1. Create an abstract class **Employee** which contains *name*, *paymentPerHour* as attribute and abstract *calculateSalary()* method. Let the sub classes extend this **Employee** class and implement its *calculateSalary()* method. Create **Contractor** and **FullTimeEmployee** classes (variable *workingHours* should be in both the classes) as we know that salary structure for a contractor and full time employee is different so let these classes override and implement a *calculateSalary()* method. Write complete Java code to implement this scenario.

package employee;  
  
public abstract class EmployeeDetails {  
 private String employeeName;  
 private double paymentPerHour;  
  
 public EmployeeDetails(String employeeName, double paymentPerHour) {  
 this.employeeName = employeeName;  
 this.paymentPerHour = paymentPerHour;  
 }  
  
 public double getPaymentPerHour() {  
 return paymentPerHour;  
 }  
  
 public abstract void calculateSalary();  
  
}

package employee;  
  
public class Contractor extends EmployeeDetails{  
 private double workingHours;  
  
 public double getWorkingHours() {  
 return workingHours;  
 }  
 public void setWorkingHours(double workingHours) {  
 this.workingHours = workingHours;  
 }  
 public Contractor(String employeeName, double paymentPerHour) {  
 super(employeeName, paymentPerHour);  
 }  
 @Override  
 public void calculateSalary() {  
 double salary=getWorkingHours()\*getPaymentPerHour();  
 System.*out*.println("Contractor salary = "+salary);  
 }  
}

package employee;  
  
public class FullTimeEmployee extends EmployeeDetails {  
 private double employeeworkingHours;  
  
 public double getEmployeeworkingHours() {  
 return employeeworkingHours;  
 }  
 public void setEmployeeworkingHours(double employeeworkingHours) {  
 this.employeeworkingHours = employeeworkingHours;  
 }  
 public FullTimeEmployee(String employeeName, double paymentPerHour) {  
 super(employeeName, paymentPerHour);  
 }  
 @Override  
 public void calculateSalary() {  
 double salary=getEmployeeworkingHours()\*super.getPaymentPerHour();  
 System.*out*.println("Salary of Full Time Employee = "+salary);  
 }  
}

1. Consider we have two interfaces, **MotorBike** and **Cycle**. **MotorBike** interface consists of the method *speed()*. **Cycle** interface consists of the method *distance()*. Also consider one parent class **Car** having method *display()*. Write a child class **TwoWheeler**, inside this class take method *speed()* to print speed as 90Km/Hr of **MotorBike**, Similarly take *distance()* method to print distance as 50 meter of **Cycle** and *display()* method to display name of **Car**. Write complete Java code for this scenario.

package Interface;  
  
public interface MotorBike {  
 public void speed();  
}

package Interface;  
  
public interface Cycle {  
 public void distance();  
}

package Interface;  
  
public class Car implements MotorBike,Cycle {  
 public void display(){  
 System.*out*.println("Car name BMW.");  
  
 }  
  
 @Override  
 public void distance() {  
 System.*out*.println("Distance as 50 meter ");  
  
 }  
  
 @Override  
 public void speed() {  
 System.*out*.println("Speed as a 90km/Hrs.");  
  
 }  
}

package Interface;  
  
public class TwoWheeler extends Car{  
 @Override  
 public void speed() {  
 super.speed();  
 }  
  
 @Override  
 public void distance() {  
 super.distance();  
 }  
  
 @Override  
 public void display() {  
 super.display();  
 }  
}

1. Write Java program to cause “*NumberFormatException*”, “*StringIndexOutOfBoundsException*”, after causing, solve these exceptions.

package NumberFormateAndStringIndexOutOfBound;  
  
public class StringClass {  
 private static String *string* = "abcd";  
 private static String *string\_1* = "bhagyshri";  
 private static int *number*;  
  
 public static void display() {  
 *number* = Integer.*parseInt*(*string*);  
 }  
 public static void display\_1() {  
 *string\_1*.charAt(20);  
 }  
}

1. Write a Java program to throw an user defined exception “*StringMismatch*” if two string are not equal.

package StringMismatch;  
  
public class StringMIsmatchClass {  
 private String str1;  
 private String str2;  
  
 public StringMIsmatchClass(String str1, String str2) {  
 this.str1 = str1;  
 this.str2 = str2;  
 }  
  
 public String getStr1() {  
 return str1;  
 }  
 public String getStr2() {  
 return str2;  
 }  
 public String displayStringResult() throws StringMismatchException{  
 if(!str1.equals(str2))  
 throw new StringMismatchException();  
 return "String are match";  
 }  
}

package StringMismatch;  
  
public class StringMismatchException extends Exception{  
 public String getMessage(){  
 return "String not match.";  
 }  
}

1. Main class for all the Packages:

import Interface.TwoWheeler;  
import NumberFormateAndStringIndexOutOfBound.StringClass;  
import StringMismatch.StringMIsmatchClass;  
import StringMismatch.StringMismatchException;  
import employee.Contractor;  
import employee.FullTimeEmployee;  
  
import java.util.Scanner;  
  
public class Main {  
 public static void main(String[] args) {  
 Scanner scanner=new Scanner(System.*in*);  
   
// ================================ StringMismatch ==============================================  
   
 System.*out*.println("Enter two string :");  
 StringMIsmatchClass stringMIsmatchClass=new StringMIsmatchClass(scanner.next(), scanner.next());  
 try {  
 System.*out*.println(stringMIsmatchClass.displayStringResult());  
 }catch (StringMismatchException exception){  
 System.*out*.println(exception);  
 }  
// ================================ Interface ====================================================  
   
 TwoWheeler twoWheeler=new TwoWheeler();  
 twoWheeler.speed();  
 twoWheeler.distance();  
 twoWheeler.display();  
   
// ================================ Employee ========================================================  
  
 System.*out*.println("Enter Employee name and salary :");  
 FullTimeEmployee fullTimeEmployee=new FullTimeEmployee(scanner.next(), scanner.nextDouble());  
 System.*out*.println("Enter working hours for employee :");  
 fullTimeEmployee.setEmployeeworkingHours(scanner.nextDouble());  
 fullTimeEmployee.calculateSalary();  
  
 System.*out*.println("Enter Contractor name and salary :");  
 Contractor contractor=new Contractor(scanner.next(), scanner.nextDouble());  
 System.*out*.println("Enter working hours for conterctor :");  
 contractor.setWorkingHours(scanner.nextDouble());  
 contractor.calculateSalary();  
  
// ================================ Number Formate Exception ================================================  
 try {  
 StringClass.*display*();  
 }  
 catch (NumberFormatException numberFormatException){  
 System.*out*.println(numberFormatException);  
 }  
 try{  
 StringClass.*display\_1*();  
 }  
 catch (StringIndexOutOfBoundsException stringIndexOutOfBoundsException){  
 System.*out*.println(stringIndexOutOfBoundsException);  
 }  
 }  
}

1. How can we use user defined package in Java? Explain with proper example. Also give one example of built-in package.

**Answer:-**

**Package:-**

* It is a group of similar type of classes, subclasses, interface and subpackages.
* It is use to organise the file in a synchronized manner.
* When we create any project, it contains huge number of files, it is very difficult to understand which file is of which type, to overcome this situation we can use packages.
* We can create package by following steps:

**Project->src->New ->package->package name.**

* If we create an execution class within the package there is no need to import the packge in the execution class.
* But whenever the execution class is outside the package, the method declaration class needs to be public and import the method declaration class in that execution class.
* Otherwise it will display an error “Please create local variable.”
* **Example 1: In same package**

Package student;

Class StudentDetails{

Static Void display(){

System.out.println(“Data display.”);

}

}

Class MainForStudent{

StudentDetails.display;

}

**Example 2: In different packages**

Package student;

Class StudentDetails{

Static Void display(){

System.out.println(“Data display.”);

}

}

Import student.StudentDetails;

Class MainForStudent

{

StudentDetails.display;

}