**Assignment 3**

1. What is Debugging?

Debugging is the process of identifying and fixing any issues in the source code of a program.

2. What are the default packages present in java?

Java compiler imports java.lang package internally by default. It provides the fundamental classes that are necessary to design a basic Java program.

3.Program to print a) Full Star b) Left Star c) Right Star?

a)

**package** Star;

**import** java.util.\*;

**public** **class** FullStar {

**public** **static** **void** main(String args[])

{

**int** n, i, j, space = 1;

System.***out***.print("Enter the number of rows: ");

Scanner s = **new** Scanner(System.***in***);

n = s.nextInt();

space = n - 1;

**for** (j = 1; j<= n; j++)

{

**for** (i = 1; i<= space; i++)

{

System.***out***.print(" ");

}

space--;

**for** (i = 1; i <= 2 \* j - 1; i++)

{

System.***out***.print("\*");

}

System.***out***.println("");

}

space = 1;

**for** (j = 1; j<= n - 1; j++)

{

**for** (i = 1; i<= space; i++)

{

System.***out***.print(" ");

}

space++;

**for** (i = 1; i<= 2 \* (n - j) - 1; i++)

{

System.***out***.print("\*");

}

System.***out***.println("");

}

}

}

b)

**package** Star;

**public** **class** LeftStar {

**public** **static** **void** StarleftTriangle(**int** k)

{

**int** a, b;

**for** (a = 0; a < k; a++) {

**for** (b = 2 \* (k - a); b >= 0; b--) {

System.***out***.print(" ");

}

**for** (b = 0; b <= a; b++) {

System.***out***.print("\* ");

}

System.***out***.println();

}

}

**public** **static** **void** main(String args[])

{

**int** k = 5;

*StarleftTriangle*(k);

}

}

c)

**package** Star;

**public** **class** RightStar {

**public** **static** **void** main(String args[]) {

**int** i, j, row=6;

**for**(i=0; i<row; i++)

{

**for**(j=0; j<=i; j++)

{

System.***out***.print("\* ");

}

System.***out***.println();

}

}

}

4.Can we have an else condition without if condition?

It is not possible to write else condition without if condition.

5. Other than Boolean can we use anything in if condition?

Nothing else can be used inside if condition.

6. Why break is important in switch case?

The case statements in a switch statement are simply labels. When you switch on a value, the switch statement essentially does a goto to the label with the matching value. This means that the break is necessary to avoid passing through to the code under the next label.

7. Understand the naming conventions?

the Java naming convention is to always start with a lowercase letter and then capitalize the first letter of every subsequent word. Variables in Java are not allowed to contain white space, so variables made from compound words are to be written with a lower camel case syntax.

8. Write add program using oops concept?

Add using polymorphism

**Interface**

**public** **interface** Calculation2 {

**public** **int** cal2(**int** a, **int** b);

}

**Class**

**public** **class** Add **implements** Calculation2{

**public** **int** cal2(**int** a, **int** b)

{

**return** a+b;

}

}

9. Understand diff between WAR, EAR, JAR file?

JAR file allows Java Runtime Environment (JRE) to deploy an entire application including the classes and related resources in a single request. WAR file allows testing and deploying web applications easily while EAR file allows deploying different modules onto an application server simultaneously.

10. How can you tell the technology by seeing URL?

Yes. The source code's file extensions and URLs can tell you what type of platform the website is built on.

11. Is java 100% Object oriented language?

Java is not a pure object-oriented language because it supports Primitive datatype such as int, byte, long? etc, to be used, which are not objects.

12. Write codes for all oops concepts?

Abstraction

**package** OOPS;

**abstract** **class** Bank{

**abstract** **int** getInterestRate();

}

**class** Citi **extends** Bank{

**int** getInterestRate(){**return** 7;}

}

**class** HSBC **extends** Bank{

**int** getInterestRate(){**return** 6;}

}

**class** MainAbstraction{

**public** **static** **void** main(String args[]){

Bank b;

b = **new** Citi ();

System.***out***.println("Citi Rate of Interest is: "+b.getInterestRate()+"%");

b = **new** HSBC ();

System.***out***.println("HSBC Rate of Interest is: "+b.getInterestRate()+"%");

}

}

Inheritance

**package** OOPS;

**class** Doctor {

**void** Doctor\_Details() {

System.***out***.println("Doctor Details are :-");

System.***out***.println("Dr Rajan");

System.***out***.println("Dr Sasi");

}

}

**class** Surgeon **extends** Doctor {

**void** Surgeon\_Details() {

System.***out***.println("Surgen Details are :-");

System.***out***.println("Dr Gopi");

System.***out***.println("Dr Soman");

}

}

**public** **class** HospitalInheritance {

**public** **static** **void** main(String args[]) {

Surgeon s = **new** Surgeon();

s.Doctor\_Details();

s.Surgeon\_Details();

}

}

Encapsulation

**package** OOPS;

**class** Encapsulate {

**private** String Name;

**private** **int** Roll;

**private** **int** Age;

**public** **int** getAge() { **return** Age; }

**public** String getName() { **return** Name; }

**public** **int** getRoll() { **return** Roll; }

**public** **void** setAge(**int** newAge) { Age = newAge; }

**public** **void** setName(String newName)

{

Name = newName;

}

**public** **void** setRoll(**int** newRoll) { Roll = newRoll; }

}

**public** **class** TestEncapsulation {

**public** **static** **void** main(String[] args)

{

Encapsulate obj = **new** Encapsulate();

obj.setName("Govind");

obj.setAge(22);

obj.setRoll(39);

System.***out***.println("Geek's name: " + obj.getName());

System.***out***.println("Geek's age: " + obj.getAge());

System.***out***.println("Geek's roll: " + obj.getRoll());

}

}

Polynorphism

**package** OOPS;

**class** Animal {

**public** **void** animalSound() {

System.***out***.println("The animal makes a sound");

}

}

**class** Pig **extends** Animal {

**public** **void** animalSound() {

System.***out***.println("The pig says: wee wee");

}

}

**class** Dog **extends** Animal {

**public** **void** animalSound() {

System.***out***.println("The dog says: bow wow");

}

}

**class** MainPolymorphism {

**public** **static** **void** main(String[] args) {

Animal myAnimal = **new** Animal();

Animal myPig = **new** Pig();

Animal myDog = **new** Dog();

myAnimal.animalSound();

myPig.animalSound();

myDog.animalSound();

}

}