# Section A

Tutorial 1 Classes and Object

1. Given the following programs:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35 | // Program name: Computer.java  public class Computer { private String brand; private String model;  double price;  public Computer(String b, String m ,double p){  brand = b;  model = m;  price = p;  }  public void setBrand(String b) { brand = b;  }  public void setModel(String m) { model = m;  }  Public void setPrice(double p){  price = p;  }  public String getBrand() { return brand;  }  public String getModel() {  return model;  }  Public double price(){  return price;  }  public void displayInfo() { System.out.println("Brand: " + brand); System.out.println("Model: " + model);  System.out.println(“Price:RM” +price);  }  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // Program name: ComputerApp.java  public class ComputerApp {  public static void main(String [] args) {  Computer comp1 = new Computer("Apple", " iMac"); comp1.displayInfo();  }  } |

* 1. Identify the followings:
     1. Which file is class implementation?
* Computer.java
  + 1. Which file contains main method?
* ComputerApp.java
  + 1. Which file contains constructor method?
* Computer.java
  1. What is the output of the above program?
* Brand: Apple

Model: iMac

* 1. Define a new object named comp2. Create the object and display with the brand and model name of your own computer.
* Computer comp2 = new Computer(“Acer”,”Espire”);

Comp2.displayInfo();

* 1. Modify the above programs to add an instance variable named price of type

double.

1. Given the following programs:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17 | public class EmployeeTest {  public static void main(String args[]) {  /\* Create two objects using constructor \*/ Employee empOne = new Employee("James Smith"); Employee empTwo = new Employee("Mary Anne");  // Invoking methods for each object created empOne.empAge(26);  empOne.empDesignation("Senior Software Engineer"); empOne.empSalary(1000);  empOne.printEmployee();  empTwo.empAge(21); empTwo.empDesignation("Software Engineer"); empTwo.empSalary(500); empTwo.printEmployee();  }  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10 | public class Employee { String name;  int age;  String designation; double salary;  // This is the constructor of the class Employee public Employee(String name) {  this.name = name;  } |

|  |  |
| --- | --- |
| 11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30 | // Assign the age of the Employee to the variable age. public void empAge(int empAge) {  age = empAge;  }  /\* Assign the designation to the variable designation.\*/ public void empDesignation(String empDesig) {  designation = empDesig;  }  /\* Assign the salary to the variable salary.\*/ public void empSalary(double empSalary) {  salary = empSalary;  }  /\* Print the Employee details \*/ public void printEmployee() {  System.out.println("Name:"+ name ); System.out.println("Age:" + age ); System.out.println("Designation:" + designation ); System.out.println("Salary:" + salary);  }  } |

* 1. Trace the above program and identify the output.

Name:James Smith

Age:26

Designation:Senior Software Engineer

Salary:1000.0

Name:Mary Anne

Age:21

Designation:Software Engineer

Slary:500.0

1. Consider the following class definition.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21 | // Program name: Staff.java public class Staff {  private String staffNo; private String name; private int age;  private String department;  public Staff(String m, int a) { staffNo = m;  name = “Unspecified”;  age = a;  department = “Unknown”  }  public Staff(String m,String n, int a, String dept) { staffNo = m;  name = n; age = a;  department = dept;  } |

|  |  |
| --- | --- |
| 22  23  24  25  26  27  28 | public void displayInfo() { System.out.println("staffNo: " +staffNo); System.out.println("Name: " + name); System.out.println("Age: " + age); System.out.println("Department: " + department);  }  } |

1. Write the output of the following statements.
   1. Staff staff1 = new Staff ("A172455", "Aideen Adam", 20, “SK”);

staff1.displayInfo();

stafNo:A172455

Name:Aideen Adam

Age:20

Department:SK

* 1. Staff staff2 = new Staff ("A123457", 21); staff2.displayInfo();

StafNo:A123457

Name:

Age:21

Department:

1. State the reason why the following statement will result in a compilation error.

Staff staff2 = new Staff ("A123457", “Saidatul Aisyah”);

Because it has the same variable name as staff2

# Section B

1. The following code is to calculate and display the area of rectangle:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12 | // Program name: Rectangle.java public class Rectangle {  private int width; private int height;  public Rectangle(int w, int h) {  width = w;  height = h;  }  public int getWidth() {  return width;  } |

|  |  |
| --- | --- |
| 13  14  15  16  17  18  19  20  21  22  23  24  25  26 | public int getHeight() {  return height;  }  public void setWidth(int w) {  width = w;  }  public void setHeight(int h) {  height = h;  }  public int getArea() {  return width\*height;  }  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | // Program name: RectangleApp.java  public class RectangleApp {  public static void main(String [] args) {  Rectangle rect1 = new Rectangle(5, 3); System.out.println (rect1.getArea());  }  } |

In the class of Rectangle,

* 1. Complete the constructor in line 7 and 8.
  2. Complete the method getWidth and getHeight in line 11 and line 14.
  3. Complete the method setWidth and setHeight in line 17 and line 20.
  4. Complete the method getArea in line 23.
  5. Understand and explain the flow of programs.

The file rectangle.java contain constructors of the program with rectangle as its object. The method are getWidth,GetHeight,setWidth,setHeight and getArea. The file RectangleApp.java contains the program that run through the Rectangle Class constructors and method from the file Rectangle.java

* 1. What is the output?

15

1. Given the following programs and the sample output:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | // Program name: Account.java import java.text.DecimalFormat;  class Account{  DecimalFormat df = new DecimalFormat("0.00"); int acc\_no;  String name; double amount;  //Method to initialize object  public Account (int a,String n,double amt){ |

|  |  |
| --- | --- |
| 12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43 | acc\_no=a;  name = n;  amount = amt;  }  //deposit method  void deposit(double amt){  System.out.println(df.format(amt) + “deposited”);  amount=amount+amt;  }  //withdraw method  void withdraw(double amt){ if(amount<amt){  System.out.println("Insufficient Balance");  }  else{  System.out.println(df.format(amt)+”withdrawn”);  amount=amount-amt;  }  }  //balance method  public double getBalance(){  return amount;  }  //method to display the values of an object public String toString(){  return acc\_no+" "+name+" "+ df.format(amount);  }  } |

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | // Program name: AccountApp.java import java.text.DecimalFormat;  class AccountApp{  public static void main(String[] args){ DecimalFormat df = new DecimalFormat("0.00"); Account cust1=new Account(136340,"Azue",1500); System.out.println(cust1.toString()); cust1.deposit(2000); System.out.println(df.format(cust1.getBalance())); cust1.withdraw(500); System.out.println(df.format(cust1.getBalance()));  }  } |

## Sample Output:

136340 Azue 1500.00

2000.00 deposited

3500.00

500.00 withdrawn 3000.00

In the class of Account,

* 1. Complete the constructor method in line 12-14.
  2. List all static attribute(s).

acc\_no

name

* 1. List all dynamic attribute(s).

amount

* 1. Complete the deposit method in line 19-20.
  2. Complete the withdraw method in line 29-30.
  3. Complete the accessor method getBalance in line 36.
  4. Understand and explain the flow of programs.

The constructor is public Account() that hold data such as int a that hold data acc\_no , String n that hold data name and double amt that hold data amount

* 1. Define a new object named cust2. Create the object and display with Account no 123456, name Ali and current amount 2000. Deposit 250 and display the balance deposited. Withdraw 2500 and display the balance withdrawn. Write the output.

Output:

123456 Ali 2000.00

250 deposited

2250.00

2500 withdrawn

Insufficient Balance

Account cust2 = new Account(123456,”Ali”,2000);

System.out.println(cust2.toString());

Cust2.deposit(250);

System.out.println(df.format(cust2.getBalance()));

Cust2.withdraw(2500);

System.out.println(df.format(cust2.getBalance()));

# Section C

1. A class named Circle is defined as follows.
   * Private instance variable named radius that represents the radius of the circle.
   * The constructor Circle()that initialised radius to 0.0.
   * setRadius(double) method to set a new radius value
   * getRadius() method that returns the radius
   * getArea() method that returns the area of the circle.
   * getCircumference() method to returns the circumference of the circle. The UML Class Diagram for class Circle is as the following.

|  |
| --- |
| **Circle** |
| -radius: double |
| +Circle()  +setRadius(radius:double): void  +getRadius(): double  +getArea(): double  +getCircumference(): double |

1. Write the program for Circle class in the following space given.

class Circle{

private radius;

public class Circle(){

radius=0.00;

}

public void setRadius(double radius){

this.radius=radius;

}

public double getRadius(){

return radius;

}

public double getArea(){

return Math.PI\*Math.pow(radius,2);

}

public double getCircumference(){

return 2\*Math.PI\*radius;

}

}

1. Run the programs with the following sample input. Write the output in the following space given.

// File name: CircleApp.java import java.util.Scanner; import java.text.DecimalFormat;

public class CircleApp {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

DecimalFormat df = new DecimalFormat("0.00"); double radius;

int N = sc.nextInt();

for (int i = 1; i <= N; i++) {

Circle myCircle = new Circle(); // create Circle object radius = sc.nextDouble();

myCircle.setRadius(radius); // sets radius

System.out.println("Case #" + i + ": " + df.format(myCircle.getRadius()) +

" " + df.format( myCircle.getArea) + " " + df.format(myCircle.getCircumference());

}

}

}

## Sample Input Output

|  |  |
| --- | --- |
| **Sample Input** | **Sample Output** |
| 2  9.0  5.5 | Case #1: 9.00 254.47 56.55  Case #2: 5.00 78.54 31.42 |