New Course Form

For help filling out the form press F1 or look at the bottom of the screen. For additional instructions, see Course Form Instructions.	
Type of Action	**New Course – Course not previously offered.**

1. Catalog Prefix and Number*: **CIT 144**

2. Course Title: **Python I**

   Pilot Course Information:

   Are you requesting pilot status only at this time? Yes ☐ No ☒
   Implementation Date?

   Are you requesting pilot status in addition to regular approval process? Yes ☒ No ☐
   Implementation date? Fall or Spring? Spring Year? 2013

   Has this course been previously approved as pilot status Yes ☐ No ☒

3. Justification for requested action.
   A Python programming course is needed for students in Computer & Information Technologies and Geographic Information Systems. This course has been offered successfully as CIT299, Special Topics.

4. Submitting Entity: Curriculum Committee:
   Or College: **Bluegrass Community and Technical College**

5. Person(s) Primarily Responsible for Proposal **(Complete item only if course is not part of a curriculum package. Verify that members are still current and active prior to submission.)**:

Name	Teaching Area	College
James Kolasa	**Computer & Information Technologies**	**Bluegrass Community and Technical College**

Involvement of Others (Identify Individuals):

6. System Office Staff:

7. Others:

8. Is this course offered at other colleges? ☐ Yes ☒ No
   If yes, have they been involved in the development of this course? ☐ Yes ☐ No ☐ N/A

9. Is this course duplicative or similar to other courses offered by KCTCS? ☐ Yes ☒ No
   If yes, Justification:

10. Credit / Contact Hours: 10a. Semester Credit Hours: Minimum **3** Maximum **3**
    10b. Semester Contact Hours: **45** If lab, etc., ratio of contact hours to credit hours. (See contact/ credit hour ratio chart)

11. Grading Basis: ☒ Letter Grades ☐ Pass/Fail ☐ Letter Grades/No GPA

12. Repeat for additional credit: ☐ Yes ☒ No

If yes, complete the following:

Total credit earned in course: Total completions:

13. Open Entry – Open Exit: ☐ Yes ☒ No

14a. Components (Check all components that require scheduling. For each component that is checked, enter the credit hours and contact hours for each component that is checked.):

	Component	Credit Hours	Contact Hours		Component	Credit Hours	Contact Hours
☒	Lecture	**3**	**45**	☐	Practicum		
☐	Laboratory			☐	Co-Op		
☐	Clinical			☐	Discussion		

OR

14b. Integrated Components (If components are integrated, and only one component (lecture or lab) needs scheduling, rather than both lecture and lab, complete this section.)

Lecture/ Lab	Lecture Credit	Lecture Contact	Lab Credit	Lab Contact

15. Requisites:

Pre-requisite	☒	Yes	☐	No	If yes, list:	**CIT120 or Consent of Instructor**
Co-requisite	☐	Yes	☒	No	If yes, list:	
Pre-requisite or Co-requisite	☐	Yes	☒	No	If yes, list:	

16. Implementation Term (Course scheduled to begin, ex. Fall 2012): **Fall 2013**

17. Proposed Course Description (Course description as it will appear in the catalog. Begin each statement with a verb.):

Emphasizing clarity over complexity, Python has become one of the most heavily used languages for a multitude of applications including scripting, web applications, and general purpose programming. Python runs on nearly every platform and is easily integrated with .NET, .COM, and Java objects. Python is also open-source and easy to learn.

Course Proposal Rationale:

18. Will this course be a part of an approved curriculum/curricula? ☒ Yes ☐ No

If yes, which curriculum/curricula? **Computer & Information Technologies**
(Submit a New Curriculum or Revision Form)

Python will be added to the list of Level I Programming Languages on the next CIT curriculum revision.

Course Competencies and Delivery:

19. Proposed Course Competencies/Student Outcomes *(If part of an organized curriculum, how does it relate to program competencies /outcomes?* ***Begin statement with a capital letter and end with a period.****):*

Upon completion of this course, the student can:

1. Design, develop, compile, debug, test, run, and document programs in the Python language.
2. Design and develop programs using operators and assignments.
3. Design and develop programs using a variety of data types.
4. Demonstrate the input and output processes in Python.
5. Design and develop programs using sequence, selection, and repetition structures.
6. Design and develop programs using arrays, lists, and tuples.
7. Design and develop programs using simple GUI components.

8. Demonstrate Python objects.
9. Evaluate and critique effectiveness and efficiency of code.

20. Course Outline *(Two-level outline required. Although courses may have more than two levels, the third level is not necessary.)*

I. Software Development
   A. Write Python code
   B. Using the Python interpreter
   C. Debug Python code
II. Python Fundamentals
   A. Keywords
   B. Data types
   C. Variables
   D. Constants
III. Operators and Assignments
   A. Assignment operators
   B. Arithmetic operators
   C. Relational operators
   D. Logical operators
   E. String operators
IV. Input and Output
   A. Input statements
   B. Output statements
V. Control Structures
   A. Sequence
   B. Selection
   C. Repetition
VI. Modularization
   A. Functions
   B. Parameters
VII. Files and Exceptions
   A. File input/output
   B. Handling exceptions
VIII. Software Objects
   A. Classes and methods
   B. Attributes
   C. Controlling access
   D. Inheritance
   E. Polymorphism
IX. GUI
   A. The root window
   B. Labels
   C. Buttons
   D. Text areas
   E. Layout managers
   F. Check buttons
   G. Radio buttons
X. Evaluation of Programming
   A. Effectiveness of code
   B. Efficiency of code

21. List of experiments/activities *(Courses with components other than lecture. e.g., laboratory, clinical, practicum, etc., must include a* ***sample*** *list of experiment topics or activities. Does not have to be all*

-inclusive.):

22. Indicate sample suggested classroom resources for course (Should not have publishing date greater than five years.)

SUGGESTED LEARNING RESOURCES FOR THIS COURSE

Lambert, Kenneth A. (2012). *Fundamentals of Python: First Programs*. Boston: Course Technology. ISBN-10: 1-111-82270-0 ISBN-13: 978-1-111-82270-5

22. Provide a rationale for using textbook/references older than five years.

n/a

23. May this course be used as an equivalent for other courses? **Yes** **No X**

If yes, please list.

Signatures: Complete and submit a signature page for every proposal.

*The System Office assigns new course numbers. Contact Mary Kleber at Mary.Kleber@kctcs.edu