ISL ENGINEERING COLLEGE

Chandrayangutta, Bandlaguda, Hyderabad - 500 005.
INTERNAL ASSESSMENT EXAMINATION FOR B.E. II YEAR

BRANCH : CSE YEAR: II SECTION : A
SUBJECT NAME: Shareef ROLL.NO: 160519733009
SIGNATURE OF STUDENT: [illegible] SIGNATURE OF PARENTS: S. Nayyer Fatima

Q. NO.	1	2	3	4	5	6	Total Marks
Marks Obtained							

PART-A

1) Explain the use of super keyword in JAVA

Sol: The super Key word in java is a reference variable which is used to refer immediate parent class object.

Usage of JAVA Super Keyword

→ Super can be used to refer immediate parent class instance variable.

→ Super can be used to invoke immediate parent class method.

→ Super() can be used to invoke immediate parent class constructor.

Roll. No : 160519733009 Signature of Student : [illegible] Date 23/6/21

2) why JAVA is architecture neutral?

Sol → Programmer face a problem that if they write a program today, there is no guarantee that it will run tomorrow, even on the same machine.

→ To solve this issue, the java designers made several hard decisions in the Java language & the java virtual machine.

→ Their goal was "write once; run anywhere, any-time, forever". To a great extent, this goal was accomplished. by making java architecture neutral.

3, What is the need of an interface?

Sol Interface is helpful in,

→ Achieveing total abstraction.

→ Since java does not support multiple ~~inlary~~ inheritance in case of class, but by using interface it can be achieve multiple inheritance.

Shareef 16051973300 9 [illegible]

→ It is also used to achieve loose coupling.

→ Interface are used to implement abstraction..

PART-B

4) Explain creating & using packages in Java with example program.

Sol: → Packages are containers for classes that are used to keep the class name space compartmentalized.

→ Packages are stored in a hierarchical manner & are explicitly imported into new class definitions

Creating packages

→ To create a package, simply include a package command with some name as the first statement in a java source file.

→ Any classes declared within that file will belong to the specified package.

→ The package statement defines a name space in which classes are stored.

General form:-

package pkg;

For example:

package mypackage;

→ Java uses file system directories to store packages.

→ More than one file can include the same package statement. The package statement simply specifies to which package the classes defined in a file belong.

→ We can ~~c~~ create a hierarchy of packages.

General form:-

```
package pkg1[.pkg2[.pakg3]];
```

→ A package hierarchy must be reflected in the file system of your java development system.

for example

```
package java.awt.image; // Java\awt\ image.
```

Exomple Program:

```
package MyPack;
class Balance {
      String name;
      double bal;
      Balance (~~St~~String n, double b) {
            this.name = n;
            this. bal = b;
      }
      void show() {
            if(bal<0)
                  System.out.print("--> ");
            System.out.print(name + ": $" + bal);
      }
}
Class  Account Balance{
      public static void main(String args[ ]) {
            Balance current [ ]= new Balance[3];
            current[0]= new Balance("K.J. F
            -ielding", 123.23);
```

```
        current[2]= new Balance("Tom Jackson", -12.33);
        for(int i=0; i<3; i++)
            current[i].show();
    }
}
```

→ Call this file AccountBalance.java & put it in a directory called MyPack.

Q5) Differentiate between method overloading & method overriding in Java.

Sol

Method overloading	method overriding.
i, Method overloading is used to increase the readability of the program ii) Overloading is performed within class.	i, method overriding is used to provide the specific implementation of the method that is already provided by its super class. ii, overriding occurs in 2 classes. that have inheritance relationship.

Shareef 160319733009 [illegible]

iii, Return type can be same or different, but you must have to change the parameters	iii, Return type must be same or co-variant
iv, Parameter must be different	iv, parameters must be same

v, ex:
class overlading {
static int add (int a, int b){
return a+b;
}
static int add (int a, int b, int c){
    return a+b+c;
}
	v, ex:
	class Animal {
	    void eat () {
	      System.out.println("eat...");
	    }
	}
	class Dog extends Animal {
	    void eat () {
	      System.out.println("eat...");
	    }
	}