Introduction to OOPs

"Characteristics of Object Oriented Programming Languages"

Fundamentals of OOPs

Shakirullah Waseeb
shakir.waseeb@gmail.com

Nangarhar University

September 27, 2017

Agenda

1. Objects and Classes
2. Inheritance, Reusability and New Data Types
3. Polymorphism and Overloading
4. Data abstraction and Encapsulation
5. What's next?
6. Questions and Discussion

Objects

- Approaching a programming problem in object-oriented language, require how to divide it into objects rather than functions
- A close match between real word objects and the objects in the programming sense
- How to find these matches?
  - **Physical Objects**
  - **Elements in Computer Environment**
  - **Data-Structures**
  - **Human Entities**
  - **Simulation Components in Computer Games**
  - **Collections**

Classes

- Objects are instances of classes
- What does this mean?
- Thus a class serve as a plan, or template; specifying which data and functions should be included in instance of given class
- A class doesn't create any objects

Inheritance

- The idea of classes leads to the idea of inheritance
- Concept of classes divided into subclasses
- Eg. Religion Class can be further divided into Islam, Christianity, Buddhism, Communism and so on. The Matter class can be further divided into Solid, Liquid, Gas, and Plasma
- **Principle:** each sub-class shares common characteristics with the class from which it inherits
- *b*ase and derived classes
- Example on board

Reusability

- The use of a written, debugged, and compiled class by programmers is called reusability
- The idea of inheritance provides and important extension to the idea of reusability
- How? Take existing class and, without modifying it, add additional features and capabilities to it, by deriving a new class from it


New Data Types

- Objects facilitate to construct new data types
- Example: Assume you want to add two points
  new_position = origin + next_position
- Creating a class that incorporates two values for each point, and creating *new_positioin, origin, and next_position* as objects, we in fact create new data type


Polymorphism and Overloading

- The '=' and '+' operators performed operation on simple data types
- How do the '=' and '+'operators know how to operate on objects?
- We can define new behaviors for these operators
- Using operators and functions in different ways, depending on what they are operating on, is called polymorphism
- **Polymorphism**: one thing with several distinct forms
- **Operator overloading**: giving the capability to existing operators to operate on new data type

Data abstraction and Encapsulation

- **Encapsulation**: wrapping up of data and functions into a single unit called class is known as encapsulation
- **Data abstraction**: the act of representing essential features without including the background details or explanations

What's next?

C++ Programming Basics
Dealing with Classes and Objects in C++ programming language


Your Turn: Time to hear from you!

1

[1] https://fensafitters.files.wordpress.com/2013/07/3d095.jpg

References


Robert Lafore
Object-Oriented Programming in C++, 4th Edition .
2002.