

The Islamia University of Bahawalpur

Department of CS&IT

Baghdad-ul-Jadeed Campus

Course Outline: Programming Fundamentals **Class:** BSCS 1st **Prerequisites:** None

Course Overview: Computer Programming, Principles of Structured and Modular Programming, Overview of Structured Programming Languages, Algorithms and Problem Solving, Program Development: Analyzing Problem, Designing Algorithm/Solution, Testing Designed Solution, Translating Algorithms into Programs, Fundamental Programming Constructs, Data Types; Basics of Input and Output, Selection and Decision (If, If-Else, Nested If-Else, Switch Statement and Condition Operator), Repetition (While and For Loop, Do-While Loops), Break Statement, Continue Statement, Control Structures, Functions, Arrays, Pointers, Files (Input-Output), Testing & Debugging, Strings, Lists, Tuples, Dictionaries.

Online Link to Class Outline: <https://sites.google.com/view/visiontracker/profun>

Recommended Books (Mandatory to read)

- **Book that covers Mid Term syllabus:** Object Oriented Programming in C++, Robert Lafore, Pearson, 4th edition, Twentieth Impression, 2017 (Chapters 1-5, Chapter 6 (few topics), Chapter 7 (few topics), Chapter 10(few topics) For reference I'll call this book OOPinC++
 - **Book that covers Final Term syllabus:** Python for Everybody: Exploring Data in Python 3 (Chapters: 1-10), For reference I'll call this book PY4E.
- Down load links:** <https://www.py4e.com/book.php> & <https://www.py4e.com/>

Recommended MOOCs

Python for Everybody Specialization <https://www.coursera.org/specializations/python> 1st two courses:

- **Programming for Everybody (Getting Started with Python)** <https://www.coursera.org/learn/python?specialization=python#about> . This course will cover Chapters 1-5 of the textbook "Python for Everybody".
- **Python Data Structures** <https://www.coursera.org/learn/python-data?specialization=python> This course will cover Chapters 6-10 of the textbook "Python for Everybody".
- **Programming Fundamentals:** <https://www.coursera.org/learn/programming-fundamentals/home/info>

Reference MOOCs (not mandatory)

- Webpage that points to many Python related resources: <https://sites.google.com/view/visiontracker/python>
- Learn Python: https://www.codecademy.com/courses/learn-python/lessons/python-syntax/exercises/print-statements?action=resume_content_item
- Pointers, Arrays, Recursion: <https://www.coursera.org/learn/pointers-arrays-recursion>
- Writing, Running, and Fixing Code in C: <https://www.coursera.org/learn/writing-running-fixing-code>
- Introduction to Python: Absolute Beginner: <https://courses.edx.org/courses/course-v1:Microsoft+DEV236x+1T2018/course/>
- Code Yourself! An Introduction to Programming: <https://www.coursera.org/learn/intro-programming>

Reference Books & Materials

Some useful books and links for interested readers (not mandatory):

- Discrete Mathematics and Its Applications, Kenneth H. Rossen, Mc-Graw Hill, 7th edition,
- *C How to Program*, Paul Deitel and Harvey Deitel, Prentice Hall; 7th edition (March 4, 2012)
- *Programming in C*, Stephen G. Kochan, Addison-Wesley Professional; 4th edition (September 25, 2013). ISBN-10: 0321776410
- *C++ How to Programme*, Paul Deitel and Harvey Deitel, Prentice Hall; 9th edition (February, 2013)

Evaluation Criteria

Eligibility to sit in the examinations: 80% attendance (as per university examination rules)	
Evaluation Type	Marks
Mid Term Examination	30
Final Term Examination	50
Session Evaluation:	
Attendance	5
Behavior & Class Participation	5
Assignments	10
-----	-----
Total	20
Total	100

The Seven Steps to Solve a Problem

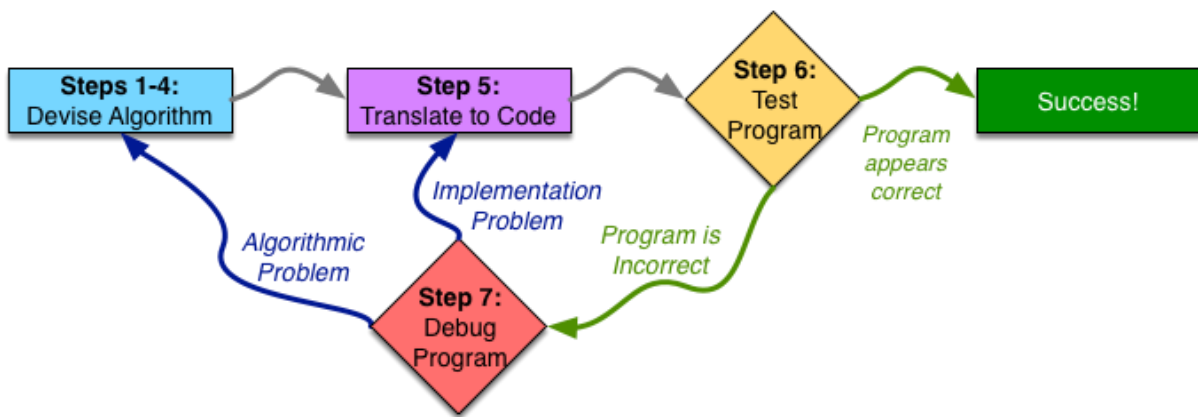


Figure 1. High-level overview of the process of writing a program

Figure Source [<https://www.coursera.org/learn/programming-fundamentals/home/info>]

Planning (Steps 1-4): A programmer starts by devising the algorithm for the task/problem in hand. We split this planning phase into four steps. At these four steps, the programmer should have a complete plan for the task at hand—and be convinced that the plan is a good one:

- **Work an Example Yourself,**
- **Write Down What You Just Did,**
- **Generalize Your Steps,**
- **Test Your Algorithm** (Dry run).

Translate to Code (Step 5): Translating the plan into code in the programming language (C/C++, Python, Java). Initially, translation to code will go slowly, as you will be unfamiliar with the syntax, likely needing to look up the specific details often.

Translation is fairly straightforward, even for beginners, because we have done all the actual problem-solving tasks in planning phase (Steps 1-4).

Algorithm can have some complex steps, it's OK!. We will turn a complex step into its own separate programming task and repeat this seven step programming process on it.

Testing the Code (Step 6): Try to uncover the errors in algorithm/implementation (code).

Debug Program (Step 7): If the programmer finds errors in the program, then it requires debugging: Finding out the cause of the error, and fixing it.

The programmer may need to return to the algorithm design steps (if the error lies in the algorithm) or to translation to code (if the error lies in the implementation) to correct the error. The programmer then repeats all of the later steps. At some point, the programmer completes enough test cases with no errors to become convinced that her program is correct. Note that, no amount of testing can guarantee that the program is correct. Instead, more testing increases the programmer's confidence that the code is correct. We will discuss testing and debugging in detail.

Week#	Lecture Breakup	Activity/Assignments
1.	Introduction of class philosophy and course material. Introduction to Massive open online courses platform (MOOCs). Seven Steps of Solving Problem. Chapter 2 OOP in C++: C++ Programming Basics	Activity 1: Install Integrated development Environment (IDE) Code::Blocks from: http://www.codeblocks.org/downloads/26 or http://www.codeblocks.org/ Assignment 1: Ch 2 Solve Questions & Exercises
2.	Chapter 2 OOP in C++ (Contd.) Chapter 3 OOP in C++: Loops and Decisions	Activity 2: Discuss Ch 2 Questions & Exercises Assignment 2: Ch 3 Solve Questions & Exercises
3.	Chapter 3 OOP in C++ (Contd.) Chapter 4 OOP in C++: Structures	Activity 3: Discuss Ch 3 Questions & Exercises Assignment 3: Ch 4 Solve Questions & Exercises
4.	Chapter 5 OOP in C++: Functions	Activity 4: Discuss Ch 4 Questions & Exercises Assignment 4: Ch 5 Solve Questions & Exercises
5.	Chapter 5 OOP in C++: Functions (Contd.) Chapter 6 OOP in C++: Objects and Classes (Selected Pages)	Activity 5: Discuss Ch 5 Questions & Exercises Assignment 5: Ch 5 Solve Questions & Exercises (Contd.)
6.	Chapter 7 OOP in C++: Arrays and Strings (Selected pages) Introduction to Algorithms, Properties, Basic Algorithms	Activity 6: See Examples of Discrete Mathematics Section 3.1. Assignment 6: Ch 7 Solve Questions & Exercises
7.	Chapter 7 OOP in C++ (Cont.) Chapter 10 OOP in C++: Pointers <u>Quiz:</u> Based on Assignments 10-15	Activity 7: Converting algorithms in Section 3.1. to Pseudo code Assignment 7: Ch 10 Questions & Exercises
8.	Chapter 12 OOP in C++: Streams and Files (Selected Pages) Principles of Structured and Modular Programming. Overview of Structured Programming	Activity 8: Discuss Ch 10, Questions & Exercises
MID TERM EXAMINATION		
9.	Chapter 1 PY4E: Why should you learn to write programs?	Activity 9a: Make an account on following MOOCs platforms and search related courses to programming fundamentals: www.coursera.org , www.edx.org , www.codeacademy.com Activity 9b: Explore Python for Everybody Specialization https://www.coursera.org/specializations/python and Enroll into first course of the specialization for free using Audit option https://www.coursera.org/learn/python/home/info Activity 9c: Install Anaconda on your home system/laptop (See Week 2 of the MOOC) Activity 9d: Complete Week 1-3 of the MOOC Assignment 9: Solve Ch 1 PY4E exercise on folder.
10.	Chapter 2 PY4E: Variables and Expressions, and statements	Activity 10a: Discuss Chapter 1 Exercise Activity 10b: Complete Week 4 of the MOOC Assignment 10: Exercises Chapter 2 of PY4E
11.	Chapter 3 PY4E: Conditional Execution	Activity 11a: Discuss Chapter 2 Exercise Activity 11b: Complete Week 5 of the MOOC Assignment 11: Exercises Chapter 3 of PY4E
12.	Chapter 4 PY4E: Functions	Activity 12a: Discuss Chapter 3 Exercise Activity 12b: Complete Week 6 of the MOOC Assignment 12: Exercises Chapter 4 PY4E
13.	Chapter 5 PY4E: Loops and Iterations	Activity 13a: Discuss Chapter 4 Exercise Activity 13b: Complete Week 7 of the MOOC Assignment 13: Exercises Chapter 5 of PY4E
14.	Chapter 6 PY4E: Strings Chapter 7 PY4E: Files <u>Quiz:</u> Based on Assignments 1-5	Activity 14a: Discuss Chapter 5 Exercise Activity 14b: Enroll into 2 nd course of the specialization for free using Audit option. https://www.coursera.org/learn/python-data/home/info Activity 14c: Complete Weeks1 & 2 of the MOOC Assignment 15: Exercises Chapter 6 of PY4E
15.	Chapter 7 PY4E (Cond.): Files Chapter 8 PY4E: Lists Chapter 9 PY4E: Dictionaries	Activity 15a: Discuss Chapter 6 Exercise Activity 15b: Complete Weeks 3 & 4 of the MOOC Assignment 15: Exercises Chapters 7-8 of book PY4E
16.	Chapter 9 PY4E (Cont.): Dictionaries Chapter 10 PY4E: Tuples	Activity 16a: Discuss Chapters 8-10 Exercises Activity 16b: Complete Weeks 5-6 of the MOOC
Final Term Examination		