COURSE DESCRIPTION CARD - SYLLABUS

Course name
Visual Programming for Bioinformaticians [S1Bioinf1>PWIZ]

Course

Field of study
Bioinformatics

Year/Semester
2/4

Area of study (specialization)
–

Profile of study
general academic

Level of study
first-cycle

Course offered in
Polish

Form of study
full-time

Requirements
elective

Number of hours

Lecture
30

Laboratory classes
30

Other
0

Tutorials
0

Projects/seminars
0

Number of credit points

4,00

Coordinators

dr inż. Marcin Radom
marcin.radom@put.poznan.pl

Lecturers

Prerequisites

A student starting this module should have a knowledge of structured and object-oriented programming, should know at least one integrated programming environment (Eclipse, Visual Studio, Code::Blocks, others). Student should have the ability to solve algorithmic problems and to divide the solution into elementary steps.

Course objective

1. To provide students with knowledge of C# language and designing window applications in Visual Studio environment. 2. To acquaint students with the capabilities of the Visual Studio environment in developing applications using Windows Forms and Windows Presentation Foundation (WPF). 3. To teach students how to design graphical interfaces. 4. To introduce new Visual Studio modules and the possibilities of designing in various programming styles. 5. To present known algorithms for computer problems to be implemented using visual application design technique with modern graphical interface.

Course-related learning outcomes

Knowledge:
As a result of the course, the student knows:

1. issues in bioinformatics algorithms, basic and advanced elements of C# language,
2. principles of object-oriented programming in C# language which he/shee can apply to bioinformatics problems.

Skills:
As a result of the course, the student will be able to:
1. design and develop software for bioinformatics applications according to a given specification, using appropriate methods, techniques and tools.

Social competences:
Passing the course means that the student:
1. understands the need for lifelong learning and improving competence due to new discoveries and methods in bioinformatics sciences and the continuous development of programming languages.

Methods for verifying learning outcomes and assessment criteria

Learning outcomes presented above are verified as follows:

Learning outcomes presented above are verified as follows:
Formative assessment:
a) in the field of lectures the verification of the assumed learning outcomes is realized by the answers to the questions concerning the material discussed during the previous lectures,
b) in the field of laboratory, verification of the assumed learning outcomes is realized by:
- evaluation and "defense" of the weekly projects carried out by the students,
- evaluation of knowledge and skills related to the realization of laboratory tasks, for example, through colloquium/quizzes during the semester or oral answers.
Summative assessment:
a) in the scope of lectures, verification of the assumed educational effects is realized by evaluation of knowledge and skills demonstrated on a written colloquium in the form of a multiple-choice test and few open questions. The test consists of 10-15 questions with a total value of 20-30 points distributed depending on the difficulty of the question. Students receive a passing grade after scoring at least half of the points.
b) in the scope of the laboratory, the verification of the learning outcomes is carried out by writing up to two quizzes on the practical applications of the programming issues discussed during the lectures. Students receive a positive mark from the laboratory if they pass at least one test and pass at least 80% of the positively verified weekly projects from the classes.
The activity during the classes is rewarded with additional points, especially for the effectiveness of the application of the knowledge gained while writing the program in a manner exceeding the minimum specified in the specification.
For assigned programs/projects, verification may or may not include: sending the project, describing the project, talking about the sent project, modifying the code of the sent project, talking about modifications to the code of the project, describing modifications to the code of the sent project and (obviously) talking about it. Or others.

Programme content

The lecture program includes topics on C# language, visual programming and selected bioinformatics algorithms.
Laboratory exercises are conducted in the form of fifteen two-hour classes which take place in a computer laboratory. The first class is intended to familiarize students with the rules of using the laboratory and passing the exercises.

Course topics

The lecture program includes topics on C# language, visual programming and selected bioinformatics algorithms. During the lectures students get acquainted with:
- syntax, objects and programming styles in C# language,
- Visual Studio environment,
- Rider environment
- methods of creating window programs within Windows Forms methodology.
Laboratory exercises are conducted in the form of fifteen two-hour classes which take place in a computer

laboratory. The first class is intended to familiarize students with the rules of using the laboratory and passing the exercises. The program of laboratory classes includes the following:
- practicing and consolidating knowledge from lectures on various elements of the C# language,
- writing independent programs to consolidate knowledge from the lectures,
- development of a larger program as part of the laboratory classes to illustrate such concepts as codetransparency, consistent writing style, division of the program into different functional units, etc.,
- exercises on more difficult topics necessary to be mastered, before independently writing programs for in Windows Forms. Larger laboratory projects include bioinformatics topics.

Teaching methods

1. Lecture: multimedia presentation and additional examples given on the blackboard as needed.
2. Laboratory exercises: solving tasks, practical exercises, teamwork.

Bibliography

Basic
Andrew Troelsen, “Pro C# 5.0 and the .NET 4.5 Framework”, 2012.
Additional
-

Breakdown of average student's workload

	Hours	ECTS
Total workload	100	4,00
Classes requiring direct contact with the teacher	60	2,50
Student's own work (literature studies, preparation for laboratory classes/ tutorials, preparation for tests/exam, project preparation)	40	1,50