**Java Advance Assignment Questions**

**Q1. Write a java program as per the below given specification:**

**i) Design a Dish class having dishID, dishName, creationTime as the attributes. To initialize the attributes of the class and create a parameterized constructor. Also override the toString() method to print the object's details on the console.**

**ii) Design a Menu class having menuList as an attribute which is a static ArrayList. Load the menuList with a Dish object comprising some sample dishes using a static block.**

**iii) Design a Restaurant class comprising the main method. It should give a menu driven interface to a user as follows:**

**Press 1 to display Dishes.**

**Press 2 to search Dish.**

**iv) On the click of 1 all the dishes offered by the restaurant should be displayed to a user.**

**v) On the click of 2 it prompts the user to enter the dishID, if the ID entered matches with the dish, its details should be printed on the console.**

**Ans:**

// Dish class

*package* Ques1;  
  
*public class* Dish {  
 *private int* dishId;  
 *private* String dishName;  
 *private* String time;  
  
 *public* Dish(*int* dishId, String dishName, String time){  
 *this*.dishId = dishId;  
 *this*.dishName = dishName;  
 *this*.time = time;  
 }  
  
 *public int* getDishId(){  
 *return* dishId;  
 }  
  
 *public* String getDishName(){  
 *return* dishName;  
 }  
  
 *public* String getTime(){  
 *return* time;  
 }  
  
 *@Override*  
 *public* String toString() {  
 *return* "Dish{" + "Dish ID=" + dishId + ", dishName='" + dishName + '\'' + ", time='" + time + '\'' + '}';  
 }  
  
  
}

// Menu class

*package* Ques1;  
  
*import* java.util.ArrayList;  
  
*public class* Menu {  
 *static* ArrayList<Dish> MenuList = *new* ArrayList<Dish>();  
 *static* {  
 MenuList.add(*new* Dish(1,"Chikken Tikka","16:45"));  
 MenuList.add(*new* Dish(2,"Matar Paneer","19:00"));  
 MenuList.add(*new* Dish(3,"Garlic Bread","18:20"));  
 MenuList.add(*new* Dish(4,"White Sauce Pasta","21:18"));  
 }  
}

// Restaurant class

*package* Ques1;  
  
*import* java.util.Scanner;  
  
*public class* Restaurant {  
 *public static void* main(String[] args) {  
 System.out.println("Press 1 to display menu");  
 System.out.println("Press 2 to Search Dish");  
 Scanner sc = *new* Scanner(System.in);  
 *int* a = sc.nextInt();  
 System.out.println();  
 Menu menu = *new* Menu();  
 *if*( a == 1){  
 *int* i = 1;  
 *for* (Dish dish: Menu.MenuList){  
 System.out.println("Dish -->> " + i + " " + dish.getDishName());  
 i++;  
 }  
 }  
 *else if* ( a == 2 ){  
 System.out.println("Enter the Dish ID : ");  
 *int* checkid = sc.nextInt();  
 System.out.println();  
 *for* (Dish dish: Menu.MenuList){  
 *if*( checkid == dish.getDishId()){  
 System.out.println("Dish Id -->> " + dish.getDishId());  
 System.out.println("Dish Name -->> " + dish.getDishName());  
 System.out.println("Time -->> " + dish.getTime());  
 *break*;  
 }  
 }  
 }  
  
 *else*  
System.out.println("Please enter correct option");  
 }  
  
}

**Q2. Write a java program to maintain a Phone book using Map. Program should give a menu driven interface as shown below:**

**Press 1 to Add new phone book entry**

**Press 2 to Search a Phone Number**

**Press 3 to Quit.**

**On the click of 1 a user is prompted to enter his name and phone number. The details entered by the user should be maintain in a Map phonebook, where the name is the key and the phone number is the value.**

**On the click of 2 a user is prompted to enter the name of the user whose number needs to be searched. Once the name entered correctly his number is displayed on the console.**

**On the click of 3 programs terminates.**

**Use HashMap to store phone book entries.**

**Ans:**

*package* Ques2;  
  
*import* java.util.HashMap;  
*import* java.util.Scanner;  
  
*public class* Main {  
 *static private* HashMap<String ,String > phoneBook = *new* HashMap<String , String>();  
  
 *public static void* main(String[] args) {  
 Scanner sc = *new* Scanner(System.in);  
 *while* (*true*){  
 System.out.println("Press 1 to Add new phone book entry");  
 System.out.println("Press 2 to Search a Phone Number");  
 System.out.println("Press 3 to Quit.");  
 *int* flag = sc.nextInt();  
 *if*( flag == 1){  
 System.out.println("Enter Name : ");  
 String userName = sc.next();  
 System.out.println("Enter phone number : ");  
 String phoneNum = sc.next();  
 *if*(!phoneBook.containsKey(userName)){  
 phoneBook.put(userName, phoneNum);  
 }  
 *else*  
System.out.println("Number already exist");  
 }  
 *else if*(flag == 2)  
 {  
 System.out.println("Enter Name in Phone Book to search : ");  
 String userName = sc.next();  
 *if*(phoneBook.containsKey(userName))  
 {  
 System.out.println("Name : " + userName + "Phone Number : " + phoneBook.get(userName));  
 }  
 }  
  
 *else if*( flag == 3)  
 *break*;  
 }  
 }  
}

**Q5.Write a program WordCount that reads a file and reports how many lines, words, and characters appear in it. Suppose, for example, that the file lear.txt contains the following passage from Shakespeare’s King Lear:**

**Poor naked wretches, wheresoe'er you are,**

**That bide the pelting of this pitiless storm,**

**How shall your houseless heads and unfed sides,**

**Your loop'd and window'd raggedness, defend you**

**From seasons such as these? O, I have ta'en**

**Too little care of this!**

**Given this file, your program should be able to generate the following sample run:**

**File: lear.txt**

**Lines = 6**

**Words = 47**

**Chars = 248**

**Ans:**

*package* Ques5;  
  
*import* java.io.\*;  
  
*public class* Main {  
 *public static void* main(String[] args) *throws* IOException {  
 File file = *new* File("/home/knoldus/Desktop/JavaAdvance-Assignment/src/Ques5/lear.txt");  
 BufferedReader br = *new* BufferedReader(*new* FileReader(file));  
 *int* words = 0, lines = 0, characters = 0;  
 String str;  
 *while* ((str = br.readLine()) != *null*) {  
 characters = characters + str.length();  
 String word[] = str.split("\\s");  
 words = words + word.length;  
 lines++;  
 }  
 System.out.println("Total Lines = " + lines + "\n" + "Total words = " + words + "\n" + "Total characters = " + characters);  
 }  
}

**Q6. Write a program that reads a list of exam scores from the file MidtermScores.txt (which contains one score per line) and then displays a histogram of those numbers, divided into the ranges 0–9, 10–19, 20–29, and so forth, up to the range containing only the value 100.**

**If, for example, MidtermScores.txt contains the data shown in the right margin, your program should then be able to generate a histogram that looks as much as possible like the following sample run:**

**00-09 :**

**10-19 : \***

**20-29 : \***

**30-39 : \*\***

**40-49 : \***

**50-59 : \*\*\*\*\***

**60-69 : \*\*\*\*\*\*\***

**70-79 : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**80-89 : \*\*\*\*\*\***

**90-99 : \*\*\*\*\*\*\*\*\*\***

**100 : \***

**Ans:**

*package* Ques6;  
  
*import* java.io.\*;  
  
*public class* Main {  
 *public static void* main(String[] args) *throws* IOException {  
 File file = *new* File("/home/knoldus/Desktop/JavaAdvance-Assignment/src/Ques6/MidTermScores.txt");  
 BufferedReader br = *new* BufferedReader(*new* FileReader(file));  
 String str;  
 *int* range0to9 = 0, range10to19 = 0, range20to29 = 0, range30to39 = 0, range40to49 = 0, range50to59 = 0,  
 range60to69 = 0, range70to79 = 0, range80to89 = 0, range90to99 =0, range100 = 0;  
 *while* ((str = br.readLine()) != *null*){  
 *int* y = Integer.parseInt(str);  
 *if* ( y>=0 && y<=9 )  
 range0to9++;  
 *else if*(y>=10 && y<=19)  
 range10to19++;  
 *else if*(y>=20 && y<=29)  
 range20to29++;  
 *else if*(y>=30 && y<=39)  
 range30to39++;  
 *else if*(y>=40 && y<=49)  
 range40to49++;  
 *else if*(y>=50 && y<=59)  
 range50to59++;  
 *else if*(y>=60 && y<=69)  
 range60to69++;  
 *else if*(y>=70 && y<=79)  
 range70to79++;  
 *else if*(y>=80 && y<=89)  
 range80to89++;  
 *else if*(y>=90 && y<=99)  
 range90to99++;  
 *else if*(y==100)  
 range100++;  
 }  
 System.out.print("00 to 09 : ");  
 *for*(*int* i=0;i<range0to9;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("10 to 19 : ");  
 *for*(*int* i=0;i<range10to19;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("20 to 29 : ");  
 *for*(*int* i=0;i<range20to29;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("30 to 39 : ");  
 *for*(*int* i=0;i<range30to39;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("40 to 49 : ");  
 *for*(*int* i=0;i<range40to49;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("50 to 59 : ");  
 *for*(*int* i=0;i<range50to59;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("60 to 69 : ");  
 *for*(*int* i=0;i<range60to69;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("70 to 79 : ");  
 *for*(*int* i=0;i<range70to79;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("80 to 89 : ");  
 *for*(*int* i=0;i<range80to89;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("90 to 99 : ");  
 *for*(*int* i=0;i<range90to99;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 System.out.print("100 : ");  
 *for*(*int* i=0;i<range100;i++)  
 System.out.print("\*");  
 System.out.println();  
  
 }  
}

**Q7. Write and explain the output of the following program?**

public class JavaHungry {

public static void main(String args[])

{

try

{

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

for (int i = 0; i <= 5; i++)

{

System.out.print ("Array elements are : " + arr[i] + "\n");

}

}

catch (Exception e)

{

System.out.println ("Exception : " + e);

}

catch (ArrayIndexOutOfBoundsException ex)

{

System.out.println ("ArrayIndexOutOfBoundsException : "+ ex);

}

}

}.

**Output:** The above program will not run because it will throw an exception “ArrayIndexOutOfBoundException”. There should be multiple catch block catching an exception of a parent class before than a sub – class.

**Q8. WAP to handle the exception using try and multiple catch blocks.**

**Ans:**

*package* Ques8;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 *try*{  
 String s=*null*;  
 System.out.println(s.length());  
 }  
 *catch*(ArithmeticException e) {  
 System.out.println("Arithmetic Exception occurs");  
 }  
 *catch*(ArrayIndexOutOfBoundsException e) {  
 System.out.println("ArrayIndexOutOfBounds Exception occurs");  
 }  
 *catch*(Exception e) {  
 System.out.println("Parent Exception occurs");  
 }  
 }  
}

**Q9. Create a class TwoDim which contains private members as x and y coordinates in package P1. Define the default constructor, a parameterized constructor and override toString() method to display the coordinates. Now reuse this class and in package P2 create another class ThreeDim, adding a new dimension as z as its private member. Define the constructors for the subclass and override toString() method in the subclass also. Write appropriate methods to show dynamic method dispatch. The main() function should be in a package P.**

**Ans:**

// TwoDim class

*package* Ques9.P1;  
  
*public class* TwoDim {  
 *private double* x;  
 *private double* y;  
  
 *public* TwoDim(*double* x, *double* y){  
 *this*.x = x;  
 *this*.y = y;  
 }  
  
 *@Override*  
 *public* String toString(){  
 *return* "TwoDim {" + "x = " + x + "y = " + y + '}';  
 }  
}

// ThreeDim class

*package* Ques9.P2;  
  
*import* Ques9.P1.TwoDim;  
  
*public class* ThreeDim *extends* TwoDim {  
 *private double* z;  
  
 *public* ThreeDim(*double* x, *double* y, *double* z){  
 *super*(x,y);  
 *this*.z = z;  
 }  
  
 *@Override*  
 *public* String toString(){  
 *return* "ThreeDim {" + "z = " + z + '}';  
 }  
}

// main method

*package* Ques9.P;  
*import* Ques9.P1.TwoDim;  
*import* Ques9.P2.ThreeDim;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 TwoDim obj=*new* TwoDim(2.3,4.5);  
 System.out.println(obj.toString());  
 System.out.println("\n\n");  
  
  
  
 System.out.println("Dynamic method dispatch");  
 obj=*new* ThreeDim(2.3,4.5,5.6);  
 System.out.println(obj.toString());  
  
 }  
}

Q10. Is the below code written correctly? If not then give reason.

class A

{

String s = "AAA";

void methodA()

{

System.out.println(s);

}

static class B

{

void methodB()

{

methodA();

}

}

}.

**Ans:** The above code will not run because the code is written incorrectly.

1. static inner classes can only access static members of the outer class.
2. non-static reference cannot be referenced from a static context.

**Q11. Write the output of the following code.**

class Outer {

void outerMethod() {

System.out.println("Inside outerMethod");

// Inner class is local to outerMethod()

class Inner {

void innerMethod() {

System.out.println("Inside innerMethod");

}

}

Inner y = new Inner();

y.innerMethod();

}

}

class Test1 {

public static void main(String[] args) {

Outer x = new Outer();

x.outerMethod();

}

}.

**OUTPUT**

Inside outerMethod

Inside innerMethod

**Q12. Design an Employee class having attributes as empID, empName & empAge. The sample data of Employees is stored in a text file Employee.txt where the details are separated by “,”. Load an array of Employee using static block which comprises the Employee object created from the sample data stored in Employee.txt file. Allow the user to modify the Employee details in the array and save the updated information back to the file Employee.txt.**

**Ans:**

// Employee class

*package* Ques12;  
  
*public class* Employee {  
 *private int* EmpID;  
 *private* String EmpName;  
 *private int* age;  
  
 *public* Employee(*int* EmpID, String EmpName, *int* age){  
 *this*.EmpID = EmpID;  
 *this*.EmpName = EmpName;  
 *this*.age = age;  
 }  
  
 *public int* getEmpID() {  
 *return* EmpID;  
 }  
  
 *public* String getEmpName() {  
 *return* EmpName;  
 }  
  
 *public int* getAge() {  
 *return* age;  
 }  
  
 *@Override*  
 *public* String toString(){  
 *return* "Employee{" +  
 "empId=" + EmpID +  
 ", empName='" + EmpName + '\'' +  
 ", age=" + age +  
 '}';  
  
 }  
}

// main method

*package* Ques12;  
  
*import* java.io.\*;  
*import* java.util.Scanner;  
*import* java.util.logging.FileHandler;  
  
*public class* Main {  
 *static* Employee array[];  
  
 *static* {  
 array = *new* Employee[1];  
 }  
  
 *public static void* main(String[] args) *throws* IOException {  
 File file = *new* File("/home/knoldus/Desktop/JavaAdvance-Assignment/src/Ques12/Employee.txt");  
  
 *// Reading from the file*  
  
BufferedReader br = *new* BufferedReader(*new* FileReader(file));  
 String str;  
 System.out.println("Employee Details Currently inside Employee.txt: ");  
 *while* ((str = br.readLine()) != *null*) {  
 String[] vartemp = str.trim().split("\\,");  
  
 *// Storing inside employee object*  
  
Employee emp = *new* Employee(Integer.parseInt(vartemp[0]),vartemp[1],Integer.parseInt(vartemp[2]));  
 array[0] = emp;  
 }  
 br.close();  
  
 *// Displaying the content of object*  
  
System.out.println(array[0]);  
  
 *// Writing new content to file*  
  
Scanner sc = *new* Scanner(System.in);  
 System.out.println("Enter the detail of another employee ");  
 System.out.println("Enter the Name : ");  
 String name = sc.nextLine();  
 System.out.println("Enter the ID : ");  
 *int* id = sc.nextInt();  
 System.out.println("Enter the age : ");  
 *int* age = sc.nextInt();  
 String text = id + "," + name + "," + age ;  
 FileWriter fw = *new* FileWriter(file);  
 fw.write(text);  
 fw.flush();  
 fw.close();  
  
 *// Displaying the new content of Employee*  
  
br = *new* BufferedReader(*new* FileReader(file));  
 System.out.println("New employee details are : ");  
 *while* ((str = br.readLine()) != *null*){  
 System.out.println(str);  
 }  
 }  
}

**Q13. Write a Java program that implements a multi-thread application that has three threads. First thread generates a random integer every 1 second and if the value is even, the second thread computes the square of the number and prints. If the value is odd, the third thread will print the value of the cube of the number.**

**Ans:**

// Square class

*package* Ques13;  
  
*public class* Square *implements* Runnable{  
 *int* x;  
  
 *public* Square(*int* x){  
 *this*.x = x;  
 }  
 *@Override*  
 *public void* run() {  
 System.out.println("Square of the number = " + x\*x);  
  
 }  
}

// Cube class

*package* Ques13;  
  
*public class* Cube *implements* Runnable{  
  
 *int* x;  
  
 *public* Cube(*int* x){  
 *this*.x = x;  
 }  
 *@Override*  
 *public void* run() {  
 System.out.println("Cube of the number = " + x\*x\*x);  
 }  
}

// main method

*package* Ques13;  
  
*import* java.util.Random;  
*import* java.util.concurrent.Executors;  
*import* java.util.concurrent.ScheduledExecutorService;  
*import* java.util.concurrent.TimeUnit;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 ScheduledExecutorService exec = Executors.newSingleThreadScheduledExecutor();  
 exec.scheduleAtFixedRate(*new* Runnable() {  
 *@Override*  
 *public void* run() {  
 Random random = *new* Random();  
 *int* num = random.nextInt(1000);  
 System.out.println("Random Integer = "+num);  
 *if*( num % 2 == 0) {  
 Square obj=*new* Square(num);  
 obj.run();  
 }  
 *else* {  
 Cube obj=*new* Cube(num);  
 obj.run();  
 }  
 }  
 }, 0, 1, TimeUnit.SECONDS);  
 }  
  
}

**Q14. Write a program having user interface like**

**4. accept first name and surname**

**5. display total name**

**6. exit**

**Option A should accept First Name and SurName from command prompt and save that to Vector object**

**Option B it has to display how many names entered in the vector object**

**This menu should be repeated till users select exit.**

**To store first name and surname, create a class Name with these two attributes.**

**Ans:**

// InputName class

*package* Ques14;  
  
*public class* InputName {  
 *private* String firstName;  
 *private* String lastName;  
  
 *public* InputName(String firstName, String lastName){  
 *this*.firstName = firstName;  
 *this*.lastName = lastName;  
 }  
  
 *public* String getFirstName(){  
 *return* firstName;  
 }  
  
 *public* String getLastName(){  
 *return* lastName;  
 }  
  
 *@Override*  
  
 *public* String toString(){  
 *return* "Name{" +  
 "firstName='" + firstName + '\'' +  
 ", lastName='" + lastName + '\'' +  
 '}';  
  
 }  
}

// main method

*package* Ques14;  
  
*import* java.util.List;  
*import* java.util.Scanner;  
*import* java.util.Vector;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 Scanner sc = *new* Scanner(System.in);  
 List<InputName> name = *new* Vector<InputName>();  
 *while* (*true*){  
 System.out.println("To check Press one of the following options");  
 System.out.println("4. accept first name and surname");  
 System.out.println("5. display total name");  
 System.out.println("6. exit");  
 System.out.println("Enter the option : ");  
 *int* choice = sc.nextInt();  
 *if* (choice == 4) {  
 System.out.print("Enter First Name : ");  
 String firstName=sc.nextLine();  
 System.out.print("Enter LastName : ");  
 String lastName=sc.nextLine();  
 InputName tempName=*new* InputName(firstName,lastName);  
 name.add(tempName);  
  
 }  
 *else if* ( choice == 5){  
 *if* (name.size() == 0 ){  
 System.out.println("No names are available.");  
 }  
 *else*{  
 System.out.println("Total name are : ");  
 *for* (InputName name1: name){  
 System.out.println(name1);  
 }  
 }  
 }  
 *else if* ( choice == 6 )  
 *break*;  
 *else*  
System.out.println("Enter correct option." + "\n");  
 }  
 }  
}

**Q15. Write a Java program to get the last modified time of a file.**

**Ans:**

*package* Ques15;  
  
*package* Ques15;  
  
*import* java.io.File;  
*import* java.util.Date;  
  
*public class* Main {  
 *public static void* main(String[] args) {  
 File file = *new* File("/home/knoldus/Desktop/JavaAdvance-Assignment/src/Ques15/LastModifiedTime.txt");  
 Date date = *new* Date(file.lastModified());  
 System.out.println("\nThe file was last modified on: "+date+"\n");  
 }  
}