NEW HORIZON COLLEGE OF ENGINEERING

DEPARTMENT OF INFORMATION SCIENCE AND ENGINEERING

EXPERT TALK SERIES -EVEN SEM (2018-2019)

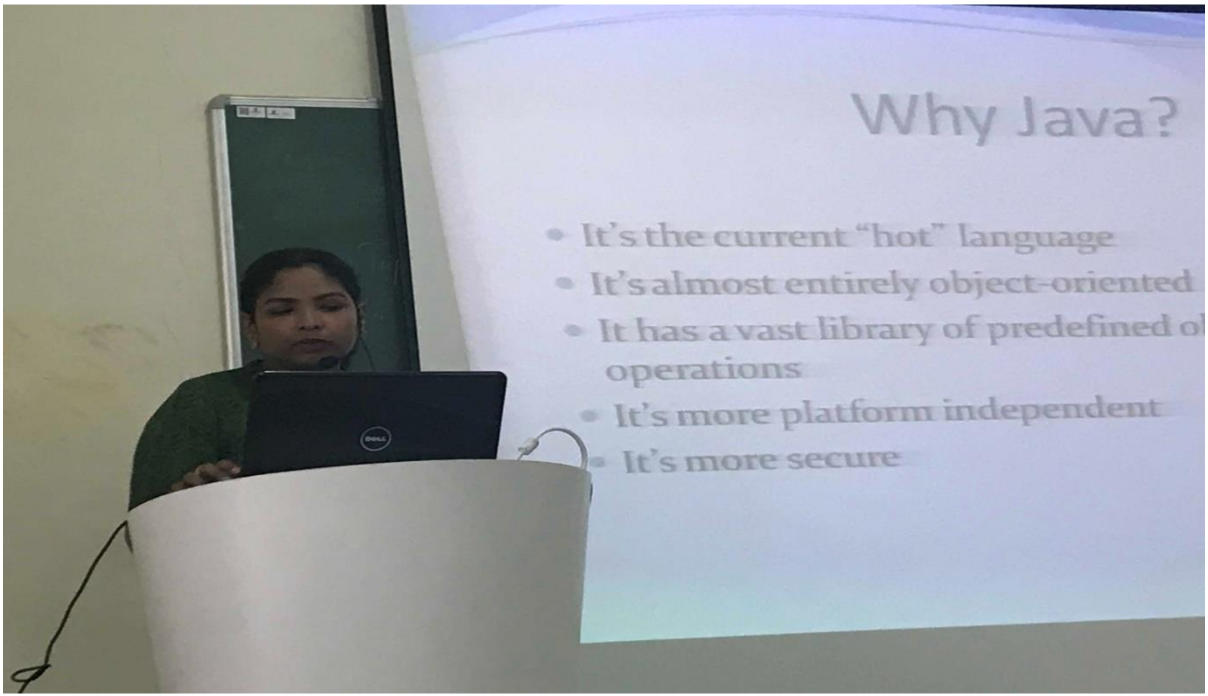
Subject: Object Oriented Programming **Expert name: Ms. Sushma**

Audience: IV A&B **Date: 14-04-2019(11:00AM-12:00AM)**

The department of Information Science and Engineering has conducted Expert talk on the topic **“Object Oriented Programming”** for the 4th semester students on 14th APRIL, 2019 under the supervision of ISE Head of the department, Dr. Anandhi R J at ISE department. The expert speaker, Ms. Sushma was invited to conduct the same.

- Speaker is multi skilled professional with 7+ years of experience in the IT industry specialized with OS, Data Structures, Data Science, and Java.

<u>VARIOUS SESSIONS THROUGHT THE PROGRAM:</u>



Why Java?

- It's the current "hot" language
- It's almost entirely object-oriented
- It has a vast library of predefined o operations
- It's more platform independent
- It's more secure

Why Java?
current "hot" language
ost entirely object-oriented
a vast library of predefined objects and
ns
platform independent
ecure
Java

Expert talk on Object Oriented Programming taken by Ms. Sushma

TOPICS COVERED:

- Introduction to Object Oriented Programming
- Features
- OOP Languages
- Design Patterns

❖ INTRODUCTION TO OBJECT ORIENTED PROGRAMMING

- Object-oriented programming (OOP) is a programming paradigm based on the concept of "objects", which can contain data, in the form of fields (often known as attributes), and code, in the form of procedures (often known as methods). A feature of objects is an object's procedures that can access and often modify the data fields of the object with which they are associated (objects have a notion of "this" or "self"). In OOP, computer programs are designed by making them out of objects that interact with one another. OOP languages are diverse, but the most popular ones are class-based, meaning that objects are instances of classes, which also determine their types.

- Many of the most widely used programming languages (such as C++, Object Pascal, Java, Python, etc.) are multi-paradigm and they support object-oriented programming to a greater or lesser degree, typically in combination with imperative, procedural programming. Significant object-oriented languages include Java, C++, C#, Python,PHP, JavaScript, Ruby, Perl, Object Pascal, Objective-C, Dart, Swift, Scala, Common Lisp, and Smalltalk.

❖ FEATURES

- Class-based vs prototype-based

In class-based languages the classes are defined beforehand and the objects are instantiated based on the classes. If two objects apple and orange are instantiated from the class Fruit, they are inherently fruits and it is guaranteed that you may handle them in the same way; e.g. a programmer can expect the existence of the same attributes such as color or sugar content or is ripe.

- Dynamic dispatch/message passing

It is the responsibility of the object, not any external code, to select the procedural code to execute in response to a method call, typically by looking up the method at run time in a table associated with the object. This feature is known as dynamic dispatch, and distinguishes an object from an abstract data type (or module), which has a fixed (static) implementation of the operations for all instances. If the call variability relies on more than the single type of the object on which it is called (i.e. at least one other parameter object is involved in the method choice), one speaks of multiple dispatch.

- Encapsulation

Encapsulation is an object-oriented programming concept that binds together the data and functions that manipulate the data, and that keeps both safe from outside interference and misuse. Data encapsulation led to the important OOP concept of data hiding.

If a class does not allow calling code to access internal object data and permits access through methods only, this is a strong form of abstraction or information hiding known as encapsulation.

- Polymorphism

Subtyping - a form of polymorphism - is when calling code can be agnostic as to which class in the supported hierarchy it is operating on - the parent class or one of its descendants. Meanwhile, the same operation name among objects in an inheritance hierarchy may behave differently.

❖ OOP LANGUAGES

Simula (1967) is generally accepted as being the first language with the primary features of an object-oriented language. It was created for making simulation programs, in which what

came to be called objects were the most important information
representation. Smalltalk (1972 to 1980) is another early example, and the one with which much of the theory of OOP was developed. Concerning the degree of object orientation, the following distinctions can be made:

- Languages called "pure" OO languages, because everything in them is treated consistently as an object, from primitives such as characters and punctuation, all the way up to whole classes, prototypes, blocks, modules, etc. They were designed specifically to facilitate, even enforce, OO methods.
  Examples: Python, Ruby, Scala, Smalltalk, Eiffel, Emerald,[26] JADE, Self.
- Languages designed mainly for OO programming, but with some procedural elements. Examples: Java, C++, C#, Delphi/Object Pascal, VB.NET.
- Languages that are historically procedural languages, but have been extended with some OO features. Examples: PHP, Perl, Visual Basic (derived from BASIC), MATLAB, COBOL 2002, Fortran 2003, ABAP, Ada 95, Pascal.
- Languages with most of the features of objects (classes, methods, inheritance), but in a distinctly original form. Examples: Oberon (Oberon-1 or Oberon-2).

❖ DESIGN PATTERNS

Challenges of object-oriented design are addressed by several approaches. Most common is known as the design patterns codified by Gamma et al.. More broadly, the term "design patterns" can be used to refer to any general, repeatable, solution pattern to a commonly occurring problem in software design. Some of these commonly occurring problems have implications and solutions particular to object-oriented development.

The outcome of this program is that the students were provided good knowledge about the object oriented programming concepts.

List of Participants

USN	NAME
1NH17IS001	JAMUNA.A
1NH17IS002	ABHINAV ANAND
1NH17IS004	ADITYA KOKANAY
1NH17IS005	AISWARYA V KUMAR
1NH17IS006	AKASH.K.R
1NH17IS007	AKEPATI SASSANK GOPAL REDDY
1NH17IS008	AKHILA S
1NH17IS009	AKSHAY S PRATHAP
1NH17IS011	PAVEL ANUP
1NH17IS012	ANEESH MOHAN KUMAR
1NH17IS013	ANITHA.B
1NH17IS015	ANUSHA K
1NH17IS016	APURBA BHATTACHARJEE
1NH17IS017	AYUSH ANAND SAHU
1NH17IS018	AYUSH SHARMA

1NH17IS019	BHOOMIKA K C
1NH17IS020	BHUMIKA.V
1NH17IS021	BISHAL KUMAR SAH
1NH17IS022	BRUNDA S G
1NH17IS023	HARSHA VARDHAN.C.R
1NH17IS026	CHETAN YG
1NH17IS027	CHINMAYA KUMAR NAYAK
1NH17IS028	DARSHAN CHANDRA SHEKAR SHETTY
1NH17IS029	DEEPAK KUMAR SAH
1NH17IS030	DEEPTHI P
1NH17IS031	DHANUSH R
1NH17IS032	DHARANI
1NH17IS034	DISHA SINGH
1NH17IS035	DIVYA SHREE M
1NH17IS036	EDWIN JOSHUA JOHN
1NH17IS038	GIRISH R
1NH17IS039	HAMSA P O
1NH17IS040	HARINI V N
1NH17IS042	HELEN HEPHZIBAH. D
1NH17IS043	HIMANSHU BHATT
1NH17IS044	HITESH SUHAS
1NH17IS045	JOICY CASTILINO
1NH17IS046	K N BHANU PRIYA
1NH17IS047	KAKI RAVI TEJA
1NH17IS048	KIRTI DEVI
1NH17IS049	KSHITIJ RAJ
1NH17IS050	KUSHALA R
1NH17IS051	MALAVIKA N
1NH17IS052	MANOJ R
1NH17IS053	MERVIN SHIBU GEORGE
1NH17IS055	MOHAMMED OWEZ
1NH17IS056	MONISHA K
1NH17IS057	N MANVITHA REDDY
1NH17IS059	NETHAN SHAIK
1NH17IS060	N G DIVYA
1NH17IS061	NIRANJAN YADAV
1NH17IS062	NIVEDHA S
1NH17IS063	P KUMAR SATEESH
1NH17IS064	P VISHNUVARDHAN REDDY
1NH17IS067	POOJA M.SAJJAN
1NH17IS068	POOJA S SINGH
1NH17IS069	PRAJWAL G

1NH17IS070	PRANOY ROY
1NH17IS071	PRASANNA BHAT
1NH17IS072	PREETHI S
1NH17IS073	PURAB SHREENIWAS. A
1NH17IS075	R RANJITHA
1NH17IS076	R S SATHVIK REDDY
1NH17IS077	RAAHUL NARAYANA REDDY
1NH17IS078	RACHANA M S