

T-104 2022

Course Specification

Course Title: Programming Fundamentals
Course Code: 181CIS-3
Program: Programming and Database
Department: Computer department
College: Applied college
Institution: Najran university
Version: T -104 2022
Last Revision Date: 7 Aug 2023





Table of Contents:

Content	Page
A. General Information about the course	3
 Teaching mode (mark all that apply) Contact Hours (based on the academic semester) 	3
B. Course Learning Outcomes, Teaching Strategies and Assessment Methods	4
C. Course Content	5
D. Student Assessment Activities	6
E. Learning Resources and Facilities	6
1. References and Learning Resources	6
2. Required Facilities and Equipment	6
F. Assessment of Course Quality	7
G. Specification Approval Data	7





A. General information about the course:

Course Identification

1. Credit hours: 3(2+1)

2. Course type

- a. University \Box College \Box Department \boxtimes Track \Box Others \Box
- b. Required ⊠ Elective□

3. Level/year at which this course is offered:

^{۱ nd} Level

4. Course general Description

This course is about Computer Programming Fundamentals using python programming language. It includes Understand fundamental terms and definitions, Understand Python's logic and structure, literals and variables, operators and data types, Input/Output console operations, decisions and flow. This course is essential for obtaining the professional certificate PCEP (PCEP-30-02), and updated periodically according to the certificate exam

5. Pre-requirements for this course (if any):

None

6. Co- requirements for this course (if any):

None

7. Course Main Objective(s)

This course is intended to:

- Provide students with a good understanding of concepts and terminology related to the Computer Programming using Python Language.
- Enable students to translate the real computing problems into a programms that solve it.
- Develop the programming skills and experience needed to write Python language programs.
- Enable students to communicate with others effectively to solve real computing Problems.

1. Teaching mode (mark all that apply)

No	Mode of Instruction	Contact Hours	Percentage
1.	Traditional classroom	4 hours per week	90%
2.	E-learning		٥%
3.	HybridTraditional classroomE-learning		
4.	Distance learning		100%





2. Contact Hours (based on the academic semester)

No	Activity	Contact Hours
1.	Lectures	30
2.	Laboratory/Studio	3.
3.	Field	
4.	Tutorial	
5.	Others (specify)	
	Total	60

B. Course Learning Outcomes (CLOs), Teaching Strategies and Assessment Methods

Code	Course Learning Outcomes	Code of CLOs aligned with program	Teaching Strategies	Assessment Methods
1.0	Knowledge and understanding			
1.1	Define the basic concepts of programming language, algorithm, flowchart, and program structure.	K1	Lecturers Labs	Exam Quiz Assignment
1.2	Understand the language syntax, statements, and derived data types	КЗ	Lecturers Labs	Exam Quiz Assignment
1.3	Write python programs	К3		
2.0	Skills		-	
2.1	2.1 Design programs to S1 solve problems	S1	Lecturers Labs	Exam Quiz Assignment
2.2	Write flowcharts to understand the program modules	S1	Lecturers Labs	Exam Presentation
	fix errors in python programs	S1		
3.0	Values, autonomy, ar	nd responsibility		
3.1	Demonstrate projectsV3and assignments in teamwork for designing and developing python programs		Project Small group report	Presentation





Computer Programming and Python Fundamentals: (18% of exam – 7 exam items) Understand fundamental terms and definitions 6 Understand fundamental terms and definitions • interpreting and the interpreter, compilation and the compiler 6 Understand Python's logic and structure • keywords 4 • instructions 4 • indentation • comments Introduce literals and variables into code and use different numeral systems 10 • binary, octal, decimal, and hexadecimal numeral systems • scientific notation 3 • Strings 10 • implementing PEP-8 recommendation • unarring conventions 10 • numeric operators: ** / % // + - • string operators: ** / % // + - 9 4 • binary, octal, decimal, and, or 9 9 • binary, octal, decimal, and hexadecimal numeral systems • variables 10 • naming conventions • implementing PEP-8 recommendation 10 • binary and binary operators: *+ • % // + - • string operators: ** / % // + 9 • bitwise operators: ** / % // + • sosignment and shortcut operators • unary and binary operators 9 • bitwise operators: ** / % // + costing • Boolean operators: *	C. (Course Content	
exam - 7 exam items) Understand fundamental terms and definitions 6 1 Understand fundamental terms and definitions 6 2 interpreting and the interpreter, compilation and the compiler 6 2 Lexis, syntax, and semantics 4 3 instructions 4 1 indentation 6 2 indentation 6 2 indentation 6 3 comments 4 10 binary, octal, decimal, and hexadecimal numeral systems 10 3 Strings 10 4 implementing PEP-8 recommendation 10 5 naming conventions 10 6 implementing PEP-8 recommendation 9 7 esting operators: ** + /% // + - string operators: ** + 9 string operators: ** + 9 9 bitwise operators: (* * < 9 9 bitwise operators: (* * < 9 10 bitwise operators: (* * < 9 10 bitwise operators: (* * < 9 10 bitwise operators: (* * < <th>No</th> <th>List of Topics</th> <th>Contact Hours</th>	No	List of Topics	Contact Hours
1 • interpreting and the interpreter, compilation and the compiler 6 2 lexis, syntax, and semantics Understand Python's logic and structure 4 2 instructions 4 2 instructions 4 3 Introduce literals and variables into code and use different numeral systems 10 3 Boolean, integers, floating-point numbers 10 3 Strings 10 4 implementing PEP-8 recommendation 10 5 Priorities and binding 10 6 numeric operators: ** * /% // + - string operators: ** * /% // + - 6 String operators: ** * /% // + - string operators: ** * /% // + - 9 Boolean operators and data types adequate to the problem 9 9 Introduce preators: ** * /% // + - string operators: ** * /% // + - 9 Boolean operators: ** * /% // + - string operators: ** * /% // + - 9 Introduce operators: ** * /% // + - string operators: ** * /% // + - 9 Boolean operators: not, and, or Boolean operators: ** * /% // + - 9 Boolean operators: not, and, or Boolean expressions <			
• lexis, syntax, and semantics Understand Python's logic and structure • keywords 2 • instructions 4 2 • indentation • comments 3 Introduce literals and variables into code and use different numeral systems • scientific notation 3 Introduce literals and variables into code and use different numeral systems • scientific notation 3 • Strings 10 4 • binary, octal, decimal, and hexadecimal numeral systems • variables • naming conventions • implementing PEP-8 recommendation 10 Choose operators and data types adequate to the problem • numeric operators: * * /% // + - string operators: * * • assignment and shortcut operators • priorities and binding 9 • bitwise operators: ~ & ^ <<>> • Boolean expressions • relational operators: (== ! > >= < <=)		Understand fundamental terms and definitions	
Understand Python's logic and structure42instructions42instructions42indentation4indentationcomments3Introduce literals and variables into code and use different numeral systems103Boolean, integers, floating-point numbers • scientific notation103Strings • naming conventions • implementing PEP-8 recommendation104Choose operators and data types adequate to the problem • numeric operators: ** + • assignment and shortcut operators94Boolean expressions • priorities and binding • bitwise operators: c & ^ << >> • Boolean expressions • relational operators (== != >>< <=) • the accuracy of floating-point numbers • type casting95Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items)67Make decisions and branch the flow with the if instruction • multiple conditional statements: if, if-else, if-elif, if-elif-else • multiple conditional statements: if, if-else, if-elif, if-elif-else • multiple conditional statements:12	1		6
• keywords 4 2 • instructions 4 2 • instructions 4 • indentation • comments 3 • Boolean, integers, floating-point numbers • scientific notation 3 • Strings 10 • 4 • binary, octal, decimal, and hexadecimal numeral systems 10 • > variables 10 • naming conventions • implementing PEP-8 recommendation Choose operators and data types adequate to the problem • numeric operators: ** * /% // + - • string operators: ** /% // + - • ssignment and shortcut operators • unary and binary operators • unary and binary operators • unary and binary operators • unary and binary operators • unary and binary operators • Boolean operators: ** * /% // * - • bitwise operators: ** * /% // * • Boolean operators: ** * /% • priorities and binding • bitwise operators: ** * /% /* •		· · · · ·	
2 instructions indentation comments 4 3 Introduce literals and variables into code and use different mumeral systems Boolean, integers, floating-point numbers scientific notation Strings scientific notation Strings binary, octal, decimal, and hexadecimal numeral systems variables naming conventions implementing PEP-8 recommendation 10 4 Choose operators and data types adequate to the problem numeric operators: ** * /% // + - string operators: ** / % // + - string operators: ** * /% // + - string operators: ^* & ^ <<>> Boolean operators: not, and, or Boolean operators (= l = >> = <<=) the accuracy of floating-point numbers type casting for the sep= and end= keyword parameters the int() and float() functions the sep= and end= keyword parameters the int() and float() functions conditional statements: if, if-else, if-elif, if-elife-else multiple conditional statements 12<td></td><td></td><td></td>			
• indentation • comments• Introduce literals and variables into code and use different numeral systems • Boolean, integers, floating-point numbers 			
Introduce literals and variables into code and use different numeral systemsIntroduce literals and variables into code and use different numeral systemsIntroduce literals and variables into code and use different numeral systemsBoolean, integers, floating-point numbersImage: StringsBoolean, integers, floating-point numbers10Image: StringsStrings10Image: StringsImage: Network of Strings10Image: StringsImage: Network of Strings10Image: StringsImage: Network of Strings10Image: String conventionsImage: Network of String Conventions10Image: String conventionsImage: Network of String Conventions10Image: String conventionsImage: Network of String Conventions10Image: String conventors: *** / % // + -Image: String conventors: *** / % // + -Image: String conventorsImage: String conventors: ** a assignment and shortcut operators:Image: String conventors9Image: String conventors: ** a assignment and shortcut operators:Image: String conventors9Image: String conventors: ** a assignment and shortcut operatorsImage: String conventors9Image: String conventors: ** bitwise operators: Image: String conventorsImage: String conventors9Image: String conventors: ** bitwise conventors: ** bitwise conventorsImage: String conventors6Image: String conventors: ** bitwise conventors: *	2		4
Introduce literals and variables into code and use different numeral systems Boolean, integers, floating-point numbers 10 3 Boolean, integers, floating-point numbers 10 3 Strings 10 3 Strings 10 4 binary, octal, decimal, and hexadecimal numeral systems 10 5 implementing PEP-8 recommendation 10 6 Choose operators and data types adequate to the problem 10 9 string operators: ** * /% // + - string operators: ** * /% // + - 9 string operators: ** * /% // + - string operators: ** * /% // + - 9 Boolean operators: ** * /% // + - string operators: ** * /% // + - 9 Boolean operators: ** * /% // + - 9 9 Boolean operators: ** * /% // + - 9 9 Boolean operators: ** * /% // + - 9 9 Boolean operators: ** * /% // + - 9 9 Boolean operators: ** * /% // + - 9 9 Boolean operators: ** * /% // * 9 9 Boolean expressions 10 10 the accuracy of floating-point numbers 10 10 <td></td> <td></td> <td></td>			
numeral systems 			
Boolean, integers, floating-point numbersImage: Scientific notation3Strings103Strings104Strings104Choose operators and data types adequate to the problem • numeric operators: ** * /% // + - • string operators: ** * /% // + - 94Priorm Ingut/Output console operators • relational operators (== !=>>= <<=) • the accuracy of floating-point numbers • type casting95Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items)66Control Flow - Conditional Blocks and Loops: (29% of exam - s exam items)12			
3• scientific notation • Strings103• Strings • binary, octal, decimal, and hexadecimal numeral systems • variables • naming conventions • implementing PEP-8 recommendation104Choose operators and data types adequate to the problem • numeric operators: ** * /% // + - • string operators: ** * /% // * - • string operators: ** * % // * - • string operators: ** * * % // * - • Boolean operators: ** * * % // * - • Boolean expressions • relational operators (== != >>< <=) • the accuracy of floating-point numbers • type casting95Perform Input/Output console operations • the print() and input() functions • the sep= and end= keyword parameters • the int() and float() functions • the sep= and end= keyword parameters • the int() and float() functions • the int() and float() functions • conditional Blocks and Loops: (29% of exam - s exam items)6 <t< td=""><td></td><td>-</td><td></td></t<>		-	
3• Strings • binary, octal, decimal, and hexadecimal numeral systems • variables • naming conventions • implementing PEP-8 recommendation104Choose operators and data types adequate to the problem • numeric operators: ** * /% // + - • string operators: ** * * * assignment and shortcut operators • unary and binary operators • priorities and binding • bitwise operators: ** * /% // + - • string operators: ** * * * assignment and shortcut operators • priorities and binding • bitwise operators: ** * / * < • Boolean expressions • relational operators (== != >>= < • type casting95Perform Input/Output console operations • the print() and input() functions • the sep= and end= keyword parameters • the int() and float() functions • the sep= and end= keyword parameters • the int() and float() functions • the sep = and end= keyword parameters • the int() and float() functions • conditional statements: if, if-else, if-elif, if-elif-else • multiple conditional statements12 <td></td> <td></td> <td></td>			
 binary, octal, decimal, and hexadecimal numeral systems variables naming conventions implementing PEP-8 recommendation Choose operators and data types adequate to the problem numeric operators: ** * /% // + - string operators: * + assignment and shortcut operators unary and binary operators priorities and binding bitwise operators: not, and, or Boolean operators (==!=>>= <<=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the sep= and end= keyword parameters the int() and float() functions the sep= and end= keyword parameters the int() and float() functions the accisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-else multiple conditional statements 	_		
 variables naming conventions 	3	-	10
Image: state of the set of t			
• implementing PEP-8 recommendationChoose operators and data types adequate to the problem• numeric operators: ** * / % // + -• string operators: * +• assignment and shortcut operators• unary and binary operators• unary and binary operators• priorities and binding• bitwise operators: ~ & ^ << >>• Boolean operators: not, and, or• Boolean expressions• relational operators (== != >>= <<=)			
Choose operators and data types adequate to the problem• numeric operators: ** * /% // + -• string operators: * +• assignment and shortcut operators• unary and binary operators• unary and binary operators• priorities and binding• bitwise operators: ~ & ^ << >>• Boolean operators: not, and, or• Boolean expressions• relational operators (== != > >= <<=)		-	
 numeric operators: ** * /% // + - string operators: * + assignment and shortcut operators unary and binary operators priorities and binding bitwise operators: ~ & ^ << >> Boolean operators: not, and, or Boolean expressions relational operators (== != >> = <<=) the accuracy of floating-point numbers type casting the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the sep= and end= keyword parameters the int() and float() functions the semitters the int() and float() functions 			
4• string operators: * + • assignment and shortcut operators • unary and binary operators • priorities and binding • bitwise operators: ~ & ^ << >> • Boolean operators: not, and, or • Boolean expressions • relational operators (== != >> = < <=) • the accuracy of floating-point numbers • type casting95Perform Input/Output console operations • the print() and input() functions • the sep= and end= keyword parameters • the int() and float() functions • the int() and statements: if, if-else, if-elif, if-elif-else • multiple conditional statements6			
 assignment and shortcut operators unary and binary operators unary and binary operators priorities and binding bitwise operators: ~ & ^ << >> Boolean operators: not, and, or Boolean expressions relational operators (== != > > = < <=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the examitems) Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elife-else multiple conditional statements 			
 unary and binary operators priorities and binding bitwise operators: ~& ^ << >> Boolean operators: not, and, or Boolean expressions relational operators (== != >> = < =) the accuracy of floating-point numbers type casting 5 Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the int() and float() functions the int() and float() functions the int() and blocks and Loops: (29% of exam - 8 exam items) 7 Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 			
 priorities and binding bitwise operators: ~ & ^ << >> Boolean operators: not, and, or Boolean expressions relational operators (== != >>= <<=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions 			
 bitwise operators: ~ & ^ << >> Boolean operators: ~ & ^ << >> Boolean operators: not, and, or Boolean expressions relational operators (== != >>= < =) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions 6 Control Flow – Conditional Blocks and Loops: (29% of exam – 8 exam items) 7 Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 			
 Boolean operators: not, and, or Boolean expressions relational operators (== != >>= <<=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the int() and float() functions the int() and float() functions the int() and blocks and Loops: (29% of exam - 8 exam items) 7 Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 	4		9
 Boolean expressions relational operators (== != >>= <<=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the int() and float() functions Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items) Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 			
 relational operators (== !=>>= <=) the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions the int() and float() functions the int() and float() functions Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 		•	
 the accuracy of floating-point numbers type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items) Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 		·	
 type casting type casting Perform Input/Output console operations the print() and input() functions the sep= and end= keyword parameters the int() and float() functions 6 Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items) Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements multiple conditional statements the integration of the statements the integratements field of the statements field of the statements			
Perform Input/Output console operations 6 5 the print() and input() functions 6 6 the sep= and end= keyword parameters 6 6 Control Flow – Conditional Blocks and Loops: (29% of exam – 8 exam items) 6 7 Make decisions and branch the flow with the if instruction 12			
5• the print() and input() functions • the sep= and end= keyword parameters • the int() and float() functions66Control Flow - Conditional Blocks and Loops: (29% of exam - 8 exam items)67Make decisions and branch the flow with the if instruction • conditional statements: if, if-else, if-elif, if-elif-else • multiple conditional statements12			
5 the sep= and end= keyword parameters 6 6 the int() and float() functions 6 6 Control Flow – Conditional Blocks and Loops: (29% of exam – 8 exam items) 6 7 Make decisions and branch the flow with the if instruction 12 7 • conditional statements: if, if-else, if-elif, if-elif-else 12		· · ·	
• the int() and float() functions • 6 Control Flow – Conditional Blocks and Loops: (29% of exam – 8 exam items) 7 Make decisions and branch the flow with the if instruction • conditional statements: if, if-else, if-elif, if-elif-else 12	5		6
6 Control Flow – Conditional Blocks and Loops: (29% of exam – 8 exam items) 7 Make decisions and branch the flow with the if instruction conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 12			
8 exam items) 7 6 8 exam items) 7 9		· · ·	
7 Make decisions and branch the flow with the if instruction • conditional statements: if, if-else, if-elif, if-elif-else • multiple conditional statements	6	• •	
 conditional statements: if, if-else, if-elif, if-elif-else multiple conditional statements 		•	
multiple conditional statements			
	7		12





3	 Perform different types of iterations the pass instruction building loops with while, for, range(), and in iterating through sequences expanding loops with while-else and for-else nesting loops and conditional statements controlling loop execution with break and continue 	12
		60

D. Students Assessment Activities

No	Assessment Activities *	Assessment timing (in week no)	Percentage of Total Assessment Score
1.	Midterm exam	8	20%
2.	Homework's	From 3 to 14	10%
3.	Practical exam	15	20%
4	Final exam	16	50%

*Assessment Activities (i.e., Written test, oral test, oral presentation, group project, essay, etc.)

E. Learning Resources and Facilities

1. References and Learning Resources

Essential References	Python Essentials - Part 1 (Basics) https://edube.org/study/pe1
Supportive References	The Python Language Reference <u>The Python Language Reference</u> — Python 3.11.3 documentation
Electronic Materials	https://www.python.org/doc/
Other Learning Materials	

2. Required Facilities and equipment

Items	Resources
Accommodation (Classrooms, laboratories, demonstration rooms/labs, etc.)	Classroom with a suitable size for students
Technology Resources (AV, data show, Smart Board, software, etc.)	Whiteboard/projector
Other Resources (Specify, e.g. if specific laboratory equipment is required, list requirements or attach a list)	None



8



F. Assessment of Course Quality

Assessment Areas/Issues	Assessor	Assessment Methods
Effectiveness of teaching	Student	Direct: Questioners
Effectiveness of students assessment	Teacher Audit and review committees	Direct: CW & HW Exercises and short quizzes Projects Mid and final paper exams.
Quality of learning resources	Teachers and course description committees	Indirect: Benchmarking Self-evaluation External evaluation
The extent to which CLOs have been achieved	Teacher	Direct: Measuring the learning outcomes
Others		

Other

Assessor (Students, Faculty, Program Leaders, Peer Reviewer, Others (specify) Assessment Methods (Direct, Indirect)

G. Specification Approval Data

COUNCIL /COMMITTEE REFERENCE NO. DATE

