**Final Exam Instructions**

**OBJECT-ORIENTED PROG**

* This is a take-home exam. You can use any resources that are available for you to finish this exam, except
  + Outsourcing the exam to any person or to any third party websites
  + Copying from other students work
  + Copying direct quotes from the books or internet
* Do not lose your opportunity to learn while working on the exam. Understand the concept and write answers on your own.
* Usually, in life, we have several choices. Unfortunately, you don’t have any choice on this exam. You have to answer all the questions and each part of the problem.
* All the topics on this exam were discussed in class . So, you cannot claim that the questions are out of the syllabus!
* Refer to Microsoft Word tutorials for proper formatting
* Points will be deducted for grammatical and spelling mistakes
* No two brains think alike unless you are soulmates. Definitely your answers will not be same as other students.
* Read the code of academic integrity before you start the exam. <https://www.nwmissouri.edu/policies/academics/Academic-Integrity.pdf>
* Push your source code to GitHub and provide your GitHub link at the end of the document and in the comment section.
* Don’t use examples that already explained in class or worksheets.
* Provide the input and output screenshots for every program.

**Final Exam OBJECT-ORIENTED PROG 01FA20 150 pts**

1. (20-Points) Define the terms abstract classes and interfaces. What are the similarities and differences between abstract classes and interfaces? Why interfaces are preferred over abstract classes? Explain and demonstrate with examples.

Answer:

Abstract Classes : A class which has a abstract keyword in its declaration. Abstract classes should have atleast one abstract method without a body. Inheriting classes should implement the abstract methods.

Interfaces : A class which has a interface keyword in its declaration. It is defined as the abstract type used to specify the behavior of the class which contains static constants and abstract methods. A class can implement multiple interfaces.

Similarities of Abstract classes and Interfaces :

* Objects cannot be created.
* Abstract classes and Interfaces both contains static and final variables and provide static method implementation.
* They are inherited from common domain of itself using extends keyword.

Difference of Abstract classes and Interfaces :

* Abstract classes cannot be instantiated but can be invoked where as interfaces cannot be instantiated or invoked.
* Abstract classes does not support multiple inheritance where as interfaces support multiple inheritance.
* To avoid independence we use abstract classes for future implementation we use interfaces.
* Abstract classes have access modifier where as interfaces does not have access modifier everything defined in interface are specified as public.

Why interfaces are preferred over abstract classes:

An abstract class allows to make functionality that subclasses can implement or override whereas an interface only permits to state functionality but not to implement. An interface helps to accomplish abstraction and polymorphism, which are main principles.

Abstract class Example 1:

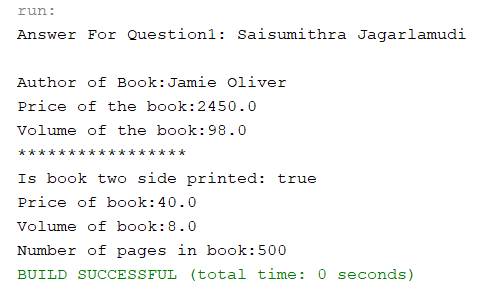
Explanation:

Here I have created abstract class for Books. Here I have given abstract method as getvolume and getprice of bag based on height, width, length, costpercubicunit.Where Books extends ChildrenBooks and CookingBooks. In ChildrenBooks I have given methods to find the volume and price of book. Where CookingBook extends Books here also we declare method volume and price of book where I have given fixed value for volume. VegetarianBooks extends CookingBooks.

In driver class I have created an objects for ChildrenBooks and CookingBooks classes. With objects created I have displayed the price, volume, authorname, istwosidesprinted, number of pages in the book.

|  |
| --- |
| **Books Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public abstract class Books {  private double length;  private double width;  private double height;  private String authorname;  private double costperunit;  public Books(double length, double width, double height, String authorname, double costperunit) {  this.length = length;  this.width = width;  this.height = height;  this.authorname = authorname;  this.costperunit = costperunit;  }    public abstract double getprice();  public abstract double getvolume();    public double getLength() {  return length;  }  public double getWidth() {  return width;  }  public double getHeight() {  return height;  }  public String getAuthorname() {  return authorname;  }  public double getCostperunit() {  return costperunit;  }  }  **ChildrenBooks Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class ChildrenBooks extends Books{  private int numberofpages;  private boolean istwosidesprinted;  public ChildrenBooks(int numberofpages, boolean istwosidesprinted, double length, double width, double height, String authorname, double costperunit) {  super(length, width, height, authorname, costperunit);  this.numberofpages = numberofpages;  this.istwosidesprinted = istwosidesprinted;  }  public int getNumberofpages() {  return numberofpages;  }  public boolean isIstwosidesprinted() {  return istwosidesprinted;  }    public double getvolume()  {  return getLength()\*getWidth()\*getHeight();  }    public double getprice()  {  return getvolume()\*getCostperunit();  }  }  **CookingBooks Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class CookingBooks extends Books {  public CookingBooks(double length, double width, double height, String authorname, double costperunit) {  super(length, width, height, authorname, costperunit);  }  public double getvolume() {  return 98;  }  @Override  public double getprice() {  return getvolume()\*getCostperunit() ;  }  }  **VegetarianBooks Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class VegetarianBooks extends CookingBooks {  public VegetarianBooks(double length, double width, double height, String authorname, double costperunit) {  super(length, width, height, authorname, costperunit);  }    }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic  System.out.println("Answer For Question1: Saisumithra Jagarlamudi\n");  VegetarianBooks sh = new VegetarianBooks(2,3,4,"Jamie Oliver",25);  sh.getprice();  sh.getAuthorname();  sh.getvolume();    ChildrenBooks hs = new ChildrenBooks(500,true,4,1,2,"Ronald Dahl",5);  hs.isIstwosidesprinted();  hs.getNumberofpages();  hs.getvolume();  hs.getprice();    System.out.println("Author of Book:"+sh.getAuthorname());  System.out.println("Price of the book:"+sh.getprice());  System.out.println("Volume of the book:"+sh.getvolume());  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("Is book two side printed: "+hs.isIstwosidesprinted());  System.out.println("Price of book:"+hs.getprice());  System.out.println( "Volume of book:"+hs.getvolume());  System.out.println("Number of pages in book:"+hs.getNumberofpages());  }  } |

Output 1 :



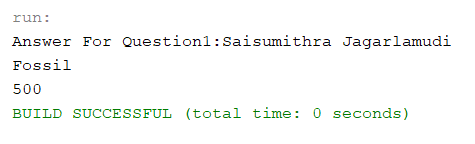
Abstarct Class Example 2:

Explanation:

I have created a class Watches and given methods to find brandname and cost of watch. And other class WristWatch that extends Watches. And a driver class to display the cost and name of the watch.

|  |
| --- |
| **Watches Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public abstract class Watches {  public abstract void brandname();  public abstract void costofwatch();  }  **WristWatch Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class WristWatch extends Watches {    public void brandname()  {  System.out.println("Fossil");  }  public void costofwatch()  {  System.out.println("500");  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question1:Saisumithra Jagarlamudi");  Watches w = new WristWatch();  w.brandname();  w.costofwatch();  }  } |

Output 2:



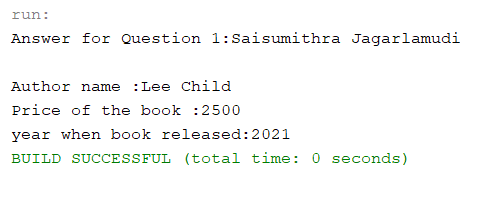
Interface Example 1:

Explanation:

I have created a Books interface class. It specifies the name of author and price of the book. And created a method to find the authorname, price of the book and release year. Where Books implement ActionBooks and ThrillerBooks. In driver created an object to find authorname, price of the book and year when the released.

|  |
| --- |
| **Books Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public interface Books {  public void authorname();  public void priceofbook();    }  **ActionBooks Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class ActionBooks implements Books {  private static final String authorname = "Jack Reacher";  private static final String priceofbook = "5000";  private static final String releaseyear = "2021";      public void authorname()  {  System.out.println("Name of author :"+authorname);  }    public void priceofbook()  {  System.out.println("Price of book:"+priceofbook);  }  public void releaseyear()  {  System.out.println("year when book released:"+releaseyear);    }  }  **ThrillerBooks Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class ThrillerBooks {  private static final String authorname = "Lee Child";  private static final String priceofbook = "2500";    public void authorname()  {  System.out.println("Author name :"+authorname);  }  public void priceofbook()  {  System.out.println("Price of the book :"+priceofbook);  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer for Question 1:Saisumithra Jagarlamudi\n");  ActionBooks hs = new ActionBooks();  ThrillerBooks sh = new ThrillerBooks();  sh.authorname();  sh.priceofbook();  hs.releaseyear();  }    } |

Output 1:



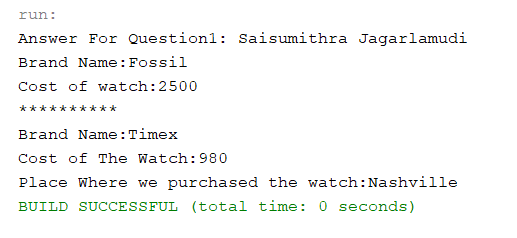
Interface Example 2:

Explanation:

I have created a class Watches and given methods to find brandname and cost of watch. And other class WristWatch and SmartWatch that implements Watches. And a driver class to display the cost and brand name of the watch and the place where we bought the watch.

|  |
| --- |
| **Watch Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public interface Watches {  public void brandname();  public void costofwatch();  }  **SmartWatch Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class SmartWatch implements Watches {  private static final String brandname="Timex";  private static final String costofwatch="980";  private static final String address="Nashville";      public void brandname()  {  System.out.println("Brand Name:"+brandname);  }  @Override  public void costofwatch() {  System.out.println("Cost of The Watch:"+costofwatch);  }  public void address()  {  System.out.println("Place Where we purchased the watch:"+address);    }  }  **WristWatch Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class WristWatch implements Watches{    private static final String brandname="Fossil";  private static final String costofwatch="2500";      public void brandname()  {  System.out.println("Brand Name:"+brandname);  }  @Override  public void costofwatch() {  System.out.println("Cost of watch:"+costofwatch);    }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_1\_2\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question1: Saisumithra Jagarlamudi");  WristWatch hs = new WristWatch();  SmartWatch sh = new SmartWatch();  hs.brandname();  hs.costofwatch();  System.out.println("\*\*\*\*\*\*\*\*\*\*");  sh.brandname();  sh.costofwatch();  sh.address();  }    } |

Output 2:



1. (10-Points) Design an interface named Colorable with a void method named howToColor(). Every class of a colorable object must implement the Colorable interface. Design a class named Square that extends GeometricObject and implements Colorable Implement howToColor to display the message Color all four sides.

Draw a UML diagram that involves Colorable, Square, and GeometricObject. Write a test program that creates an array of five GeometricObjects. For each object in the array, display its area and invoke its howToColor method if it is colorable.

Answer:

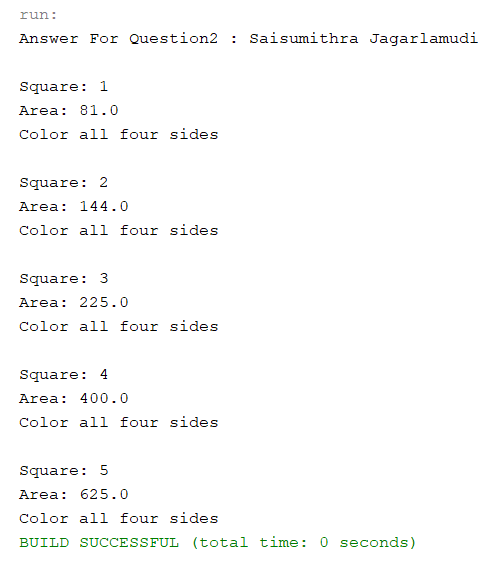
UML Diagram :

Diagram

Description automatically generated

|  |
| --- |
| **GeometricObject Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam02;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public abstract class GeometricObject {    public abstract double getArea();  }  **Colorable Interface**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam02;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public interface Colorable  {  public void howToColor();  }  **Square Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam02;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Square extends GeometricObject implements Colorable {  private double side;    public Square(){  }  public Square(double side) {  this.side = side;  }  public double getSide() {  return side;  }  @Override  public double getArea() {  return side \* side;  }  @Override  public void howToColor() {  System.out.println( "Color all four sides");  }  @Override  public String toString() {  return super.toString() + "\nSide: " + side + "\nArea: " + getArea();    }  }  **DriverClass**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam02;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question2 : Saisumithra Jagarlamudi");  GeometricObject[] hs = {new Square(9), new Square(12), new Square(15), new Square(20), new Square(25)};  for (int i = 0; i < hs.length; i++)  {  System.out.println("\nSquare: " + (i + 1));  System.out.println("Area: " + hs[i].getArea());  if(hs[i] instanceof Colorable){  ((Colorable)hs[i]).howToColor();  }  }  }  } |

Output:



1. (10-Points) What is casting? What are different types of casting? Explain and demonstrate with examples.

Answer:

Casting : It is a Method of changing an entity from one datatype to another datatype.

There are two types of Casting:

Widening Casting : This type of casting takes place when two datatypes are automatically converted. This involves the conversation of a smaller datatype to larger datatype.

Narrowing Casting : This type of casting involves the conversation of larger datatypes to smaller datatypes.

Widening Casting

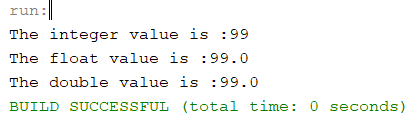
Example 1:

Explanation:

In this example I have changed datatype from int to float and double. In widening casting datatypes are converted from smaller datatypes to larger datatypes.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  int number = 99;  System.out.println("The integer value is :" +number);  float result = number;  System.out.println("The float value is :" +result);  double value = number;  System.out.println("The double value is :" +value);  }    } |

Output:

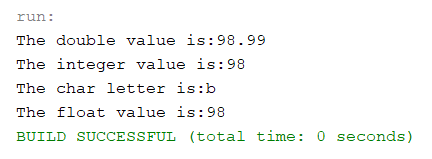


Narrowing Casting:

In this example we are changing datatype double larger datatype to int, char, short datatypes.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  double number = 98.99;  System.out.println("The double value is:" +number);  int value = (int)number;  System.out.println("The integer value is:" +value);  char output = (char)number;  System.out.println("The char letter is:"+output);  short result = (short)number;  System.out.println("The float value is:" +result);    }    } |

Output :



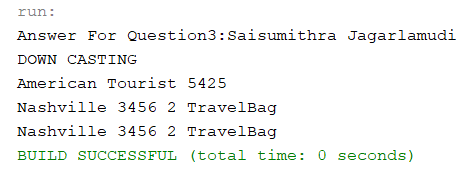
DownCasting Example:

Explanation:

I have used Bags class that extends travelbags and backpack bags. Where in bags class we have given methods to find brandname and price and getbagdetails. In travel bags we specified the number of wheels and in backpack about the address. In driver class I have created an object and using casting and polymorphism methods I displayed the name, address, number of wheel and brand of the bag.

|  |
| --- |
| **Bags Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_3;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Bags {  private String brandname;  private int price;  public Bags(String brandname, int price) {  this.brandname = brandname;  this.price = price;  }  public String getBrandname() {  return brandname;  }  public int getPrice() {  return price;  }    public String getBagDetails()  {  return getBrandname()+ " " + getPrice();  }  @Override  public String toString() {  return "Bags{" + "brandname=" + brandname + ", price=" + price + '}';  }  }  **TravelBags Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_3;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class TravelBags extends Bags{  private int wheels;  public TravelBags(int wheels, String brandname, int price) {  super(brandname, price);  this.wheels = wheels;  }  public int getWheels() {  return wheels;  }  public String getBagDetails(){  return super.getBagDetails()+" " +getWheels()+" " +"TravelBag";  }  @Override  public String toString() {  return super.toString()+ "TravelBags{" + "wheels=" + wheels + '}';  }  }  **BackPack Bags Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_3;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class BackPack extends Bags{    private String address;  public BackPack(String address, String brandname, int price) {  super(brandname, price);  this.address = address;  }  public String getAddress() {  return address;  }  public String getBagDetails(){  return getAddress()+"."+super.getBagDetails()+"BackPack";  }    @Override  public String toString() {  return super.toString()+"BackPack{" + "address=" + address + '}';  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_3;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Drive {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question3:Saisumithra Jagarlamudi");  System.out.println("DOWN CASTING");  Bags hs=new Bags("American Tourist",5425);  System.out.println(hs.getBagDetails());  TravelBags sh =new TravelBags( 2,"Nashville",3456);  System.out.println(sh.getBagDetails());  hs=sh;  sh=(TravelBags)hs;//Casting  System.out.println(sh.getBagDetails());  }  } |

Output:



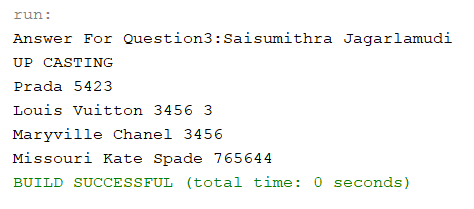
UpCasting Example

Explanation:

I have created a bag class that extends travelbags and backpack bags. Where in bags class we have given methods to find brandname and price and getbagdetails. In travel bags we specified the number of wheels and in backpack about the address. In driver class I have created an object to display the name, address, number of wheel and brand of the bag.

|  |
| --- |
| **Bags Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_4;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Bags {  private String brandname;  private int price;  public Bags(String brandname, int price) {  this.brandname = brandname;  this.price = price;  }  public String getBrandname() {  return brandname;  }  public int getPrice() {  return price;  }    public String getBagDetails()  {  return getBrandname()+ " " + getPrice();  }  @Override  public String toString() {  return "Bags{" + "brandname=" + brandname + ", price=" + price + '}';  }  }  **TravelBag Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_4;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class TravelBags extends Bags{  private int wheels;  public TravelBags(int wheels, String brandname, int price) {  super(brandname, price);  this.wheels = wheels;  }  public int getWheels() {  return wheels;  }  public String getBagDetails(){  return super.getBagDetails()+" " +getWheels();  }  @Override  public String toString() {  return super.toString()+ "TravelBags{" + "wheels=" + wheels + '}';  }  }  **BackPack Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_4;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class BackPack extends Bags{  private String address;  public BackPack(String address, String brandname, int price) {  super(brandname, price);  this.address = address;  }  public String getAddress() {  return address;  }  public String getBagDetails(){  return getAddress()+" "+super.getBagDetails();  }    @Override  public String toString() {  return super.toString()+"BackPack{" + "address=" + address + '}';  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_3\_4;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question3:Saisumithra Jagarlamudi");  System.out.println("UP CASTING");  Bags b = new Bags("Prada", 5423);  System.out.println(b.getBagDetails());  Bags b1 = new TravelBags(3,"Louis Vuitton",3456);  System.out.println(b1.getBagDetails());  BackPack bp = new BackPack("Maryville","Chanel",3456);  System.out.println(bp.getBagDetails());  Bags b2 = new BackPack("Missouri","Kate Spade",765644);  System.out.println(b2.getBagDetails());  }  } |

Output:



1. (15-Points) Suppose that Fruit, Apple, Orange, GoldenDelicious, and McIntosh are defined in the following inheritance hierarchy:

Fruit

Orange

Apple

GoldenDelicious

McIntosh

Assume that the following code is given:

Fruit fruit = new GoldenDelicious();

Orange orange = new Orange();

Answer the following questions and explain why these Statements are legal or illegal.

1. Is fruit instance of Fruit?

|  |
| --- |
| Answer:  This statement is legal, we have Fruit as the parent class for all the classes and a object is assigned to parent object which is instance of parent class that is Fruit. |

1. Is fruit instance of Orange?

|  |
| --- |
| Answer:  This statement is illegal, there is no relation for Orange and GoldenDelicious. The parent class of GoldenDelicious is Apple for which Fruit is the main class. |

1. Is fruit instance of Apple?

|  |
| --- |
| Answer:  This statement is legal, fruit is an object of GoldenDelicious class and Apple is a parent class for GoldenDelicious |

1. Is fruit instance of GoldenDelicious?

|  |
| --- |
| Answer:  This statement is legal, fruit is object for GoldenDelicious class and an object to itself. |

1. Is fruit instance of McIntosh?

|  |
| --- |
| Answer:  This statement is illegal, fruit is object for GoldenDelicious and is subclass of Fruit Class. MCIntosh is subclass of Apple and there is no relation for McIntosh and GoldenDelicious classes. |

1. Is orange instance of Orange?

|  |
| --- |
| Answer:  This statement is legal, as orange is object for Orange class. |

1. Is orange instance of Fruit?

|  |
| --- |
| Answer:  This statement is legal, orange class is child class of Fruit class. So Orange class object is instance of parent class which is Fruit. |

1. Is orange instance of Apple?

|  |
| --- |
| Answer:  This statement is illegal, Apple and Orange classes are subclass for Fruit and no relation for Orange and Apple class. |

1. Suppose the method makeAppleCider is defined in the Apple class. Can fruit invoke this method? Can orange invoke this method?

|  |
| --- |
| Answer:  Apple class has a method makeAppleCider. Fruit class cannot invoke makeAppleCider because it is present class of Apple class. Orange class cannot invoke makeAppleCider because Orange class is not the child class of Apple. |

1. Suppose the method makeOrangeJuice is defined in the Orange class. Can orange invoke this method? Can fruit invoke this method?

|  |
| --- |
| Answer:  Orange class has a method makeOrangeJuice. Orange class can invoke makeOrangeJuice because method is present in the same class. Fruit class cannot invoke makeOrangeJuice method because parent class cannot access the methods of child class. |

1. Is the statement Orange p = new Apple() legal?

|  |
| --- |
| Answer:  This statement is illegal, Orange class and Apple class are different sub classes under the parent class Fruit. |

1. Is the statement McIntosh p = new Apple() legal?

|  |
| --- |
| Answer:  This statement is illegal, we cannot create an instance of parent class with child class reference. The parent class is Apple subclass is McIntosh. |

1. Is the statement Apple p = new McIntosh() legal?

|  |
| --- |
| Answer:  This statement is legal, we can create an instance child class using the reference of parent class. The parent class is Apple while subclass is McIntosh. |

1. (10-Points) Define a class named ComparableCircle that extends Circle and implements Comparable. Draw the UML diagram and implement the compareTo method to compare the circles on the basis of area. Write a test class to find the larger of two instances of ComparableCircle objects.

Answer :

UML Diagram :

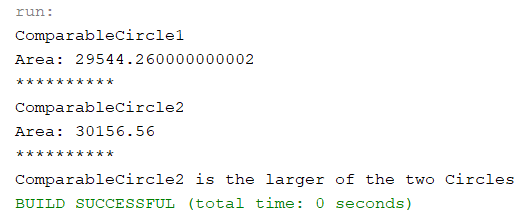
Diagram

Description automatically generated

Program :

|  |
| --- |
| **Circle Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_5;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Circle {  private double radius;  public Circle(double radius) {  this.radius = radius;  }  public double getRadius() {  return radius;  }    public double getArea() {  return radius \* radius \* 3.14;  }  }  **ComparableCircle Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_5;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class ComparableCircle extends Circle implements Comparable<ComparableCircle>  {  public ComparableCircle(double radius) {  super(radius);  }  @Override  public int compareTo(ComparableCircle o) {  if (getArea() > o.getArea())  return 1;  else if (getArea() < o.getArea())  return -1;  else  return 0;  }  @Override  public String toString() {  return "Area: " +getArea();  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_5;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  ComparableCircle comparableCircle1 = new ComparableCircle(97);  ComparableCircle comparableCircle2 = new ComparableCircle(98);  System.out.println("ComparableCircle1");  System.out.println(comparableCircle1);  System.out.println("\*\*\*\*\*\*\*\*\*\*");  System.out.println("ComparableCircle2");  System.out.println(comparableCircle2);  System.out.println("\*\*\*\*\*\*\*\*\*\*");  System.out.println((comparableCircle1.compareTo(comparableCircle2) == 1 ? "ComparableCircle1 " : "ComparableCircle2 ") +  "is the larger of the two Circles");  }  } |

Output :



1. (15-Points) What is an exception? What are checked and unchecked exceptions? Explain and demonstrate with examples.

Answer:

Exception : An exception is an unwanted event which occurs during the execution of the program.

Checked Exception : Checked Exceptions are those that are checked at compile-time.

Examples of Checked Exceptions are Class not found exceptions, IO exception, SQL exception.

Unchecked Exception : Unchecked Exceptions are those that are checked at run time.

Examples of Unchecked Exceptions are Arithmetic Exceptions, Null Pointer Exception.

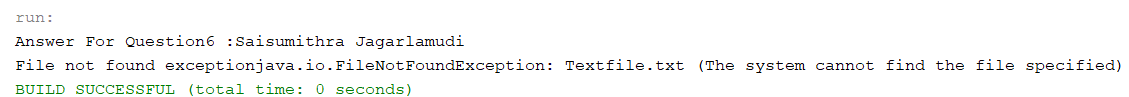
Examples :

Explanation:

In this example I have created a mainmethod to print the exception error message inside try block I have created a object fileReader and placed inputfile. And in catch I printed the error message.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_6\_1;  import java.io.FileNotFoundException;  import java.io.FileReader;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) throws FileNotFoundException {  // TODO code application logic here  System.out.println("Answer For Question6 :Saisumithra Jagarlamudi");  try {  FileReader file1 = new FileReader("Textfile.txt");  FileReader file2;  } catch (FileNotFoundException ex) {  System.out.println("File not found exception" + ex);  }  }  } |

Output:

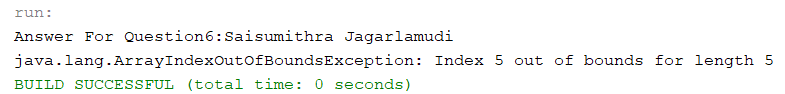


Explanation:

In this example here I have used the array of elements and in print statement I called an index value and throws IOException and here I have used ArrayIndexOutOfBounds Exception.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_6\_1\_2;  import java.io.IOException;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) throws IOException{  // TODO code application logic here  System.out.println("Answer For Question6:Saisumithra Jagarlamudi");  try {  int arr[] = { 34, 67, 8, 5, 2 };  System.out.println(arr[5]);  } catch (ArrayIndexOutOfBoundsException e) {  System.out.println(e);  }  }      } |

Output:

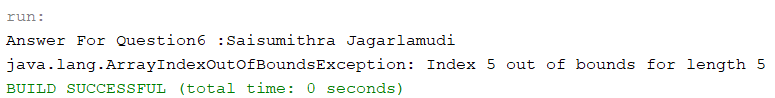


Explanation:

In this example here I have used the array of elements and in print statement I called an index value and here I have used ArrayIndexOutOfBounds Exception.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_6\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question6 :Saisumithra Jagarlamudi");  try {  int arr[] = { 34, 67, 8, 5, 2 };  System.out.println(arr[5]);  } catch (ArrayIndexOutOfBoundsException e) {  System.out.println(e);  }  }  } |

Output:

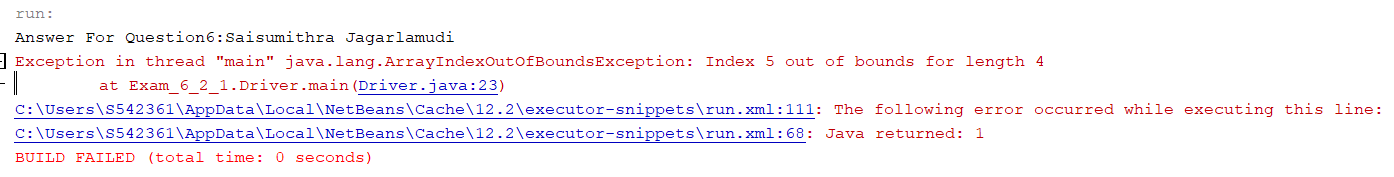


Explanation :

In this example I used array and called an array value with exception and throws IOException.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_6\_2\_1;  import java.io.IOException;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args)throws IOException {  // TODO code application logic here  System.out.println("Answer For Question6:Saisumithra Jagarlamudi");  int[] array = {17, 34, 56, 57};  System.out.println(array[5]);  }    } |

Output:



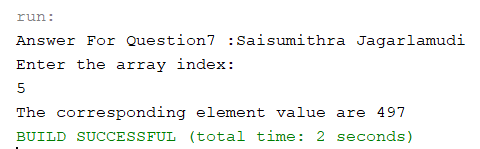
1. (10-Points) Write a program that meets the following requirements:

* Creates an array with 100 randomly chosen integers.
* Prompts the user to enter the index of the array, then displays the corresponding element value. If the specified index is out of bounds, display the message Out of Bounds.

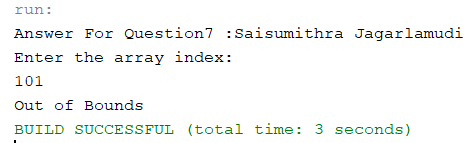
Answer:

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_07;    import java.util.\*;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question7 :Saisumithra Jagarlamudi");  int[] arr = new int[100];  for (int i = 0; i < arr.length; i++) {  arr[i] = (int) (Math.random() \* 1000) + 1;  }  Scanner input = new Scanner(System.in);  System.out.println("Enter the array index:");  try {  System.out.println("The corresponding element value are " + arr[input.nextInt()]);  } catch (ArrayIndexOutOfBoundsException ex) {  System.out.println("Out of Bounds");  }  }  } |

Output 1 :



Output 2 :



1. (10-Points) What is the purpose of declaring exceptions? How do you declare an exception, and where? Can you declare multiple exceptions in a method header? Explain and demonstrate with examples.

Answer:

The purpose of declaring exceptions is to separate error handling code from regular code.

The method must declare by using throws keyword.

To declare an exception use throws keyword in method header.

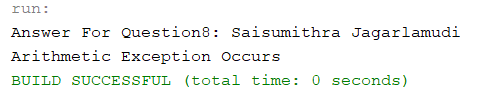
Yes, we can declare multiple exceptions in a method header are possible. If the method is declared with multiple exceptions, add exceptions list separated by commas after throws.

Explanation:

In this example I have used ArithmeticException and ArrayIndexOutofBoundsException by using the array list.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_8;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args)throws ArithmeticException, ArrayIndexOutOfBoundsException {  // TODO code application logic here  System.out.println("Answer For Question8: Saisumithra Jagarlamudi");  try {  int a[] = new int[10];  a[3]=6/0;  }catch(ArithmeticException e)  {  System.out.println("Arithmetic Exception Occurs");  }catch(ArrayIndexOutOfBoundsException e)  {  System.out.println("Array Index out of Bounds Exception");  }  }  } |

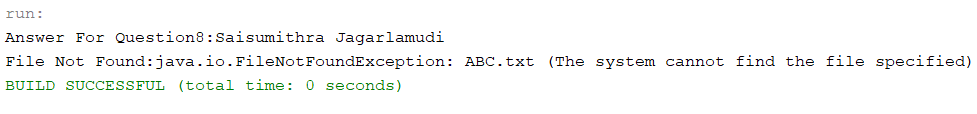
Output:



Explanation: In this example I have used Filenotfoundexception, IOException, ArrayIndexoutofBoundsException where I have used the textfile.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_8\_1;  import java.io.FileNotFoundException;  import java.io.FileReader;  import java.io.IOException;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args)throws FileNotFoundException, ArrayIndexOutOfBoundsException, IOException {  // TODO code application logic here  System.out.println("Answer For Question8:Saisumithra Jagarlamudi");  FileReader file;  try {  file = new FileReader("ABC.txt");  file.read();  }  catch(ArrayIndexOutOfBoundsException | FileNotFoundException e)  {  System.out.println("File Not Found:"+e);  }  }  } |

Output:



1. (10-Points) What is the keyword throw used for? Can you throw multiple exceptions in one throw statement? Explain with examples.

Answer:

throw : The keyword throw is used to throw an exception within a method. When a throw statement is executed, execution of current method breaks and returned to the caller.

throws : The keyword throws is used in the method signature to declare exception which might get thrown by the function while executing the code.

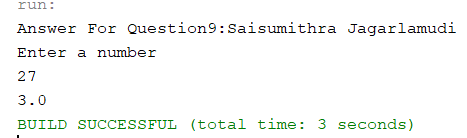
We can throw multiple exceptions. If we throw more than one exception in our code we can choose if we want to use a separate try block for each statement that could throw an exception or we should use only one try block for multiple statements that might throw multiple exceptions.

Explanation:

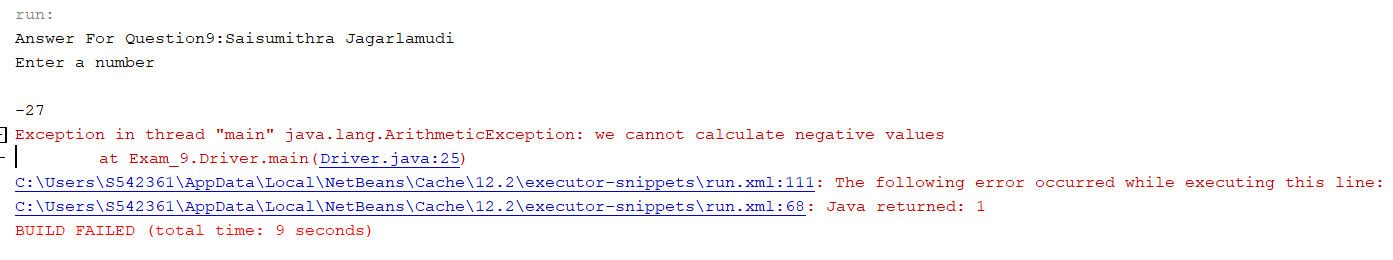
I have used Scanner to read the value of a number and displayed the cubicroot of a number.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_9;  import java.util.Scanner;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  Scanner scan = new Scanner(System.in);  System.out.println("Answer For Question9:Saisumithra Jagarlamudi");  System.out.println("Enter a number");  int value =scan.nextInt();  if(value<=0)  {  throw new ArithmeticException("we cannot calculate negative values");  }  System.out.println(Math.cbrt(value));  }  } |

Output:



Output 2 :

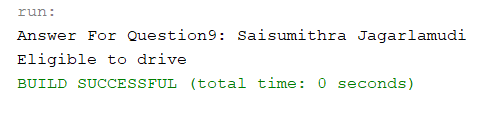


Explanation:

In this example I have given the method drivepermission where based on the age is eligible or not eligible will be calculated. If not exception raises.

|  |
| --- |
| **Example Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_9\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Example {  public void getDrivePermission(int age) throws ArithmeticException{  if(age<18)  {  throw new ArithmeticException("Not eligible to drive");  }else{  System.out.println("Eligible to drive");  }  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_9\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question9: Saisumithra Jagarlamudi");  try{  Example e = new Example();  e.getDrivePermission(25);  }catch(ArithmeticException e){  System.out.println(e);    }  }    } |

Output:



1. (15-Points) What is a recursive method? What is an infinite recursion? Explain and demonstrate with examples. Implement the search (element) in a list using recursion.

Answer:

Recursive Method : It is a technique to make a function call itself. A method that repeats steps by using one or more loops with different arguments. It is used to solve problems that can break into smaller repetitive problems.

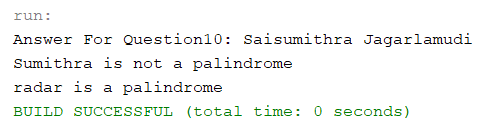
Infinite Recursion : It is a non-terminating execution of block of code. It is caused by an error in the program. If a recursion never reaches a base case it will go on making recursive calls forever and the program will never terminate.

Explanation:

In this program I have used palindrome example based on length it checks the first letter and last letter if first letter is equal to last letter it returns true else false. And the function again calls itself.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_10;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  String string = "Sumithra";  String string2 = "radar";  System.out.println("Answer For Question10: Saisumithra Jagarlamudi");  if (isPalindrome(string))  System.out.println(string + " is a palindrome");  else  System.out.println(string + " is not a palindrome");    if (isPalindrome(string2))  System.out.println(string2 + " is a palindrome");  else  System.out.println(string2 + " is not a palindrome");  }  public static boolean isPalindrome(String s)  {  if (s.length() == 0 || s.length() == 1)  return true;  if (s.charAt(0) == s.charAt(s.length() - 1))  return isPalindrome(s.substring(1, s.length() - 1));  return false;  }  } |

Output:

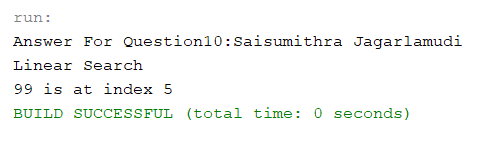


Explanation:

In this example we have given the linear search where it checks the array, match the key element with the array element if key element found return the index position of the array element. If key not found return -1.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_10\_2;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  static int linearSearch(int a[], int start, int end, int key) {  if (start > end) {  return -1;  } else if (a[start] == key) {  return start;  } else if (a[end] == key) {  return end;  }  return linearSearch(a, start+1, end-1, key);  }  public static void main(String[] args) {  int[] a = {25, 65, 67, 76, 43, 99};  int index = linearSearch(a, 0, a.length-1,99);  System.out.println("Answer For Question10:Saisumithra Jagarlamudi");  System.out.println("Linear Search");  if (index != -1) {  System.out.println(99 + " is at index " + index);  }  else {  System.out.println(99 + " is not present");  }  }  } |

Output:

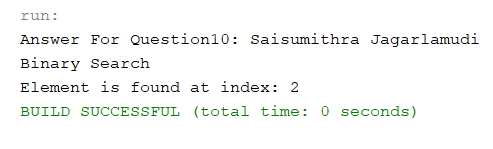


Explanation :

In this example it checks the mid value with the index. We have given the key element from the index if the value is present it returns its index value otherwise it returns index not found.

|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_10\_1;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  public static int binarySearch(int arr[], int a, int b, int key){  if (b>=a){  int mid = a + (b - a)/2;  if (arr[mid] == key){  return mid;  }  if (arr[mid] > key){  return binarySearch(arr, a, mid-1, key);  }else{  return binarySearch(arr, mid+1, b, key);  }  }  return -1;  }  public static void main(String args[]){  int arr[] = {1,4,5,8,9};  int key = 5;  int last=arr.length-1;  int result = binarySearch(arr,0,last,key);  System.out.println("Answer For Question10: Saisumithra Jagarlamudi");  System.out.println("Binary Search");  if (result == -1)  System.out.println("Element is not found");  else  System.out.println("Element is found at index: "+result);  }  } |

Output:



1. (10-Points) Write a java program that illustrates how equals() and hashCode() methods work? Explain your code in comments.

hashCode() : It is an integer value that is associated with each other. It is used to facilitate hashing in the hash tables, which are used by data structures like HashMap.

equals() : equals() method compare the two objects return true if both the strings are equal and returns false if they are not equal.

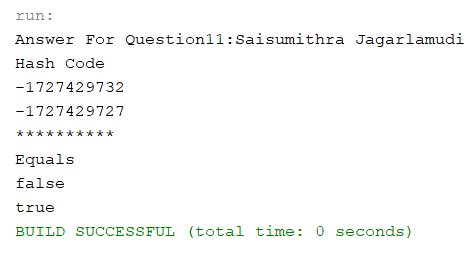
Explanation:

In this example I have created a class named chocolate and given the attributes name and price and specified the methods for name, price, hashcode, equals. In driver I have created two objected and called equals and hashcode method to find whether the name and price of chocolate for two objects are same are not.

Answer:

|  |
| --- |
| **Chocolate Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_11;  import java.util.Objects;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Chocolate {  private String name;  private int price;  /\*\*  \* constructor with name and price  \* @param name  \* @param price  \*/  public Chocolate(String name, int price) {  this.name = name;  this.price = price;  }  /\*\*  \* to access name of chocolate  \* @return name  \*/  public String getName() {  return name;  }  /\*\*  \* to modify the name of chocolate  \* @param name  \*/  public void setName(String name) {  this.name = name;  }  /\*\*  \* to access price of chocolate  \* @return price  \*/  public int getPrice() {  return price;  }  /\*\*  \* to modify the price of chocolate  \* @param price  \*/  public void setPrice(int price) {  this.price = price;  }  @Override  public String toString() {  return "Chocolate{" + "name=" + name + ", price=" + price + '}';  }  @Override  public int hashCode() {  int hash = 7;  hash = 89 \* hash + Objects.hashCode(this.name);  hash = 89 \* hash + this.price;  return hash;  }  @Override  public boolean equals(Object obj) {  if (this == obj) {  return true;  }  if (obj == null) {  return false;  }  if (getClass() != obj.getClass()) {  return false;  }  final Chocolate other = (Chocolate) obj;  if (this.price != other.price) {  return false;  }  if (!Objects.equals(this.name, other.name)) {  return false;  }  return true;  }  }  **Driver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_11;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Driver {  /\*\*  \* @param args the command line arguments  \*/  public static void main(String[] args) {  // TODO code application logic here  System.out.println("Answer For Question11:Saisumithra Jagarlamudi");  //created an object obj1 for chocolate class  Chocolate obj1 = new Chocolate("Ferrero Rocher",25);  //created an object2 obj2 for chocolate class  Chocolate obj2 = new Chocolate("Ferrero Rocher",30);  System.out.println("Hash Code");  //calculating hashcode values using hashcode method with object obj1  System.out.println(obj1.hashCode());  //calculating hashcode values using hashcode method with object obj2  System.out.println(obj2.hashCode());  System.out.println("\*\*\*\*\*\*\*\*\*\*");  System.out.println("Equals");  //comparing the objects using equals method for objects obj1 and obj2  System.out.println(obj1.equals(obj2));  //comparing the objects using equals method by its name method for obj1 and obj2  System.out.println(obj1.getName().equals(obj2.getName()));  }  } |

Output:



1. (15-Points) Design Employee class and Employee driver class as follows:
2. **Employee Class implements Comparable<Employee**>

* Data fields named empId, empName and empSalary
* A constructor with parameters, listed in the same order as above.
* Create getter methods for all the parameters.
* A toString method that prints the empId, empName and empSalary. There should be one space between each value output.
* Because Employee implements the Comparable interface, you must also implement the compareTo method as defined by the Comparable interface. Define this method in such a way that the natural ordering of employees will be by id number, in ascending order.

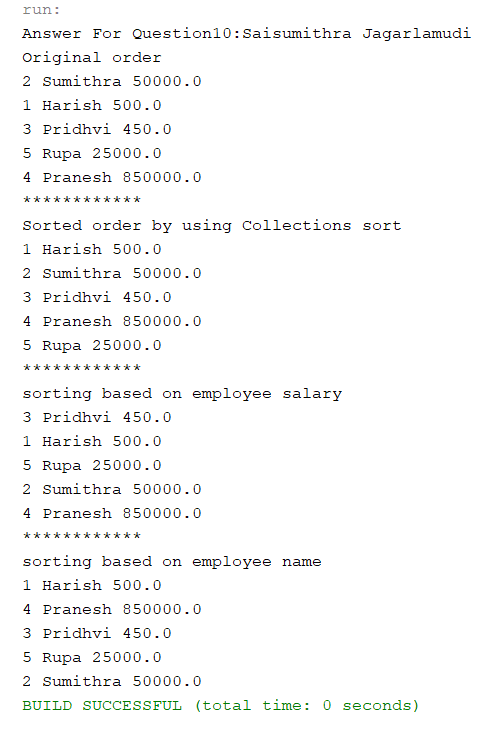
1. **EmployeeDriver Class**

* Begin by filling an ArrayList with at least 5 employees. Add employees in random order – not by id number, not by name, and not by salary. The original list should not be in order by any of these attributes.
* Use an enhanced for loop to print the original list.
* Call the one-parameter sort method of the Collections class to sort the list by its natural order (empId number) and then print the list again.
* Call the two-parameter sort method of the Collections class, supplying a new Comparator<Employee> that sorts by salary. Print the list again.
* Call the two-parameter sort method of the Collections class, supplying a new Comparator<Employee> that sorts by name. Print the list again.

Answer:

|  |
| --- |
| **Employee Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_12;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class Employee implements Comparable<Employee> {    private int empId;  private String empName;  private double empSalary;  public Employee(int empId, String empName, double empSalary) {  this.empId = empId;  this.empName = empName;  this.empSalary = empSalary;  }  public int getEmpId() {  return empId;  }  public String getEmpName() {  return empName;  }  public double getEmpSalary() {  return empSalary;  }  @Override  public String toString() {  return empId + " " + empName + " " + empSalary ;  }    @Override  public int compareTo(Employee e1)  {  return Integer.compare(this.empId,e1.empId);    }  }  **EmployeeDriver Class**  /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Exam\_12;  import java.util.\*;  /\*\*  \*  \* @author Saisumithra Jagarlamudi  \*/  public class EmployeeDriver {    public static void main(String[] args) {  ArrayList<Employee> empList = new ArrayList<>();  Employee emp1 = new Employee(2, "Sumithra", 50000);  Employee emp2 = new Employee(1, "Harish", 500);  Employee emp3 = new Employee(3, "Pridhvi", 450);  Employee emp4 = new Employee(5, "Rupa", 25000);  Employee emp5 = new Employee(4, "Pranesh", 850000);  empList.add(emp1);  empList.add(emp2);  empList.add(emp3);  empList.add(emp4);  empList.add(emp5);  System.out.println("Answer For Question10:Saisumithra Jagarlamudi");  System.out.println("Original order");  for (Employee emp : empList)  System.out.println(emp);  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("Sorted order by using Collections sort");  Collections.sort(empList);  for(Employee emp :empList)  {  System.out.println(emp);  }  Collections.sort(empList, new Comparator<Employee>()  {  @Override  public int compare(Employee emp1, Employee emp2)  {  return Double.compare(emp1.getEmpSalary(),emp2.getEmpSalary());  }  });  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("sorting based on employee salary ");  for(Employee emp:empList){  System.out.println(emp);  }  Collections.sort(empList, new Comparator<Employee>()  {  @Override  public int compare(Employee emp1, Employee emp2)  {  return emp1.getEmpName().compareTo(emp2.getEmpName());  }  });  System.out.println("\*\*\*\*\*\*\*\*\*\*\*\*");  System.out.println("sorting based on employee name ");  for(Employee emp : empList){  System.out.println(emp);  }  }  } |

Output:



Github Link: <https://github.com/Saisumithra/jagarlamudi_FinalExam.git>