

2. Holzner, S.(2007). *PHP: The Complete Reference Paperback*, McGraw Hill Education (India).

Additional Resources

1. Boronczyk, T., & Psinas, M. E. (2008). *PHP and MYSQL (Create-Modify-Reuse)*. Wiley India Private Limited.
2. Welling, L., & Thompson, L. (2008). *PHP and MySQL Web Development*. 4th edition. Addition Paperback, Addison-Wesley Professional.
3. Nixon, R. (2014). *Learning PHP, MySQL, JavaScript, CSS & HTML5*. 3rd edition. Paperback, O'reilly Media
4. Sklar, D., & Trachtenberg, A., (2014). *PHP Cookbook: Solutions & Examples for PHP Programmers*. 2nd edition. O'reilly Media

Course Teaching Learning Process

- Use of ICT tools in conjunction with traditional class-room teaching methods
- Interactive sessions
- Class discussions

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Static and dynamic websites, form handling, database connectivity.

Programming in Python (BHCS19B) Skill-Enhancement Elective Course - (SEC)

Credit: 06

Course Objective

This course is designed to introduce the student to the basics of programming using Python. The course covers the topics essential for developing well documented modular programs using different instructions and built-in data structures available in Python.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Develop, document, and debug modular python programs to solve computational problems.
2. Select a suitable programming construct and data structure for a situation.
3. Use built-in strings, lists, sets, tuples and dictionary in applications.
4. Define classes and use them in applications.
5. Use files for I/O operations.

Detailed Syllabus

Unit 1

Introduction to Programming using Python: Structure of a Python Program, Functions, Interpreter shell, Indentation. Identifiers and keywords, Literals, Strings, Basic operators (Arithmetic operator, Relational operator, Logical or Boolean operator, Assignment Operator, Bit wise operator).

Unit 2

Building blocks of Python: Standard libraries in Python, notion of class, object and method.

Unit 3

Creating Python Programs: Input and Output Statements, Control statements:-branching, looping, Exit function, break, continue and pass, mutable and immutable structures. Testing and debugging a program

Unit 4

Built-in data structures: Strings, lists, Sets, Tuples and Dictionary and associated operations. Basic searching and sorting methods using iteration and recursion.

Unit 5

Visualization using 2D and 3D graphics: Visualization using graphical objects like Point, Line, Histogram, Sine and Cosine Curve, 3D objects

Unit 6

Exception Handling and File Handling: Reading and writing text and structured files, Errors and Exceptions.

Practical

1. Execution of expressions involving arithmetic, relational, logical, and bitwise operators

in the shell window of Python IDLE.

2. Write a Python function to produce the outputs such as:

a) *
 * * *
 * * * * *
 * * *
 *

(b) 1
 232
 34543
 4567654
 567898765

3. Write a Python program to illustrate the various functions of the “Math” module.

4. Write a function that takes the lengths of three sides: **side1**, **side2** and **side3** of the triangle as the input from the user using **input** function and return the area of the triangle as the output. Also, assert that sum of the length of any two sides is greater than the third side.
5. Consider a showroom of electronic products, where there are various salesmen. Each salesman is given a commission of 5%, depending on the sales made per month. In case the sale done is less than 50000, then the salesman is not given any commission. Write a function to calculate total sales of a salesman in a month, commission and remarks for the salesman. Sales done by each salesman per week is to be provided as input. Assign remarks according to the following criteria:
 - Excellent: Sales ≥ 80000
 - Good: Sales ≥ 60000 and < 80000
 - Average: Sales ≥ 40000 and < 60000
 - Work Hard: Sales < 40000
6. Write a Python function that takes a number as an input from the user and computes its factorial.
7. Write a Python function to return nth terms of Fibonacci sequence
8. Write a function that takes a number with two or more digits as an input and finds its reverse and computes the sum of its digits.
9. Write a function that takes two numbers as input parameters and returns their least common multiple and highest common factor.
10. Write a function that takes a number as an input and determine whether it is prime or not.
11. Write a function that finds the sum of the n terms of the following series:
 - a) $1 - x^2/2! + x^4/4! - x^6/6! + \dots x^n/n!$

- b) $1 + x^2/2! + x^4/4! + x^6/6! + \dots x^n/n!$
12. Write a Python function that takes two strings as an input from the user and counts the number of matching characters in the given pair of strings.
 13. Write a Python function that takes a string as an input from the user and displays its reverse.
 14. Write a Python function that takes a string as an input from the user and determines whether it is palindrome or not.
 15. Write a Python function to calculate the sum and product of two compatible matrices
 16. Write a function that takes a list of numbers as input from the user and produces the corresponding cumulative list where each element in the list present at index i is the sum of elements at index $j \leq i$.
 17. Write a function that takes n as an input and creates a list of n lists such that i th list contains first five multiples of i .
 18. Write a function that takes a sentence as input from the user and calculates the frequency of each letter. Use a variable of dictionary type to maintain the count.
 19. Write a Python function that takes a dictionary of *word:meaning* pairs as an input from the user and creates an inverted dictionary of the form *meaning:list-of-words*.
 20. Usage of Python debugger tool-pydb and PythonTutor.
 21. Implementation of Linear and binary search techniques
 22. Implementation of selection sort, insertion sort, and bubble sort techniques
 23. Write a menu-driven program to create mathematical 3D objects
Curve, Sphere, Cone, Arrow, Ring, Cylinder.
 24. Write a program that makes use of a function to accept a list of n integers and displays a histogram.
 25. Write a program that makes use of a function to display sine, cosine, polynomial and exponential curves.
 26. Write a program that makes use of a function to plot a graph of people with pulse rate p vs. height h . The values of p and h are to be entered by the user.
 27. Write a function that reads a file **file1** and displays the number of words and the number of vowels in the file.
 28. Write a Python function that copies the content of one file to another.
 29. Write a function that reads a file **file1** and copies only alternative lines to another file **file2**. Alternative lines copied should be the odd numbered lines.

References

1. Downey, A.B., (2015), *Think Python—How to think like a Computer Scientist*, 3rd edition. O'Reilly Media.
2. Taneja, S. & Kumar, N., (2017), *Python Programming- A Modular Approach*. Pearson Education.

Additional Resources

1. Brown, M. C. (2001). *The Complete Reference: Python*, McGraw Hill Education.
2. Dromey, R. G. (2006), *How to Solve it by Computer*, Pearson Education.
3. Guttag, J.V.(2016), *Introduction to computation and programming using Python*. MIT Press.
4. Liang, Y.D. (2013), *Introduction to programming using Python*. Pearson Education.

Course Teaching Learning Process

- Use of ICT tools in conjunction with traditional class-room teaching methods
- Interactive sessions
- Class discussions

- Mini projects in the laboratory

Tentative weekly teaching plan is as follows:

Week	Content
1.	Python Programming: An Introduction Structure of a Python program, understanding Python interpreter/Python shell, indentation. Atoms, identifiers and keywords, literals, Python strings, arithmetic operator, relational operator, logical or boolean operator, bit wise operators.
2	Variables and Functions Python standard libraries such as sys and math. Variables and assignment statements. Built-in functions such as input and print.
3-4	Control Structures if conditional statement and for loop, While loop, break, continue, and pass statement, else statement. Infinite loop
5	Functions Function definition and call, default parameter values, keyword arguments, assert statement
6	Strings and Lists Strings-slicing, membership, and built-in functions on strings Lists- list operations.

7.	Mutable object Lists- built-in functions, list comprehension, passing list as arguments, copying list objects.
8	Sets, tuples, and dictionary- associated operations and built-in functions.
9	Testing and Debugging Determining test cases, use of python debugger tool- pydb for debugging
10	Searching and Sorting Linear search, binary search, selection sort, insertion sort, and bubble sort
11	Python 2D and 3D Graphics Visualization using graphical objects like point, line, histogram, sine and cosine curve, 3D objects
12	File Handling Reading and writing text and structured files.
13	Errors and Exceptions Types of errors and exceptions, and exception handling
14	Classes Notion of class, object, and method.

Assessment Methods

Written tests, assignments, quizzes, presentations as announced by the instructor in the class.

Keywords

Python Program, Control structure, Decision making, Functions, Strings, Lists, Dictionary.

Android Programming (BHCS20A) Skill-Enhancement Elective Course - (SEC)

Credit:04

Course Objective

The paper provides an introduction to development of mobile application on android platform. The topics include the Android development environment, activities, fragments, user interfaces, intents, broadcast sender/receivers, services, notifications, SQLite database handling.

Course Learning Outcomes

On successful completion of the course, students will be able to:

1. Describe characteristics of Android operating system
2. Describe components of an android applications
3. Design user interfaces using various widgets, dialog boxes, menus
4. Define interaction among various activities/applications using intents, broadcasting, services
5. Develop Android applications that require database handling

Detailed Syllabus

Unit 1

Introduction: Review to JAVA & OOPS Concepts, History of Android, Introduction to Android Operating Systems, Android Development Tools, Android Architecture, Android components including activities, view and view group, services, content providers, broadcast receivers, intents, parcels, instance state.

Unit 2

User Interface Architecture: application context, intents: explicit intents, returning results from activities, implicit intents, intent filter and intent resolution, and applications of implicit intents, activity life cycle, activity stack, application's priority and its process' states, fragments and its life cycle.

Unit 3

User Interface Design: Layouts, optimizing layout hierarchies, form widgets, text fields, button control, toggle buttons, spinners, images, menu, dialog.

Unit 4

Broadcast receivers, notifications and services: Broadcast sender, receiver, broadcasting events with intents, listening for broadcasts with broadcast receivers, broadcasting ordered intents, broadcasting sticky intents, pending intents, creating notifications, setting and customizing the notification tray UI. Create, start, and stop services, binding services to activities, using asynctasks to manage background processing, handler, looper and runnable



This document was created with the Win2PDF "print to PDF" printer available at <http://www.win2pdf.com>

This version of Win2PDF 10 is for evaluation and non-commercial use only.

This page will not be added after purchasing Win2PDF.

<http://www.win2pdf.com/purchase/>