

MID TERM ASSIGNMENT
ACADEMIC YEAR:20 TO 20

Hall Ticket No. :

1	9	B	Q	1	A	0	5	I	2
---	---	---	---	---	---	---	---	---	---

Name of the Student : Popuri. Manjunath

Course : B.Tech

Branch : ECE/CSE/EEE/IT

Subject : JAVAPROGRAMMING

ASSIGNMENT / MARKS DETAILS

To be filled by the Student			To be filled by the Subject Teacher		
Submission Date	Assignment	Signature of the Student	Max Marks	Marks Obtained	Signature of Subject Teacher
21-09-20	1	Manjunath	5		

INSTRUCTIONS TO THE STUDENTS

1. The assignment should be submitted to the subject teacher on or before the given schedule.
2. Answer should be written on both sides of the paper.

INSTRUCTIONS TO THE SUBJECT TEACHER

1. The Subject teacher has to value with red ball point pen only.
2. The Subject teacher should award the marks on the left hand side of the margin and at the end of the each answer.
3. Do not correct the marks by overwriting or by scratching and writing.
4. The Subject teacher has to post marks in the space provided.

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY, NAMBUR

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

Vision of the Department

To facilitate quality education by focusing on assimilation, generation and dissemination of knowledge in the area of Computer Science & Engineering to transform students into socially responsible engineers.

Mission of the Department

- Equip our graduates with the knowledge by *student centric teaching-learning process* and expertise to contribute significantly to the software industry and to continue to grow professionally.
- To train *socially responsible, disciplined engineers* who work with good leadership skills and can contribute for nation building.
- To make our graduates *aware of cutting edge technologies* and make them industry-ready engineers.
- To shape the department into a *centre of academic and research excellence*.

Program Educational Objectives

PEO-1	To provide the graduates with solid foundation in Computer Science and Engineering along with the fundamentals of Mathematics and Sciences with a view to impart in them high quality technical skills like modelling, analyzing, designing, programming and implementation with global competence.
PEO-2	To prepare and motivate graduates with recent technological developments related to core subjects like programming, databases, design of compilers and Network Security aspects and future technologies so as to contribute effectively for Research & Development by participating in professional activities like publishing and seeking copy rights.
PEO-3	To train graduates to choose an appropriate career in employment, higher education or entrepreneurship by empowering them to excel in competitive examinations, by preparing them for lifelong learning and by inculcating in them ethical leadership skills.
PEO-4	To train the graduates to have basic interpersonal skills and sense of social responsibility that paves them a way to become good team members and leaders.

JAVA PROGRAMMING

19BQ1A0512

ASSIGNMENT - 1

1. Write about the role of JVM and JAVA API in developing the platform independent java program with suitable example.

Role of JVM in JAVA

JVM stands for java virtual machine which is abstract or virtual computing machine is the implementation of Java virtual machine specification. It interprets the compiled java code known as the byte code and helps in programme execution depending upon the specific platform.

Java is platform independent language i.e., it can run on any platform without rewriting the code. This feature is supported by JVM.

Each JVM has :-

- * A Instruction Set.
- * A stack
- * A garbage collection heap
- * Method area
- * A set of registers
- * An Execution Environment.

Basically JVM is set of computer programs and data structures that run compiled byte code on any machine making Java "compile once, run anywhere" language.

JVM converts the byte code into machine (or) platform specific code and then runs it.

Steps in JVM implementation:-

- * Loads the class file.
- * Check whether .class file has the required byte code.
- * Interprets the byte code and convert them into machine specific code.
- * Removes useless objects and do garbage collection.

Source code (program.java) ---- compile ----> Byte code (program.class)

Byte code ----- JVM -----> Machine code

JAVA API

An application Programming Interface (API), in the context of java, is a collection of prewritten packages, classes, and interfaces with their respective methods, fields and constructors. Similar to a user interface, which facilitates interaction between humans and computers, an API serves as a software program interface facilitating interaction.

The API is a library of available Java classes, Packages and interfaces. The API helps programming task mostly in JAVA by classes and packages which are helpful in minimizing the number of lines written within pieces of code. The official API includes packages, e.g., applet packages, graphics and GUI Swing packages, input/output (IO) packages, and abstract windows toolkit (AWT).

```
import java.util.*;
```

The import statement imports all classes in the API's Java.util Package and makes them available to programmer.

Source: JAVA CODE.

(2) With an Example program explain the concept of classes and nested classes in JAVA.

Classes →

A class describes what the object will be, but it separate from the object itself.

In other words, classes can be described as blueprints, descriptions (or) definitions for an object. you can use the same class as a blue print for creating multiple objects. The first step is to define the class, which then becomes a blue print for object creation.

Each class has a name, and each is used to define attributes and behavior.

Attributes eg:- name, height, weight, gender, age.
behavior eg:- walk, run, sleep, jump, speak.

* Each class has a different types of instance variables, constructors, setters and getters, and methods

eg:- →

```
Public class Animal {
    void bark() {
        System.out.println("woof --- woof");
    }
}
```

* We declared a bark() method in our class Animal

* In order to use class and its method we need to declare an object of class

```
class MyClass {
    public static void main(String args[]) {
        Animal fox = new Animal();
        fox.bark();
    }
}
```

* Now, fox is an object for our class Animal and we called its method bark() using object fox as fox.bark()

Output

woof --- woof

Nested Classes :->

JAVA supports nesting classes; a class can be a member of another class. We use inner class to logically group classes and interfaces in one place so that it can be more readable and maintainable. Additionally, it can access all the members of outer class including private data members and methods.

Creating an inner class is quite simple. Just write a class within a class. Unlike a class, an inner class can be private, it cannot be accessed from an object outside the class.

Example

```
public class Person {
    String name = "VIT";
    class Hand {
        public void shake() {
            System.out.println("Hi");
            System.out.println(name);
        }
    }
}
```

→ Here, we declared an outer class Person and has a variable name and an inner class Hand which has a method shake() prints

Hi
VIT
to the screen when it is called.

(3). Design a class RailwayTicket having Instance Variables / datamembers as name, coach, mobno, amt and totalamt also having methods Void accept(), Void update(), Void display().

Types of coaches	amount
First- AC	700
Second- AC	500
Third- AC	250
sleeper	alone

Write main() method to create an object of a class and call above methods.

```

import java.io.*;
import java.util.Scanner;

public class RailwayTicket {
    String name;
    String coach;
    long mobno;
    int amt;
    int totalamt;
    Scanner sc = new Scanner(System.in);

    public void accept() {
        System.out.println("Enter name:");
        name = sc.nextLine();
        System.out.println("Enter coach:");
        coach = sc.nextLine();
        System.out.println("Enter mobile number:");
        mobno = sc.nextLong();
        System.out.println("Enter ticket amount:");
        amt = sc.nextInt();
    }
}
    
```



```

public void update () {
    int no;
    System.out.println("1. First-AC 100" + "\n"
        + "2. Second-AC 500" + "\n"
        + "3. Third-AC 250" + "\n"
        + "4. Sleeper None" );

```

```

    System.out.println("Enter your coach number");
    no = sc.nextInt();

```

```

    switch (no) {

```

```

        case 1 :

```

```

            totalamt = amt + 100;

```

```

            break;

```

```

        case 2 :

```

```

            totalamt = amt + 500;

```

```

            break;

```

```

        case 3 :

```

```

            totalamt = amt + 250;

```

```

            break;

```

```

        case 4 :

```

```

            totalamt = amt + 0;

```

```

            break;

```

```

        default :

```

```

            System.out.println("Enter valid
                                coach
                                number");

```

```

    }
}

```

```
public void display() {  
    System.out.println("Customer details---");  
    System.out.println("Name : " + name);  
    System.out.println("Coach : " + coach);  
    System.out.println("Total amount : "  
                        + totalamt);  
    System.out.println("mobile number : "  
                        + mobno);  
}  
public static void main (String args[]) {  
    RailwayTicket r = new RailwayTicket();  
    r.accept();  
    r.update();  
    r.display();  
}  
}
```

Output →

Enter name :

IRONMAN

Enter coach :

First-AC

Enter mobile number :

0123456789

Enter ticket amount :

200

1. First-AC 700

2. Second-AC 500

3. Third-AC 250

4. Sleeper None

Enter your coach number :

1

Customer details are -----

Name : IRONMAN

Coach : First-AC

Total amount : 900

Mobile Number : 0123456789

(4) Design a class to overload a function volume() as follows:

(i) double volume(double r) - with radius 'r' as argument returns volume of sphere using formula

$$V = \frac{4}{3} \times \frac{22}{7} \times r \times r \times r$$

(ii) double volume(double h, double r) with height 'h' and radius 'r' as arguments, returns the volume of cylinder using formula:

$$V = \frac{22}{7} \times r^2 \times h$$

(iii) double volume(double l, double b, double h) with length 'l', breadth 'b' and height 'h' as arguments, returns the volume of cuboid using formula:

$$V = l \times b \times h$$

```
import java.io.*;
class Volume {
```

```
    public double volume(double r) {
        return 4/3 * 22/7 * r * r * r;
    }
```

```
public double volume(double h, double r) {  
    return 22/7 * r * r * h;  
}  
public double volume(double l, double b, double h) {  
    return l * b * h;  
}  
public static void main(String args[]) {  
    Volume v = new Volume();  
    System.out.println(v.volume(3.0));  
    System.out.println(v.volume(3.0, 4.0));  
    System.out.println(v.volume(3.0, 4.0, 5.0));  
}
```

Output:

81.0

144.0

60.0