

Component 4: Introduction to Information and Computer Science

Unit 5: Overview of Programming Languages, Including Basic Programming Concepts Lecture 5

This material was developed by Oregon Health & Science University, funded by the Department of Health and Human
Services, Office of the National Coordinator for Health Information Technology under Award Number 1U24OC000015.

Unit 5 Objectives

- a) Define the purpose of programming languages.
- b) Define the different types of programming languages.
- c) Explain the continuum of programming languages from machine code and assembly languages through scripting languages and high level structured programming languages.
- d) Explain the compiling and interpreting process for computer programs.
- e) Use the following components of programming languages to build a simple program: variables, loops and conditional statements.
- f) Introduce additional programming concepts such as objects and modularity.

Component 4/Unit 5-5

Health IT Workforce Curriculum
Version 2.0/Spring 2011

2

Object Oriented Programming

- Object Oriented Programming (OOP) is a paradigm
- Very popular today
 - C++, C#, Java, Python, Ruby
- Supports software engineering principles
- Graphical User Interface (GUI) programming naturally conforms to OOP

Component 4/Unit 5-5

Health IT Workforce Curriculum
Version 2.0/Spring 2011

3

Objects

- Objects have
 - Identity (name)
 - Attributes (instance variables)
 - Behavior (methods)
- Way of organizing code
 - Data and related methods stored together
- Allows for code reuse
 - Modularity
 - Inheritance

Component 4/Unit 5-5

Health IT Workforce Curriculum
Version 2.0/Spring 2011

4

Classes vs. Objects

- Classes are the code definition for objects
- They are the "blueprint"
- Objects are created when the program runs
 - Instantiation
 - Similar to declaring a variable

Component 4/Unit 5-5

Health IT Workforce Curriculum
Version 2.0/Spring 2011

5

Procedural vs. OOP

```
double circleArea(double radius)
{
    return 3.14*radius*radius;
}

class Circle
{
    double radius;
    void setRadius(double rValue)
    {
        radius = rValue;
    }
    double calcArea()
    {
        return 3.14*radius*radius;
    }
}
```

- In class, radius is stored with the calcArea method
- In procedure, radius is passed into calcArea as a parameter
- How would we add circumference calculation?

Component 4/Unit 5-5

Health IT Workforce Curriculum
Version 2.0/Spring 2011

6

OOP Designs

- Programs are designed using tools
- UML (Unified Modeling Language) is very common
- Example for class of BMI Calculator

BMI Calculator
double weight double height double bmi
void setWeight(double wValue) void setHeight(double hValue) void calcBmi() void outputBmi() void outputBmiCategory()

Inheritance

- Inheritance is a powerful feature of OOP
- Classes can inherit methods and instance variables
- Makes coding less redundant
- Allows for polymorphism

BankingAccount
String accountNum double balance
void setAccountNum(double aValue) void setBalance(double bValue) double getBalance() void printAccountInfo()

CheckingAccount
double overdraft
void setOverdraft(double oValue) double getOverdraft()

SavingsAccount
double interestRate
void setInterestRate (double iValue) void accrueInterest()

Child classes inherit all methods and instance variables from parent class

Modularity

- Separation of code into components such as objects
- Non object oriented languages implement modularity
 - Procedures
- Allows for
 - reuse of code
 - maintainability

Encapsulation

- Objects can declare methods and instance variables to be private or public
 - Typically instance variables are private
 - Some (all) methods are public
- Class definition controls
 - Valid ranges for values
 - Rules for setting values, calling methods
 - Details of implementation are hidden
- Interface is public methods & documentation

Unit Summary

- In this unit, you learned
 - The purpose of programming languages
 - Different types of programming languages
 - The compilation/interpreter process
 - Programming language constructs
 - Object Oriented Programming (OOP)
- Gained an understanding
 - How programs are designed and implemented
 - What code looks like
 - What objects are and why they are used
