**Exam 2 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 before week 13. 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.

**Exam 2 OBJECT-ORIENTED PROG 01FA20 100 pts**

1. (5-Points) (1D-Array - )Write a method that removes the duplicate elements from an array list of integers using the following header:

Public static void removeDuplicate(ArrayList<Integer> list)

Write a test program that prompts the user to enter 10 integers to a list and displays the distinct integers separated by exactly one space. Provide screenshot of executable code with input and output. Here is a sample run:

|  |
| --- |
| Enter ten integers: 34 5 3 5 6 4 33 2 2 4  The distinct integers are 34 5 3 6 4 33 2 |

1(A) Get the ArrayList with duplicate values.

Create another ArrayList.

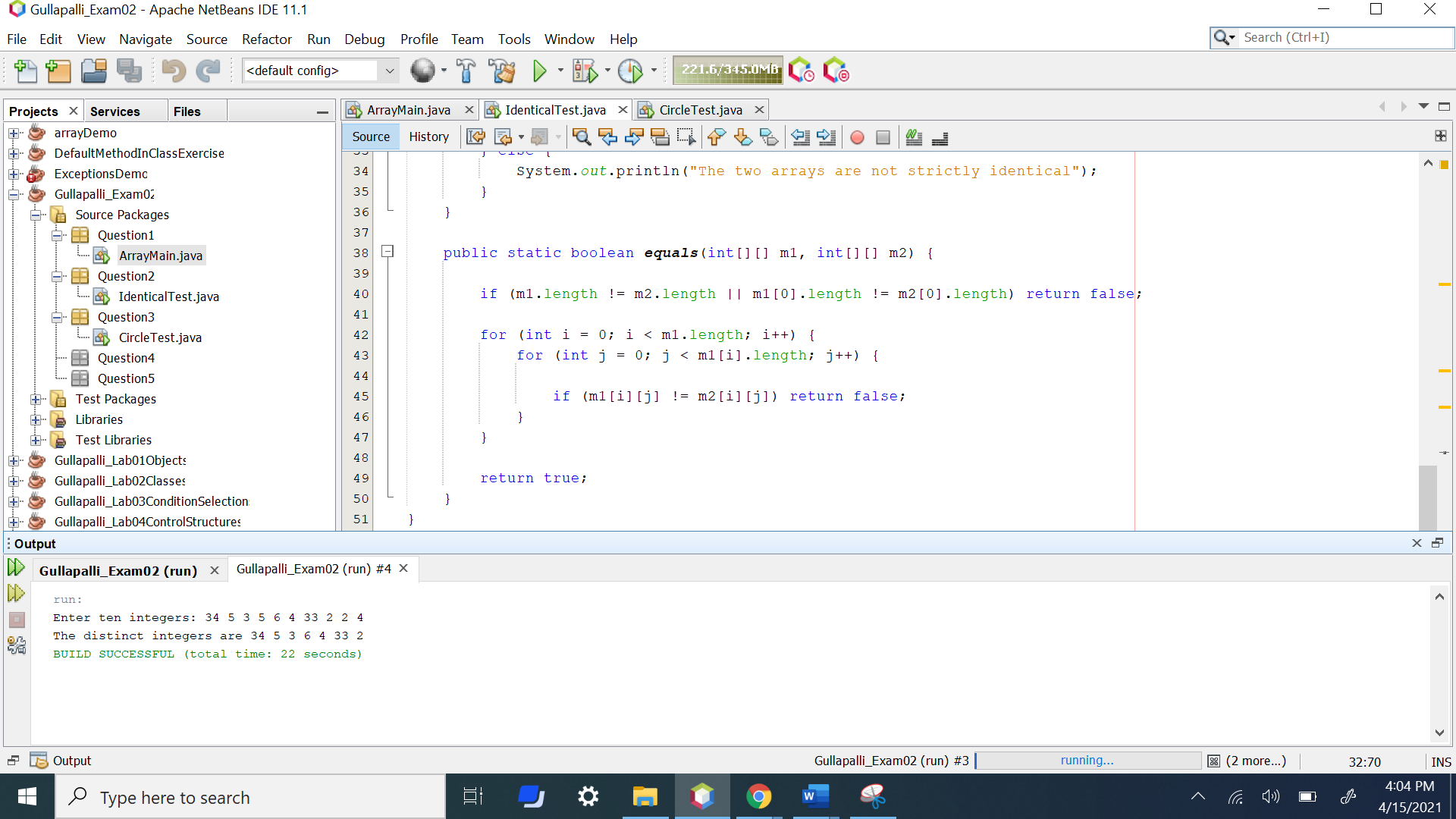
Traverse through the first arraylist and store the first appearance of each element into the second arraylist using contains() method.

The second ArrayList contains the elements with duplicates removed.

|  |
| --- |
| /\*  \* 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 Question1;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  import java.util.Scanner;  import java.util.ArrayList;  public class ArrayMain {  public static void main(String[] args) {    Scanner input = new Scanner(System.in);// Created scanner    ArrayList<Integer> list = new ArrayList<Integer>();// Created ArrayList    System.out.print("Enter ten integers: ");// enter ten integers  for (int i = 0; i < 10; i++) {  list.add(input.nextInt());  }    removeDuplicate(list);// Invoke removeDuplicate method  // Display the output  System.out.print("The distinct integers are ");  for (int i = 0; i < list.size(); i++) {  System.out.print(list.get(i) + " ");  }  System.out.println();  }  // Removes the duplicate elements from an array list of integers  public static void removeDuplicate(ArrayList<Integer> list) {  for (int i = 0; i < list.size() - 1; i++) {  for (int j = i + 1; j < list.size(); j++) {  if (list.get(i) == list.get(j))  list.remove(j);  }  }  }  } |

Output :-

|  |
| --- |
| Enter ten integers: 34 5 3 5 6 4 33 2 2 4  The distinct integers are 34 5 3 6 4 33 2 |



1. (5-Points) (2D- Array) The two-dimensional arrays m1 and m2 are strictly identical if their corresponding elements are equal. Write a method that returns true if m1 and m2 are strictly identical, using the following header:

public static boolean equals(int[][] m1, int[][] m2)

Write a test program that prompts the user to enter two 3 \* 3 arrays of integers and displays whether the two are strictly identical. Provide screenshot of executable code with input and output. Here are the sample runs.

|  |
| --- |
| Enter list1: 51 22 25 6 1 4 24 54 6  Enter list2: 51 22 25 6 1 4 24 54 6  The two arrays are strictly identical |

|  |
| --- |
| Enter list1: 51 25 22 6 1 4 24 54 6  Enter list2: 51 22 25 6 1 4 24 54 6  The two arrays are not strictly identical |

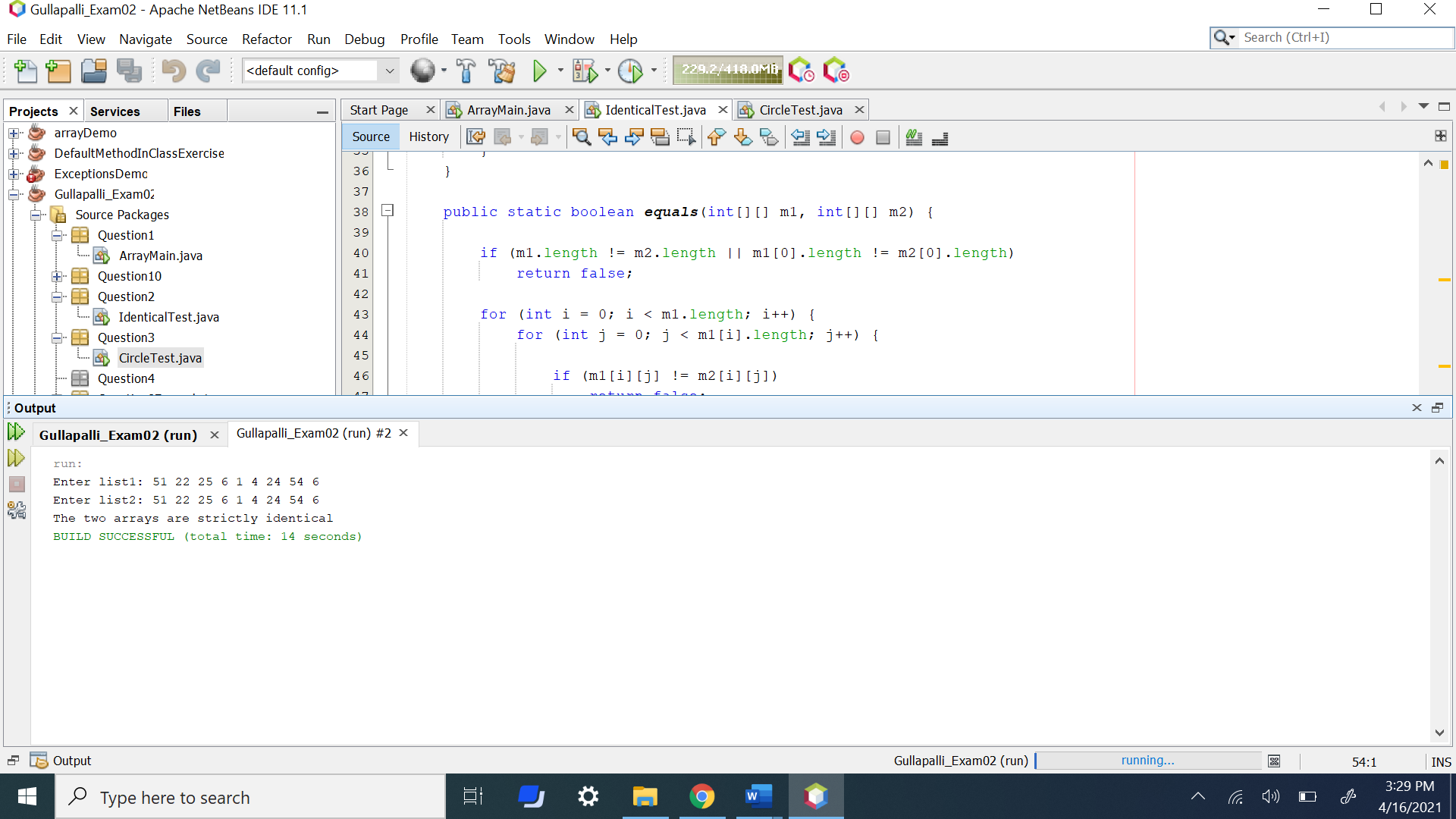
2(A)

|  |
| --- |
| /\*  \* 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 Question2;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  import java.util.Scanner;  import java.util.ArrayList;  public class IdenticalTest {  public static void main(String[] args) {  int[][] m1 = new int[3][3];  int[][] m2 = new int[3][3];  Scanner input = new Scanner(System.in);  System.out.print("Enter list1: ");  for (int i = 0; i < m1.length; i++)  for (int j = 0; j < m1[i].length; j++)  m1[i][j] = input.nextInt();  System.out.print("Enter list2: ");  for (int i = 0; i < m2.length; i++)  for (int j = 0; j < m2[i].length; j++)  m2[i][j] = input.nextInt();  if (equals(m1, m2)) {  System.out.println("The two arrays are strictly identical.");  } else {  System.out.println("The two arrays are not strictly identical");  }  }  public static boolean equals(int[][] m1, int[][] m2) {  if (m1.length != m2.length || m1[0].length != m2[0].length) return false;  for (int i = 0; i < m1.length; i++) {  for (int j = 0; j < m1[i].length; j++) {  if (m1[i][j] != m2[i][j]) return false;  }  }  return true;  }  } |

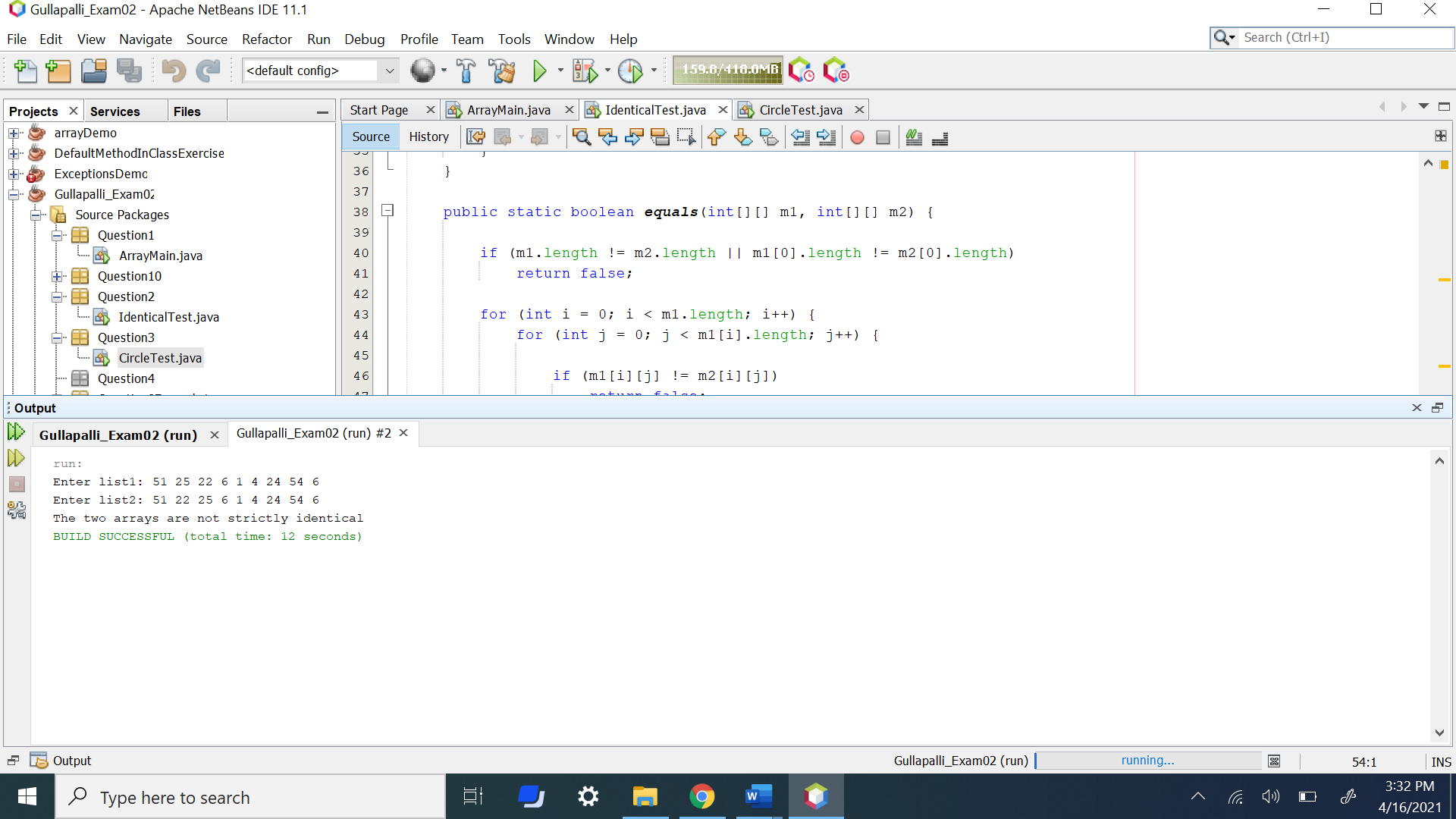
Output :-

|  |
| --- |
| Enter list1: 51 22 25 6 1 4 24 54 6  Enter list2: 51 22 25 6 1 4 24 54 6  The two arrays are strictly identical  Enter list1: 51 25 22 6 1 4 24 54 6  Enter list2: 51 22 25 6 1 4 24 54 6  The two arrays are not strictly identical |

The two arrays are strictly identical



The two arrays are not strictly identical



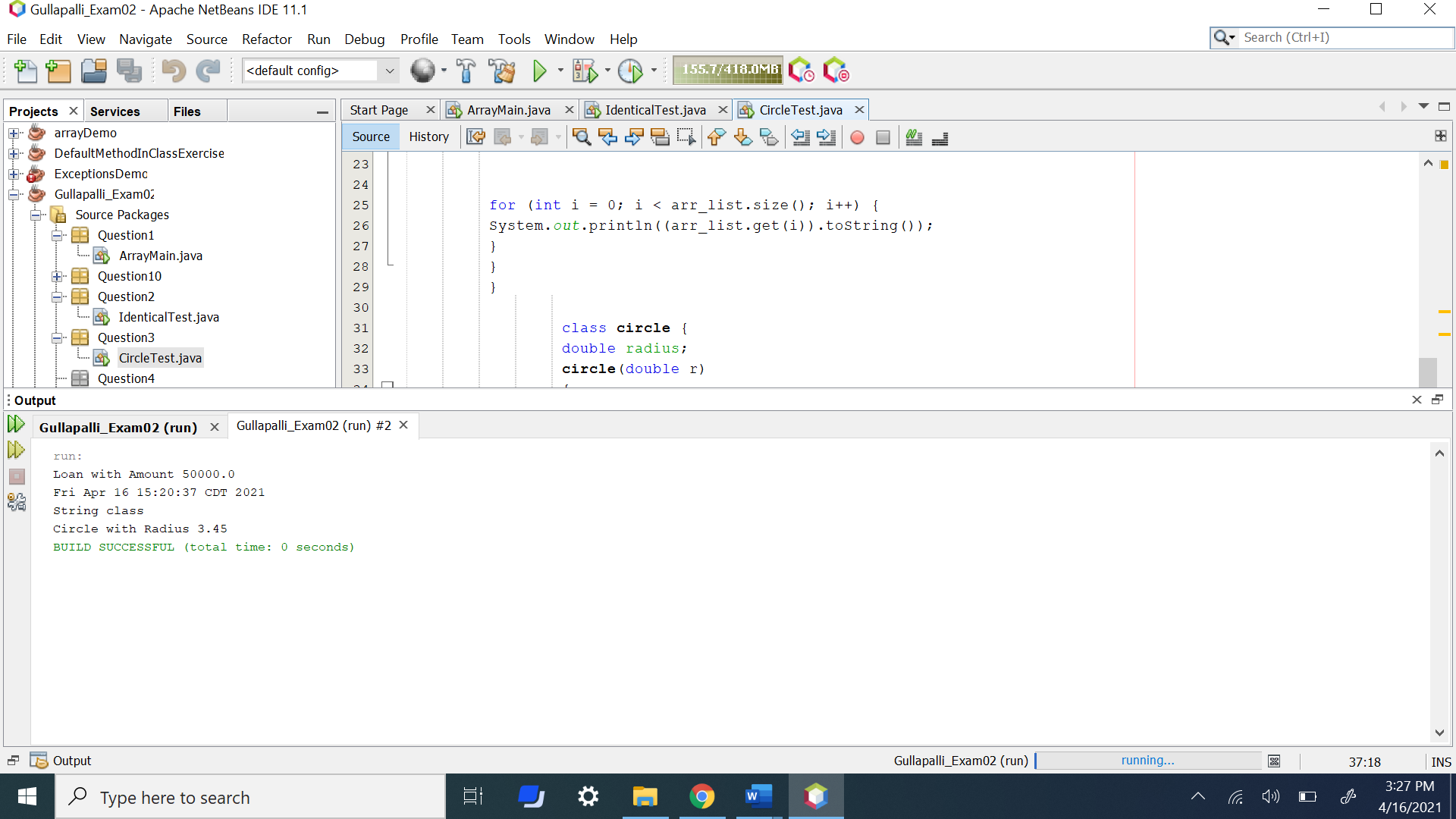
1. (10-Points) (Array List) Write a program that creates an ArrayList and adds a **Loan** object, a **Date** object (Use inbuilt method. No need to create separate class), a string, and a **Circle** object to the list, and use a loop to display all the elements in the list **by** invoking the object’s **toString**() method.

Note: For **Loan** and **Circle** you can use your own attributes and methods. **Constructor** and **tostring()** are mandatory requirements

|  |
| --- |
| /\*  \* 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 Question3;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  import java.util.ArrayList;  import java.util.Date;  public class CircleTest {    public static void main(String[] args) {  ArrayList<Object> arr\_list = new ArrayList<Object>();  arr\_list.add(new Loantest(50000.00));  arr\_list.add(new Date());  arr\_list.add(new String("String class"));  arr\_list.add(new circle(3.45));  for (int i = 0; i < arr\_list.size(); i++) {  System.out.println((arr\_list.get(i)).toString());  }  }  }  class circle {  double radius;  circle(double r)  {  this.radius=r;  }  public String toString()  {  return "Circle with Radius "+this.radius;  }  }  class Loantest  {  double amount;  Loantest(double amt)  {  this.amount=amt;  }  public String toString()  {  return "Loan with Amount "+this.amount;  }  } |

Output **:-**

|  |
| --- |
| Loan with Amount 50000.0  Fri Apr 16 16:40:54 CDT 2021  String class  Circle with Radius 3.45 |



1. (15-Points) What is Inheritance, Polymorphism and Late binding polymorphism? Explain and demonstrate with examples. Provide executable code screenshots for examples.

**4(A)** Inheritance:

Inheritance can be defined as the process where one class acquires the properties (methods) of another.

The class which inherits the properties of other is known as subclass (derived class, child class) and the class whose properties are inherited is known as superclass (base class, parent class). Keyword extends and it is used to inherit the properties.

Syntax:

class Super {

.....

.....

}

class Sub extends Super {

.....

.....

}

Example :

1. (10-Points) Design a class named **Person** and its two subclasses named **Student** and **Employee**. Make **Faculty** and **Staff** subclasses of **Employee**. A person has a name, address, phone number, and email address. A student has a grade and class status (Graduate). Define the status as a constant. An employee has an office, salary, and date hired. A faculty member has office hours and number of teaching subjects. A staff member has a title. Override the **toString** method in each class to display the class name and the person’s name.

Draw the UML diagram for the classes and implement them. Write a test program that creates a **Person**, **Student**, **Employee**, **Faculty**, and **Staff**, and invokes their **toString**() methods.

Note: All classes should have **toString()** Method.

|  |
| --- |
|  |

1. (10-Points) Design a new **Triangle** class that extends the abstract **GeometricObject** class. Draw the UML diagram for the classes **Triangle** and **GeometricObject** and then implement the **Triangle** class. Write a test program that prompts the user to enter three sides of the triangle, a color, and a Boolean value to indicate whether the triangle is filled. The program should create a **Triangle** object with these sides and set the color and filled properties using the input. The program should display the area, perimeter, color, and true or false to indicate whether it is filled or not. Provide screenshot of executable code with input and output.
2. (10-Points) What is an Enum in Java? Explain and demonstrate with some examples. Provide executable code screenshots for examples.
3. (10-points) Define the term abstract class in java? Explain and demonstrate with some examples. Provide executable code screenshots for examples.

8(A) Abstract class in java:

An abstract class is a class that is declared abstract — it may or may not include abstract methods. Abstract classes cannot be instantiated, but they can be subclassed. When an abstract class is subclassed, the subclass usually provides implementations for all of the abstract methods in its parent class.

A class which is declared with the abstract keyword is known as an abstract class in Java.

Two types:

1. abstract methods

2. non-abstract methods

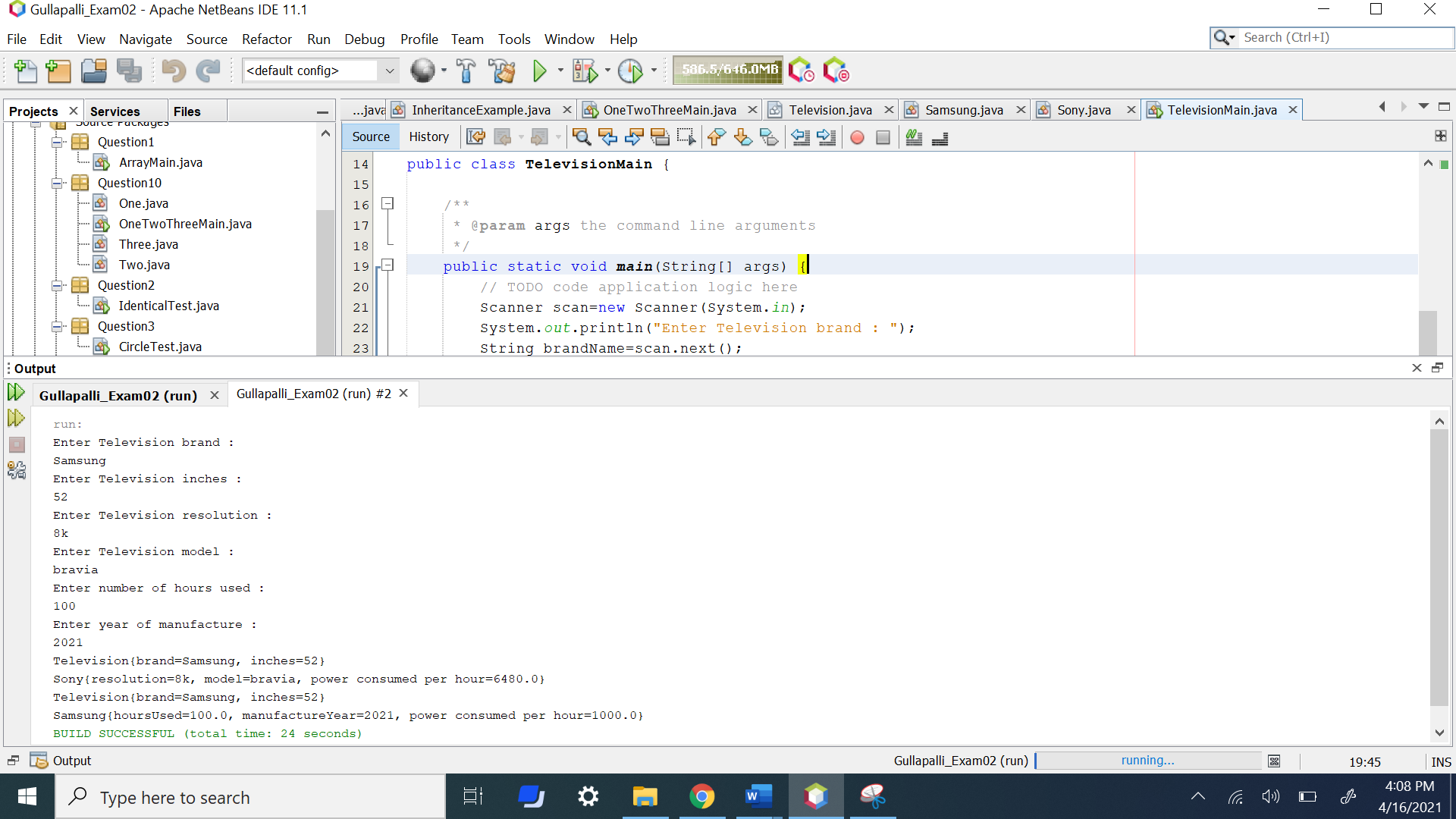
8(A) Example 1

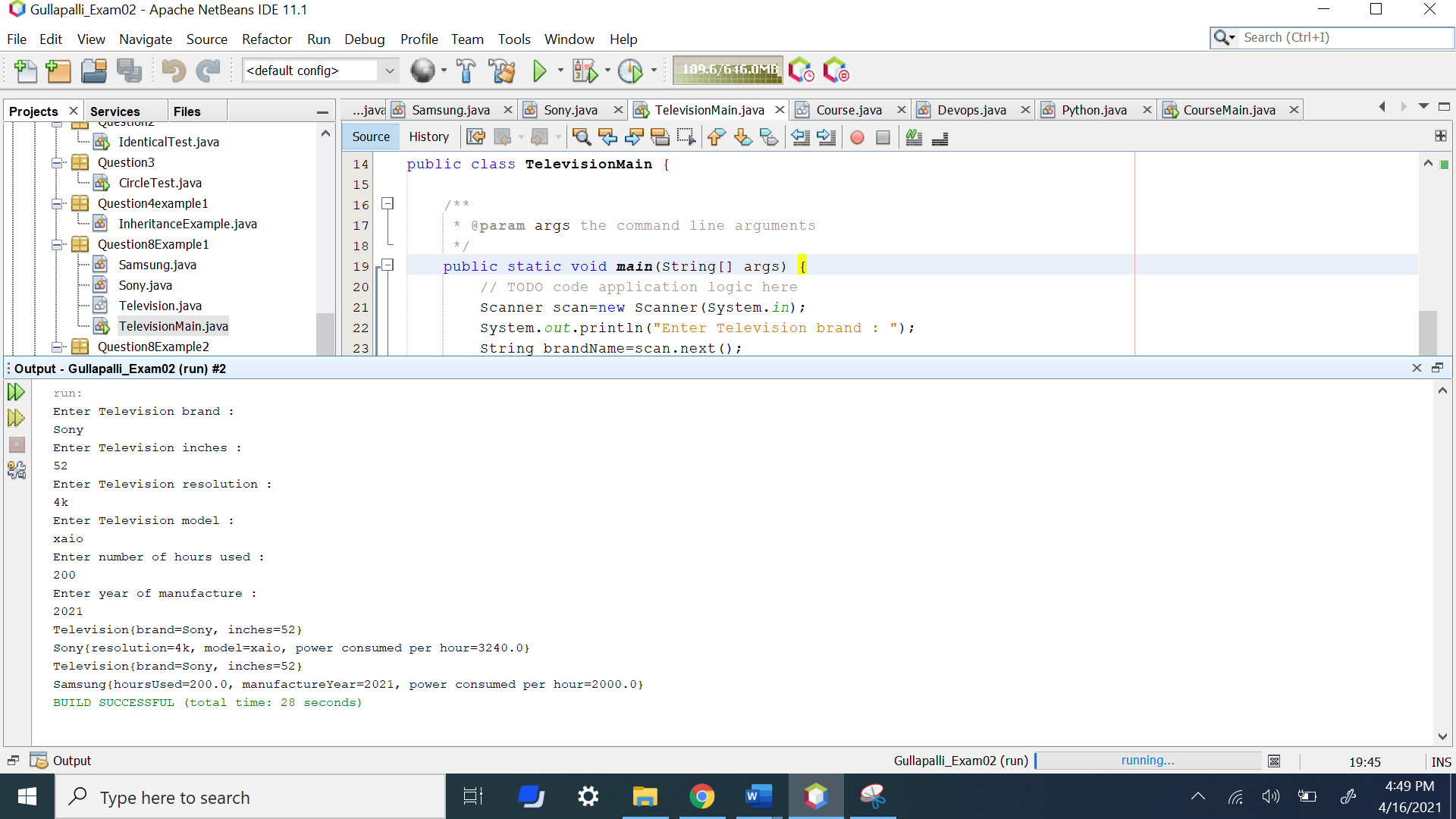
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Question8Example1;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public abstract class Television {  private String brand;  private int inches;  public Television(String brand, int inches) {  this.brand = brand;  this.inches = inches;  }  public String getBrand() {  return brand;  }  public void setBrand(String brand) {  this.brand = brand;  }  public int getInches() {  return inches;  }  public void setInches(int inches) {  this.inches = inches;  }  @Override  public String toString() {  return "Television{" + "brand=" + brand + ", inches=" + inches + '}';  }  public abstract double powerconsumption();      }  /\*  \* 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 Question8Example1;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class Samsung extends Television {  private double hoursUsed;  private int manufactureYear;  public Samsung(double hoursUsed, int manufactureYear, String brand, int inches) {  super(brand, inches);  this.hoursUsed = hoursUsed;  this.manufactureYear = manufactureYear;  }  public double getHoursUsed() {  return hoursUsed;  }  public void setHoursUsed(double hoursUsed) {  this.hoursUsed = hoursUsed;  }  public int getManufactureYear() {  return manufactureYear;  }  public void setManufactureYear(int manufactureYear) {  this.manufactureYear = manufactureYear;  }  public double powerconsumption(){  double consumed=hoursUsed\*10;  return consumed;  }  @Override  public String toString() {  return super.toString()+"\nSamsung{" + "hoursUsed=" + hoursUsed + ", manufactureYear=" + manufactureYear + ", power consumed per hour="+powerconsumption()+'}';  }          }  /\*  \* 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 Question8Example1;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class Sony extends Television{  private String resolution;  private String model;  public Sony(String resolution, String model, String brand, int inches) {  super(brand, inches);  this.resolution = resolution;  this.model = model;  }  public String getResolution() {  return resolution;  }  public void setResolution(String resolution) {  this.resolution = resolution;  }  public String getModel() {  return model;  }  public void setModel(String model) {  this.model = model;  }  public double powerconsumption(){  double consumed=0.0;  if(resolution.equalsIgnoreCase("4k")){  consumed= 1080\*3;  }  else if(resolution.equalsIgnoreCase("8k")){  consumed=2160\*3;  }  return consumed;    }  @Override  public String toString() {  return super.toString()+"\nSony{" + "resolution=" + resolution + ", model=" + model + ", power consumed per hour="+powerconsumption()+ '}';  }      }  /\*  \* 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 Question8Example1;  import java.util.Scanner;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class TelevisionMain {  /\*\*  \* @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("Enter Television brand : ");  String brandName=scan.next();  System.out.println("Enter Television inches : ");  int size=scan.nextInt();  System.out.println("Enter Television resolution : ");  String screenResolution=scan.next();  System.out.println("Enter Television model : ");  String tvModel=scan.next();  System.out.println("Enter number of hours used : ");  double usedHours=scan.nextDouble();  System.out.println("Enter year of manufacture : ");  int year = scan.nextInt();  Sony sonyTv =new Sony(screenResolution,tvModel,brandName,size);  Samsung samsungTv = new Samsung(usedHours,year,brandName,size);  System.out.println(sonyTv.toString());  System.out.println(samsungTv.toString());    }    } |

Output :

|  |
| --- |
| Enter Television brand :  Samsung  Enter Television inches :  52  Enter Television resolution :  8k  Enter Television model :  bravia  Enter number of hours used :  100  Enter year of manufacture :  2021  Television{brand=Samsung, inches=52}  Sony{resolution=8k, model=bravia, power consumed per hour=6480.0}  Television{brand=Samsung, inches=52}  Samsung{hoursUsed=100.0, manufactureYear=2021, power consumed per hour=1000.0}  Enter Television brand :  Sony  Enter Television inches :  52  Enter Television resolution :  4k  Enter Television model :  xaio  Enter number of hours used :  200  Enter year of manufacture :  2021  Television{brand=Sony, inches=52}  Sony{resolution=4k, model=xaio, power consumed per hour=3240.0}  Television{brand=Sony, inches=52}  Samsung{hoursUsed=200.0, manufactureYear=2021, power consumed per hour=2000.0} |

|  |
| --- |
|  |
|  |





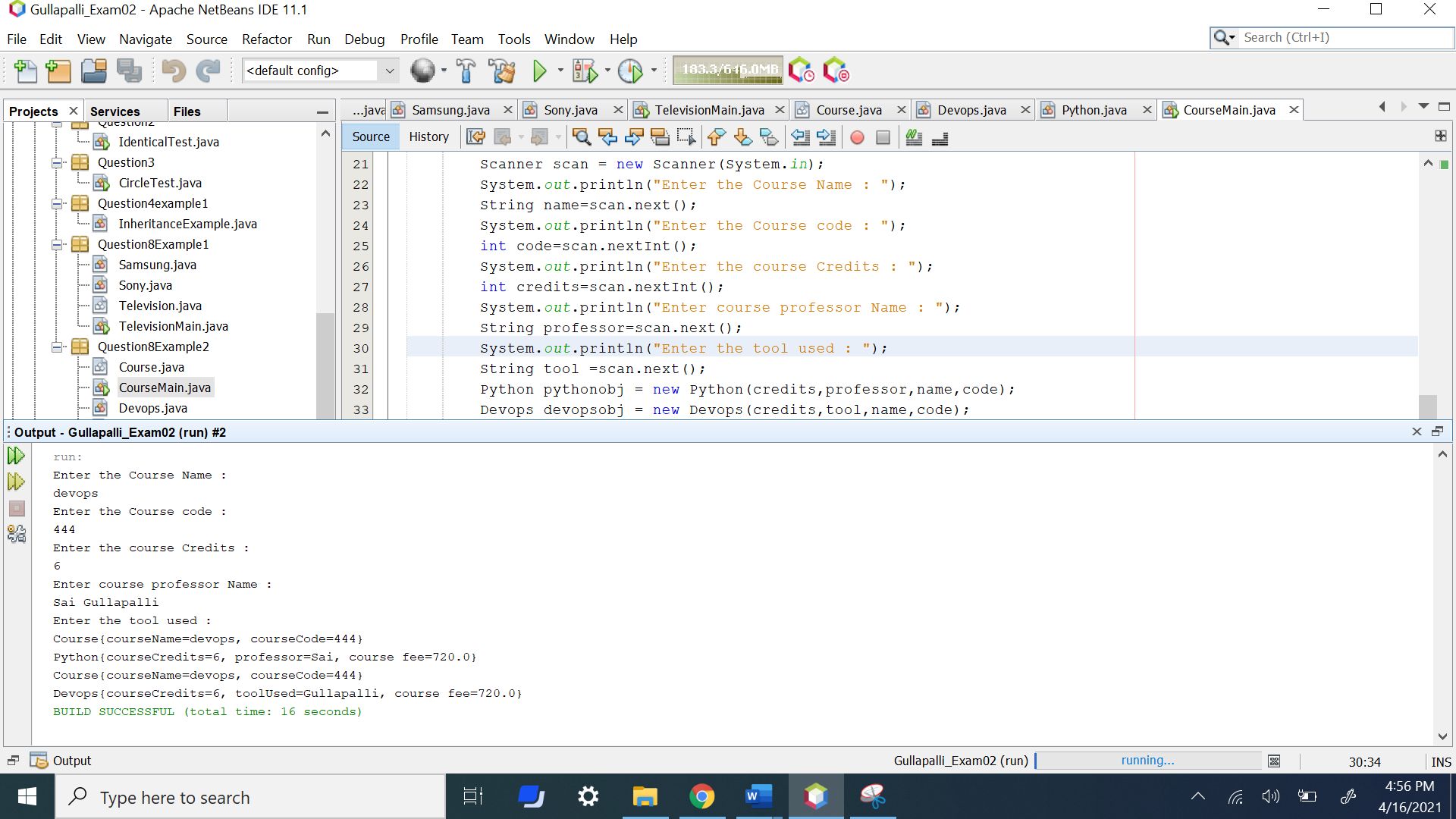
8(A) Example 2

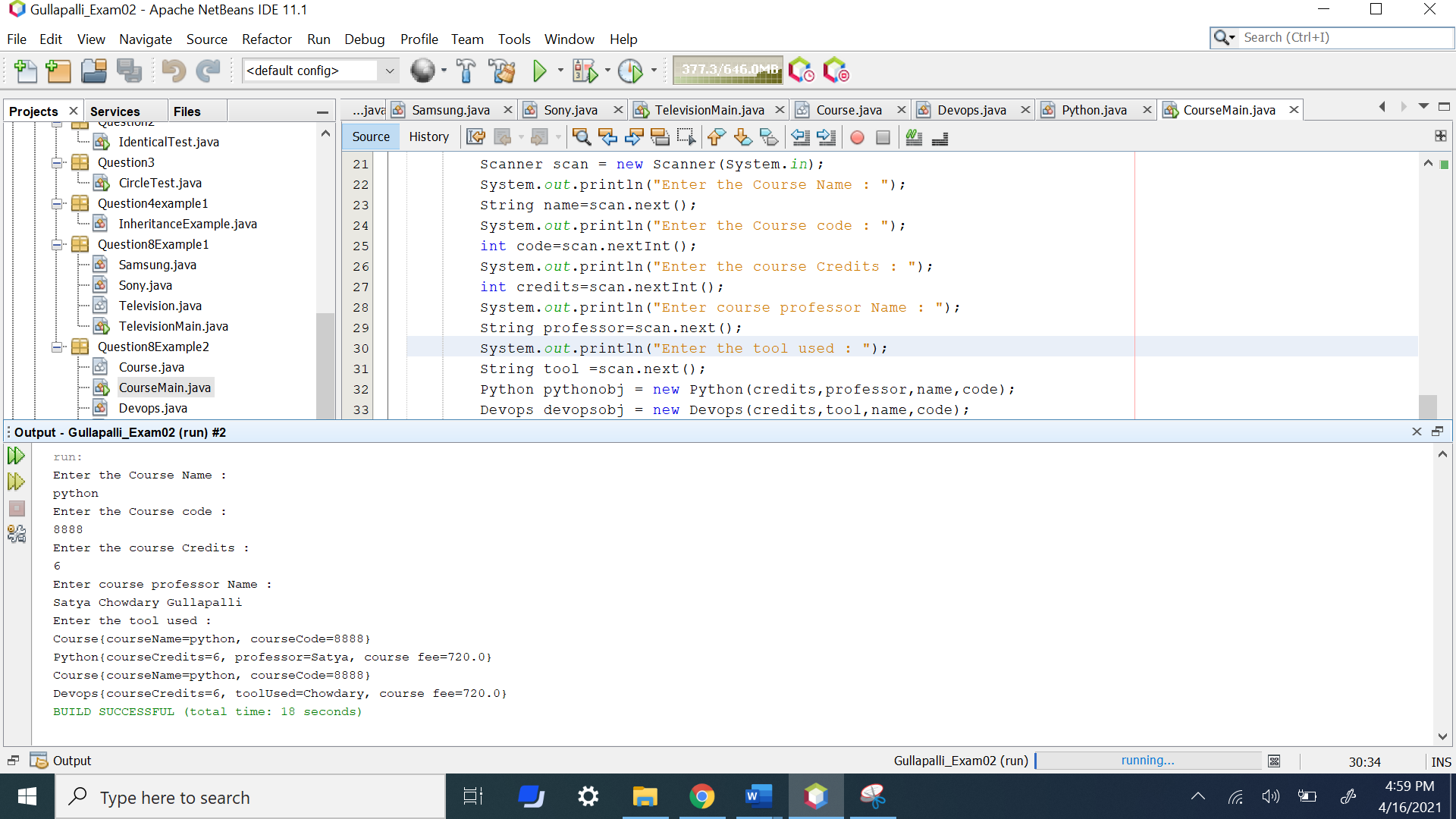
|  |
| --- |
| /\*  \* To change this license header, choose License Headers in Project Properties.  \* To change this template file, choose Tools | Templates  \* and open the template in the editor.  \*/  package Question8Example2;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public abstract class Course {  private String courseName;  private int courseCode;  public Course(String courseName, int courseCode) {  this.courseName = courseName;  this.courseCode = courseCode;  }  public String getCourseName() {  return courseName;  }  public void setCourseName(String courseName) {  this.courseName = courseName;  }  public int getCourseCode() {  return courseCode;  }  public void setCourseCode(int courseCode) {  this.courseCode = courseCode;  }  public abstract double courseCost();  @Override  public String toString() {  return "Course{" + "courseName=" + courseName + ", courseCode=" + courseCode + '}';  }      }  /\*  \* 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 Question8Example2;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class Devops extends Course {  private int courseCredits;  private String toolUsed;  public Devops(int courseCredits, String toolUsed, String courseName, int courseCode) {  super(courseName, courseCode);  this.courseCredits = courseCredits;  this.toolUsed = toolUsed;  }  public int getCourseCredits() {  return courseCredits;  }  public void setCourseCredits(int courseCredits) {  this.courseCredits = courseCredits;  }  public String getToolUsed() {  return toolUsed;  }  public void setToolUsed(String toolUsed) {  this.toolUsed = toolUsed;  }  public double courseCost(){  double cost = 0.0;  cost=courseCredits\*120;  return cost;  }  @Override  public String toString() {  return super.toString()+"\nDevops{" + "courseCredits=" + courseCredits + ", toolUsed=" + toolUsed + ", course fee="+courseCost()+'}';  }  }  /\*  \* 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 Question8Example2;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class Python extends Course {  private int courseCredits;  private String professor;  public Python(int courseCredits, String professor, String courseName, int courseCode) {  super(courseName, courseCode);  this.courseCredits = courseCredits;  this.professor = professor;  }  public int getCourseCredits() {  return courseCredits;  }  public void setCourseCredits(int courseCredits) {  this.courseCredits = courseCredits;  }  public String getProfessor() {  return professor;  }  public void setProfessor(String professor) {  this.professor = professor;  }  public double courseCost(){  double cost = 0.0;  cost=courseCredits\*120;  return cost;  }  @Override  public String toString() {  return super.toString()+"\nPython{" + "courseCredits=" + courseCredits + ", professor=" + professor +", course fee="+courseCost()+'}';  }      }  /\*  \* 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 Question8Example2;  import java.util.Scanner;  /\*\*  \*  \* @author Sai Satyanarayana Gullapalli  \*/  public class CourseMain {  /\*\*  \* @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("Enter the Course Name : ");  String name=scan.next();  System.out.println("Enter the Course code : ");  int code=scan.nextInt();  System.out.println("Enter the course Credits : ");  int credits=scan.nextInt();  System.out.println("Enter course professor Name : ");  String professor=scan.next();  System.out.println("Enter the tool used : ");  String tool =scan.next();  Python pythonobj = new Python(credits,professor,name,code);  Devops devopsobj = new Devops(credits,tool,name,code);  System.out.println(pythonobj.toString());  System.out.println(devopsobj.toString());    }    } |

Output :

|  |
| --- |
| Enter the Course Name :  devops  Enter the Course code :  444  Enter the course Credits :  6  Enter course professor Name :  Sai Gullapalli  Enter the tool used :  Course{courseName=devops, courseCode=444}  Python{courseCredits=6, professor=Sai, course fee=720.0}  Course{courseName=devops, courseCode=444}  Devops{courseCredits=6, toolUsed=Gullapalli, course fee=720.0}  Enter the Course Name :  python  Enter the Course code :  8888  Enter the course Credits :  6  Enter course professor Name :  Satya Chowdary Gullapalli  Enter the tool used :  Course{courseName=python, courseCode=8888}  Python{courseCredits=6, professor=Satya, course fee=720.0}  Course{courseName=python, courseCode=8888}  Devops{courseCredits=6, toolUsed=Chowdary, course fee=720.0} |

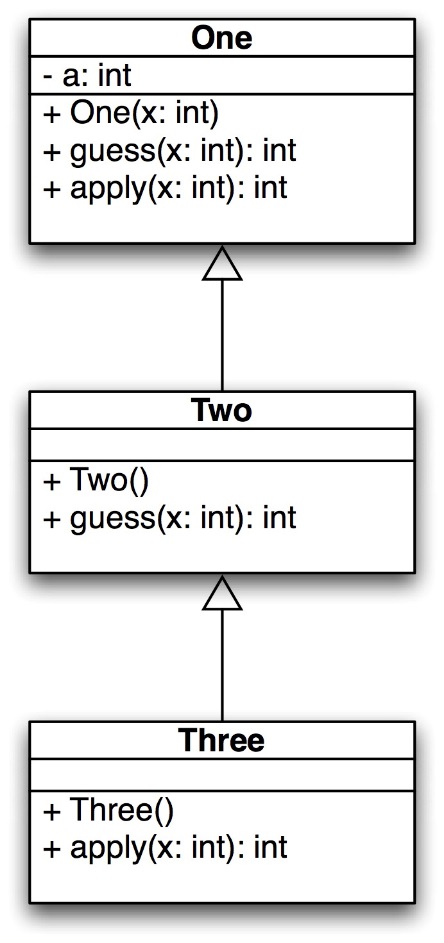
|  |
| --- |
|  |





1. (10-points) Define the term interface in java? Explain and demonstrate with some examples. Provide executable code screenshots for examples.
2. (15-Points) Consider the following code for three classes One, Two, and Three. (A UML diagram is included for your convenience.)

|  |
| --- |
| public class One {  private int a;  public One(int in){  a = in;  }  public int guess (int x){  System.out.println("One guess " + x);  return a + x;  }    public int apply (int x){  System.out.println("One guess " + x);  return guess(x + 3);  }  } // end class One  public class Two extends One {  public Two(){  super(11);  }  public int guess(int x){  System.out.println("Two guess " + x);  return super.guess(x)+10;  }    } // end class Two  public class Three extends Two {  public int apply(int x){  System.out.println("Three apply " + x);  return -10;  }  } // end class Three |



What is the output of the following code? Explain it.

|  |
| --- |
| public static void main(String[] args) {  One hippo = new Three();  System.out.println(hippo.guess(4));  System.out.println(hippo.apply(12));  One lion = new One(-1);  System.out.println(lion.guess(5));  System.out.println(lion.apply(6));  } // end |

10(A) Polymorphism in Java is a concept by which we can perform a single action in different ways. So polymorphism means many forms. There are two types of polymorphism in Java: compile-time polymorphism and runtime polymorphism. We can perform polymorphism in java by method overloading and method overriding.

Output :

|  |
| --- |
| Two guess 4  One guess 4  25  Three apply 12  -10  One guess 5  4  One guess 6  One guess 9  8 |

|  |
| --- |
|  |

