}

}

**Stringtester.java**

**package** com.capgemini.ass.java;

**public** **class** StringTester {

**public** **static** **void** main(String[] args) {

StringChecker str=**new** StringChecker();

System.***out***.println(str.isPositive("ANT"));

}

}



3.3: Create a method to accept date and print the duration in days, months and years with regards to current system date.

A)

**package** com.capgemini.ass.java;

**import** java.time.LocalDate;

**import** java.time.Month;

**import** java.time.Period;

**public** **class** DateDemo1 {

**public** **void** date(LocalDate d) {

LocalDate today=LocalDate.*now*();

Period period=d.until(today);

System.***out***.println("Months:"+period.getMonths());

System.***out***.println("Days:"+period.getDays());

System.***out***.println("Years:"+period.getYears());

}

**public** **static** **void** main(String[] args) {

DateDemo1 datedemo=**new** DateDemo1();

LocalDate d=LocalDate.*of*(2014, Month.***SEPTEMBER***, 8);

datedemo.date(d);

}

}



3.4: Revise exercise 3.3 to accept two LocalDates and print the duration between dates in days, months and years.

A)

package com.capgemini.ass.java;

import java.time.LocalDate;

import java.time.Month;

import java.time.Period;

public class DateDemo {

public void date(LocalDate d1,LocalDate d2) {

Period period=d1.until(d2);

System.out.println("Months:"+period.getMonths());

System.out.println("Days:"+period.getDays());

System.out.println("Years:"+period.getYears());

}

public static void main(String[] args) {

DateDemo datedemo=new DateDemo();

LocalDate d1=LocalDate.of(2014, Month.SEPTEMBER,8);

LocalDate d2=LocalDate.of(2018, Month.MARCH, 10);

datedemo.date(d1,d2);

}

}



3.5: Create a method to accept product purchase date and warrantee period (in terms of months and years). Print the date on which warrantee of product expires.

A)

package com.capgemini.ass.java;

import java.util.Calendar;

import java.util.GregorianCalendar;

public class WarrantyChecker {

public void check(Calendar c1,int months) {

Calendar c2=new GregorianCalendar(c1.get(Calendar.YEAR),c1.get(Calendar.DATE),c1.get(Calendar.MONTH)+months);

System.out.println(c2.get(Calendar.YEAR)+","+c2.get(Calendar.MONTH));

}

public static void main(String[] args) {

WarrantyChecker wc=new WarrantyChecker();

Calendar c1=new GregorianCalendar(2018,28,Calendar.MAY);

int months=20;

wc.check(c1, months);

}

}



3.6: Create a method which accept zone id and print the current date and time with respect to given zone. (Hint: Few zones to test your code. America/New\_York, Europe/London, Asia/Tokyo, US/Pacific, Africa/Cairo, Australia/Sydney etc.)

A)

package com.capgemini.ass.java;

import java.time.ZoneId;

import java.time.ZonedDateTime;

public class ZoneTester {

public void zone(String id) {

ZonedDateTime t=ZonedDateTime.now(ZoneId.of(id));

System.out.println(t);

}

public static void main(String[] args) {

ZoneTester test=new ZoneTester();

test.zone("America/New\_York");

}

}



3.7: Modify Lab assignment 2.3 to perform following functionalities:

1. Add a method called calculateAge which should accept person’s date of birth and calculate age of a person.
2. Add a method called getFullName(String firstName, String lastName) which should return full name of a person

Display person details with age and fullname.

A)

**Persons.java**

package com.capgemini.ass.java;

import java.time.LocalDate;

import java.time.Period;

public class Persons {

private String firstName;

private String lastName;

private Gender gender;

private int phoneNumber;

private LocalDate dob;

public Persons(){

}

public Persons(String firstName, String lastName, Gender gender,

int phoneNumber, LocalDate dob) {

super();

this.firstName = firstName;

this.lastName = lastName;

this.gender = gender;

this.phoneNumber = phoneNumber;

this.dob = dob;

}

public String getFirstName() {

return firstName;

}

public void setFirstName(String firstName) {

this.firstName = firstName;

}

public String getLastName() {

return lastName;

}

public void setLastName(String lastName) {

this.lastName = lastName;

}

public Gender getGender() {

return gender;

}

public void setGender(Gender gender) {

this.gender = gender;

}

public int getPhoneNumber() {

return phoneNumber;

}

public void setPhoneNumber(int phoneNumber) {

this.phoneNumber = phoneNumber;

}

public LocalDate getDob() {

return dob;

}

public void setDob(LocalDate dob) {

this.dob = dob;

}

public String display() {

return "FirstName=" + firstName + ", LastName=" + lastName

+ ", Gender=" + gender + ", PhoneNumber=" + phoneNumber ;

}

public int age(LocalDate dob) {

LocalDate today=LocalDate.now();

Period period=dob.until(today);

return period.getYears();

}

public String getFullName(String firstName,String lastName) {

return firstName+" "+lastName;

}

}

**PersonMain.java**

package com.capgemini.ass.java;

import java.time.LocalDate;

import java.time.Month;

public class PersonMain {

public static void main(String[] args) {

LocalDate dob=LocalDate.of(1997, Month.DECEMBER, 11);

Persons person=new Persons("Raj\_Kamal","Chenumalla",Gender.M,987654321,dob);

System.out.println(person.display());

System.out.println("Age:"+person.age(dob));

System.out.println("Full Name:"+person.getFullName("Raj\_Kamal","Chenumalla"));

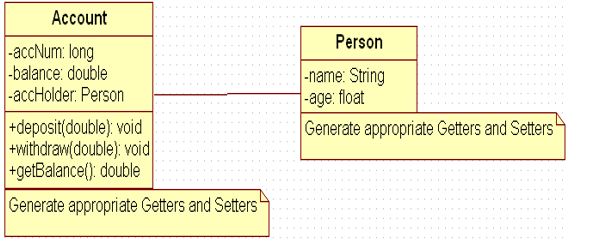
}

}



**Inheritance and Polymorphism**

4.1: Refer the case study 1in Page No: 5 and create Account Class as shown below in class diagram. Ensure minimum balance of INR 500 in a bank account is available.



1. Create Account for smith with initial balance as INR 2000 and for Kathy with initial balance as 3000. (accNum should be auto generated).
2. Deposit 2000 INR to smith account.
3. Withdraw 2000 INR from Kathy account.
4. Display updated balances in both the account.

Generate toString() method.

A) Account.java

//Account.java

**package** com.capgemini.trg.service;

**public** **class** Account **extends** PersonInheritance {

**private** Long accNum;

**private** Double balance;

**private** PersonInheritance accHolder;

**public** Account() {

**super**();

}

**public** Account(Long accNum, Double balance, PersonInheritance accHolder) {

**super**();

**this**.accNum = accNum;

**this**.balance = balance;

**this**.accHolder = accHolder;

}

**public** Long accNumAutoGeneration(){

**this**.accNum+=1000L;

**return** **this**.accNum;

}

**public** Long getAccNum() {

**return** accNum;

}

**public** **void** setAccNum(Long accNum) {

**this**.accNum = accNum;

}

**public** Double getBalance() {

**return** balance;

}

**public** **void** setBalance(Double balance) {

**this**.balance = balance;

}

**public** PersonInheritance getAccHolder() {

**return** accHolder;

}

**public** **void** setAccHolder(PersonInheritance accHolder) {

**this**.accHolder = accHolder;

}

**public** **void** deposit(Double amount){

**this**.balance=**this**.balance+amount;

}

**public** **double** withdraw(Double amount){

**if**(**this**.getBalance()<=0.0){

System.***out***.println("No balance in your Current Account");

**return** -1.0;

}**else** **if**(**this**.getBalance()<amount){

System.***out***.println("Insufficient funds in your Current Account");

**return** 0.0;

}**else** **if**((**this**.getBalance()-amount)<500.00){

System.***out***.println("Minimum balance of Rs.500 should be maintained in your current Account");

}

**this**.setBalance(**this**.getBalance()-amount);

**return** amount;

}

@Override

**public** String toString() {

**return** "Account [accNum=" + accNum + ", balance=" + balance

+ ", accHolder=" + accHolder + "]";

}

}

PersonInheritance.java

//PersonInheritance.java

**package** com.capgemini.trg.service;

**public** **class** PersonInheritance {

**private** **float** age;

**private** String name;

**public** PersonInheritance() {

**super**();

}

**public** PersonInheritance(**float** age, String name) {

**super**();

**this**.age = age;

**this**.name = name;

}

**public** **float** getAge() {

**return** age;

}

**public** **void** setAge(**float** age) {

**this**.age = age;

}

**public** String getName() {

**return** name;

}

**public** **void** setName(String name) {

**this**.name = name;

}

@Override

**public** String toString() {

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

}

}

AccountTester.java

//AccountTester.java

**package** com.capgemini.trg.ui;

**import** com.capgemini.trg.service.Account;

**import** com.capgemini.trg.service.PersonInheritance;

**public** **class** AccountTester {

**public** **static** **void** main(String args[]){

PersonInheritance person=**new** PersonInheritance(26,"Smith");

PersonInheritance person1=**new** PersonInheritance(26,"Kathy");

Account account = **new** Account((**long**)(Math.*random*()\*100000),3000.0,person);

Account account1=**new** Account((**long**)(Math.*random*()\*100000),3000.0,person1);

System.***out***.println("Account Details of smith: "+account);

System.***out***.println();

System.***out***.println("Account Details of kathy: "+account1);

System.***out***.println();

account.deposit(2000.00);

System.***out***.println("after Rs.2000.0 deposit into smith account");

System.***out***.println(account);

System.***out***.println();

System.***out***.println("after withdrawing rs.2000.0 from kathy account:");

account1.withdraw(2000.0);

System.***out***.println(account1);

}

}

