

**HIGHER EDUCATIONAL ESTABLISHMENT OF UKOOPSPILKA
«POLTAVA UNIVERSITY OF ECONOMICS AND TRADE»**

**EDUCATIONAL AND SCIENTIFIC INSTITUTE OF INTERNATIONAL
EDUCATION**

**THE DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION
TECHNOLOGY**

Allowed to defend

Head of the Department

Olkhovska O.V.
(signature, initials, surname)
2022

**EXPLANATORY NOTE
FOR THE GRADUATE THESIS**

on topic:

**DEVELOPMENT OF SIMULATOR SOFTWARE ON THE TOPIC
"CLASSES IN C++" OF THE DISTANCE LEARNING COURSE
"PROGRAMMING II"**

**from specialty 122 "Computer science"
educational program "Computer Science"
master's degree**

The executor of the work is Nyasha Stanley Chagonda

_____ " ____ " _____ 2022.
(signature)

Scientific supervisor , Ph.D., k.p.s, Oksana Koshova

_____ " ____ " _____ 2022.
(signature)

**HIGHER EDUCATIONAL ESTABLISHMENT OF UKOOSPILKA
"POLTAVA UNIVERSITY OF ECONOMICS AND TRADE"**

I APPROVE

**Head of the department _____ Olkhovska O.V.
" ____ » _____ 2022**

**TASKS AND CALENDAR SCHEDULE
OF EXECUTION OF DIPLOMA THESIS**

**Graduate of higher education in specialty 122 "Computer science"
Educational program "Computer Science"**

Surname, first name, patronymic _ Nyasha Stanley Chagonda

**1. Development of simulator software on the topic "classes in C++" of the
distance learning course "Programming ii"**

approved by the rector's order No. 139-H from 01. 09. 2022

The deadline for the student to submit a thesis « ____ » _____ 2022

2. Source data for the master's thesis: publications on the topic of educational simulators in distance courses in computer science.

3. Content of the explanatory note (list of issues to be developed)

INTRODUCTION

1. PROBLEM STATEMENT

2. INFORMATION REVIEW

2.1. Review of works where the task similar to a theme of work is considered

2.2. Positive aspects of the reviewed works

2.3. Defects in the development of the inspected works

2.4. Necessity and relevance of the topic

3. THEORETICAL PART

3.1. Algorithmization of the problem on the topic of work

3.2. Algorithm description

3.3. Justification of the choice of software for the implementation of the task

4. PRACTICAL PART

4.1. Development of a block diagram to be programmed

4.2. Run manual

4.3. Testing

4.4. Program Description

CONCLUSION

REFERENCES

APPENDIX A

APPENDIX B

4. List of graphic material : 3-4 sheets of block diagrams, other necessary illustrations.

5. Consultants of sections of the master's thesis

Section	Surname, initials, position of consultant	Signature, date	
		issued the task	accepted the task
1. Statement of the problem	<u>Koshova O.</u>		
2. Information review	<u>Koshova O.</u>		
3. Theoretical part	<u>Koshova O.</u>		
4. Practical implementation	<u>Koshova O.</u>		

6. Calendar schedule of master's work

The content of the work	Deadline	Actual performance
1. Introduction		
2. Study of methodological recommendations and standards and report to the manager		
3. Formulation of the problem		
4. Information review of library and Internet sources		
5. Theoretical part		
6. The practical part		
7. Completion of registration		
8. Student report at the department		
9. Editing (if necessary), reviewing		

Issue date of task " 8 " September 2022.

Nyasha Stanley Chagonda a higher education graduate
Scientific supervisor , Ph.D., k.ped.s., Koshova O.

The results of the thesis defense

The thesis was evaluated at

(points, assessment according to the national scale, assessment according to ECTS)

Minutes of the meeting of the EC No. _____ of " _____ " _____ 2022.

Secretary of the EC _____

(signature) (initials and surname)

I approve

Head of the department _____
Ph.D.-k.ph-m.s. O. Olkhovska

Agreed

Supervisor _____
associate professor, Ph.D.-O.Koshova

" _____ " _____ 2022

" _____ " _____ 2022

ABSTRACT

Note: 76 pp., including the main part is 25 pages, 12 sources.

The purpose of the work is to create an algorithm and an application for the distance learning course to be used as training aid on the topic “Classes in C++” in “Programming ii”.

The object of development is a simulator for the distance learning platform on the topic “Classes in C++”.

The methods that were used to solve the problem – the software was developed using Microsoft Visual Studio 2022 IDE and the C++ programming language.

CONTENT

LIST OF SYMBOLS, UNITS, ABBREVIATIONS, TERMS	6
INTRODUCTION	7
1. PROBLEM STATEMENT	9
2. INFORMATION REVIEW	10
2.1. Review of similar works	10
2.2. Positive aspects of the reviewed works	11
2.3. Defects in the development of the inspected works	11
2.4. Necessity and relevance of the topic	11
3. THEORETICAL PART	12
3.1. Algorithmization of the problem on the topic of work	12
3.2. Algorithm description	13
3.3. Justification of the choice of software for the implementation of the task	14
4. PRACTICAL PART	15
4.1. Development of a block diagram to be programmed	15
4.2. Run manual	17
4.3. Testing	27
4.4. Program Description	28
CONCLUSION	29
REFERENCES	30
APPENDIX A	32
APPENDIX B	35

LIST OF SYMBOLS, UNITS, ABBREVIATION AND TERMS

Symbols, units, abbreviations, terms	Explanation of symbols, units, abbreviations, terms
Simulator	a program enabling a computer to execute program.
Class	a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class.
Algorithm	a process or set of rules to be followed in calculations or other problem-solving operations.
C++	an object-oriented programming (OOP) language.
MS Visual Studio IDE	An environment for writing software code.

INTRODUCTION

In our world today, computers are used in many fields such as business, industry, administration, education and in various aspects of our lives, and this is only done by using one of the programming languages.

The purpose of the course project is to develop software elements for the simulator on the topic "Classes in C++" of the distance learning course "Programming II".

The object of the course project is to understand the writing and use of classes in C++ programming language.

The subject of the course project is a software product that implements a simulator on the topic of "Classes" in the C++ programming language.

C++ is a general-purpose programming language created by Bjarne Stroustrup as an extension of the C programming language, or "C with Classes". The language has expanded significantly over time, and modern C++ now has object-oriented, generic, and functional features in addition to facilities for low-level memory manipulation. It is almost always implemented as a compiled language, and many vendors provide C++ compilers, including the Free Software Foundation, LLVM, Microsoft, Intel, Oracle, and IBM, so it is available on many platforms.

C++ was designed with an orientation toward systems programming and embedded, resource-constrained software and large systems, with performance, efficiency, and flexibility of use as its design highlights. C++ has also been found useful in many other contexts, with key strengths being software infrastructure and resource-constrained applications, including desktop applications, video games, servers (e.g. e-commerce, web search, or databases), and performance-critical applications (e.g. telephone switches or space probes).

The C++ programming language has support for string handling, mostly implemented in its standard library. The language standard specifies several string

types, some inherited from C, and some designed to make use of the language's features, such as classes and RAII. The most-used of these is `std::string`.

Using C++ classes to implement a string type offers several benefits of automated memory management and a reduced risk of out-of-bounds accesses, and more intuitive syntax for string comparison and concatenation

To realize the aim of the work we have to implement the following tasks:

- study the corresponding literature and material from the distance course on classes in C++;
- review the corresponding simulators;
- to formulate corresponding problems and to work over theoretical question;
- to create the algorithm of the work of the simulator;
- to choose the language of programming taking into consideration possible implementation of it in the module platform;
- to think over the structure and design of the simulator, to write the program according to the algorithm;
- to check the correctness of the simulator's work.

The work consists of an introduction, the problem statement, information review of similar works, the theoretical part which includes the algorithmization of the topic of the work and the justification of the choice of software for the implementation of the task, the practical part which includes flowcharts. The description of the program and the run manual and a summary of the work.

The length of the explanatory note is 64 pages, of which 29 pages of the main text.

1. PROBLEM STATEMENT

The main objective of the masters' work is to create a training simulator for the topic, "Classes in C++", in order to increase students' access to relevant and practical information about the programming discipline and more importantly, to assist students to gain skills to solve problems within the mathematical discipline.

When designing the simulator, it is required to include source theoretical material about the topic, create an algorithm and to write code based off of that algorithm and to make a clear, understandable user interface of the simulator.

The simulator must be capable of performing the role of a training system with little difficulty incurred and also automatically.

The advantage of using simulator is that only relevant information about the subject is presented to the students and it is relatively easy and convenient to make any changes and to update the material.

The simulator must be easy to access and also easy to use. The simulator must include practical exercises that are relevant to the topic.

Before solving each problem in the simulator, it is necessary to formulate the necessary theoretical questions.

The simulator must also be user friendly by providing automated assistance to the user in the event of mistakes.

Before creating the simulator, it was necessary to conduct research to answer basic questions such as why a simulator is useful and to also review similar projects of this nature, and then followed by the creation of the algorithm and flowcharts for the simulator and last but not least, the user interface. Testing was conducted to ensure that the simulator was working as intended.

The resources used in this project are easily accessible, and the methods are easy to follow. The programming language used, C++, is one of the most well-known and frequently used programming languages in the world.

2. INFORMATION REVIEW

2.1. Review of works where the task similar to a theme of work is considered

There exist a wide range of material on the topic. I reviewed books, articles and web videos on the topic. An article titled “Adaptive Educational Software by Applying Reinforcement Learning” by Abdellah Bennane speaks on adopting educational software with the intention of automating and managing a tutoring system. The article goes into detail about how to use different techniques such as the Bayesian network in order to adapt the system both internal and external conditions in order for it work effectively and efficiently with the users. The article mainly serves as a foundational source of information concerning the conception of intelligent tutoring systems. The solution to the problem of how to effectively develop an automated tutoring simulator is broken down into two steps, the first is that the author needs to organize and structure the teaching environment i.e to organize the source material, and the second part is technical i.e to develop the system with the perspective of the user in mind. The article basically highlights the development of the architecture of a tutoring system. “Advantages, Limitations and Recommendations for online learning during COVID-19 pandemic era” shows the results of a case study conducted from March to April 2020 at University College of Medicine and University College of Dentistry, Lahore. The study checked the practicality of online learning among students and concluded that remote learning is great tool for education and is convenient for students because since increases accessibility to information.

Keywords – distance learning, virtual school environment, c++, tutoring system, reinforcement learning, accessibility, remote learning.

2.2. Positive Aspect of the reviewed works

The reviewed works highlight how to go about creating a distance learning simulator. The reviewed work also show that a simulator is a very important tool for learning and helps with information retention and revision. The reviewed articles managed to explain the history of c++ programming language and simulators in general. Applications are a great tool for distance learning because they increase accessibility to learning materials for students and increase productivity in students who are now more comfortable with using technology based learning platforms.

2.3. Defects in the development of the inspected works

The inspected works failed to provide a step by step guide of how a distance learning simulator can be created from scratch. The reviewed work also did not provide detailed practical examples of how to create an application like this using C++.

2.4. Necessity and relevance of the topic of work.

The topic of work serves to provide a tutoring aid to students who wish to either improve on their current knowledge of the topic of “Classes in c++” or start learning about the topic from scratch with little to no prior knowledge of the topic. The simulator is necessary because it reinforces the concepts that the students would have learnt in class or previously studied on their own and can be used as a quick revision aid. The review of the similar projects showed that there was currently no such simulator that had been previously developed on the topic “Classes in C++”.

3. THEORETICAL PART

3.1. Algorithmization of the problem on the topic of work

3.1.1. Algorithm for constructing a simulator for the distance course on the topic of “classes in C++”

Step 1: The user needs to click the “start” button to start the program.

Go to step 2.

Step 2: Classes and Objects. The user is presented with the definition and example of a Class and an Object. To advance to the next stage the user has to click on the “next” button.

Step 3: Class Methods. The user is presented with the definition and an example of Class Methods. They have click on “next” in order to progress to the next screen.

Step 4: Constructors. The user is presented with a brief definition of Constructors and an example. The user has to click the next button to got to the next screen.

Step 5: Access Specifiers. A description of access specifiers appears on the screen and the user has to click the next button in order to progress to the next step.

Step 6: Encapsulation. The user is presented with the definition and description of Encapsulation.

Step 7: Inheritance. The user is presented with detailed information on inheritance in C++ classes. The user needs to read and understand the information about Inheritance and click on the next button to progress to the next screen,

Step 8: Multilevel Inheritance. The user is presented with the definition and a description of Multilevel Inheritance. The user needs to understand the information and the example provided. After reading and understanding the user should click on the next button in order to advance to the next stage.

Step 9: Multiple Inheritance. The user is presented with a brief description of what Multilevel Inheritance is and an example. To advance to the next stage the user should click on the next button.

Step 10: Inheritance Access. The user is presented with information and an example of Inheritance access. The user should read and understand the material and then click the next button to proceed to the next stage.

Step 11: Polymorphism. The user is presented with a brief description of Polymorphism and a brief example of the syntax. To advance to the next stage the user should click on the next button.

Step 12: Test. The user is presented with multiple choice questions to test their knowledge of the studied material. The program will check individually whether the answers are correct or not. If the answer is correct the program will notify the user that the answer is correct and if not it will notify the user that the answer is wrong. In the end the program will show the user their final result.

The end.

3.2. Algorithm Description

Algorithm.

Step 0. Start the program.

Step 1. Display language selection choice.

Step 2. User makes a choice.

Step 3. Display the first topic and example.

Step 4. User clicks “next” button.

Step 5. The next topic is presented on the screen.

Step 6. User clicks “next” button till they reach the “test” section.

Step 7. Display the test questions.

Step 8. User inputs the answers.

Step 9. The program checks the answers and presents the user with the overall mark.

Step 10. Program ends.

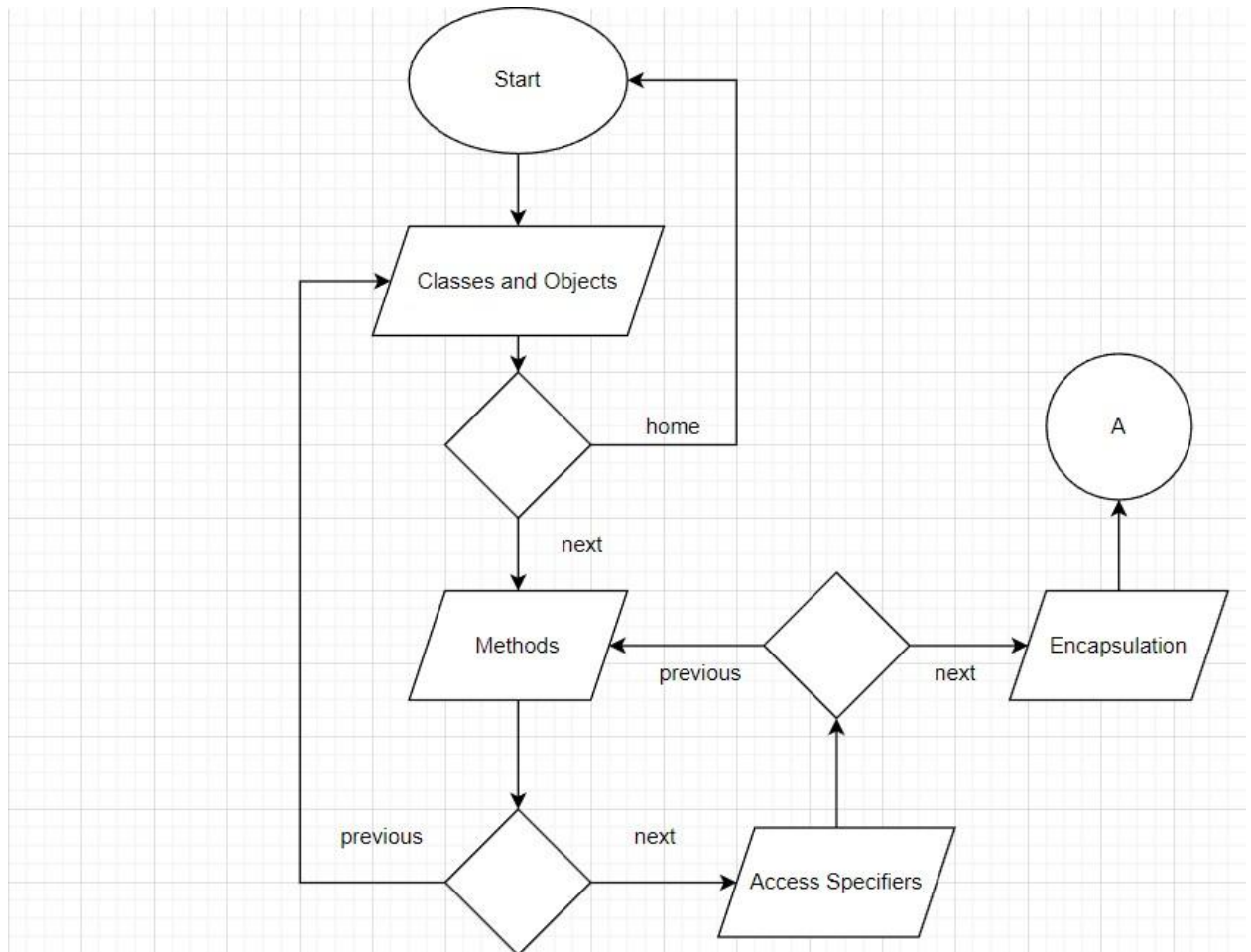
3.3. Justification of the choice of software and programming language for the implementation of the task.

C++ is a general-purpose programming language and runs on lots of platforms like Windows, Linux, Unix, Mac, etc. C++ was chosen because it is easier to learn, and understand than the other low-level languages like binary coding. Even low level programmers and beginners can easily follow the step by step guide to create a simulator of their own. C++ supports inline function and exception handling and also offers platform independence which allows the user to run the same program on different operating systems or interfaces at ease. C++ was also used in this case because it is easy to convert the developed software into a small “exe” file that can then be easily uploaded and downloaded from the internet.

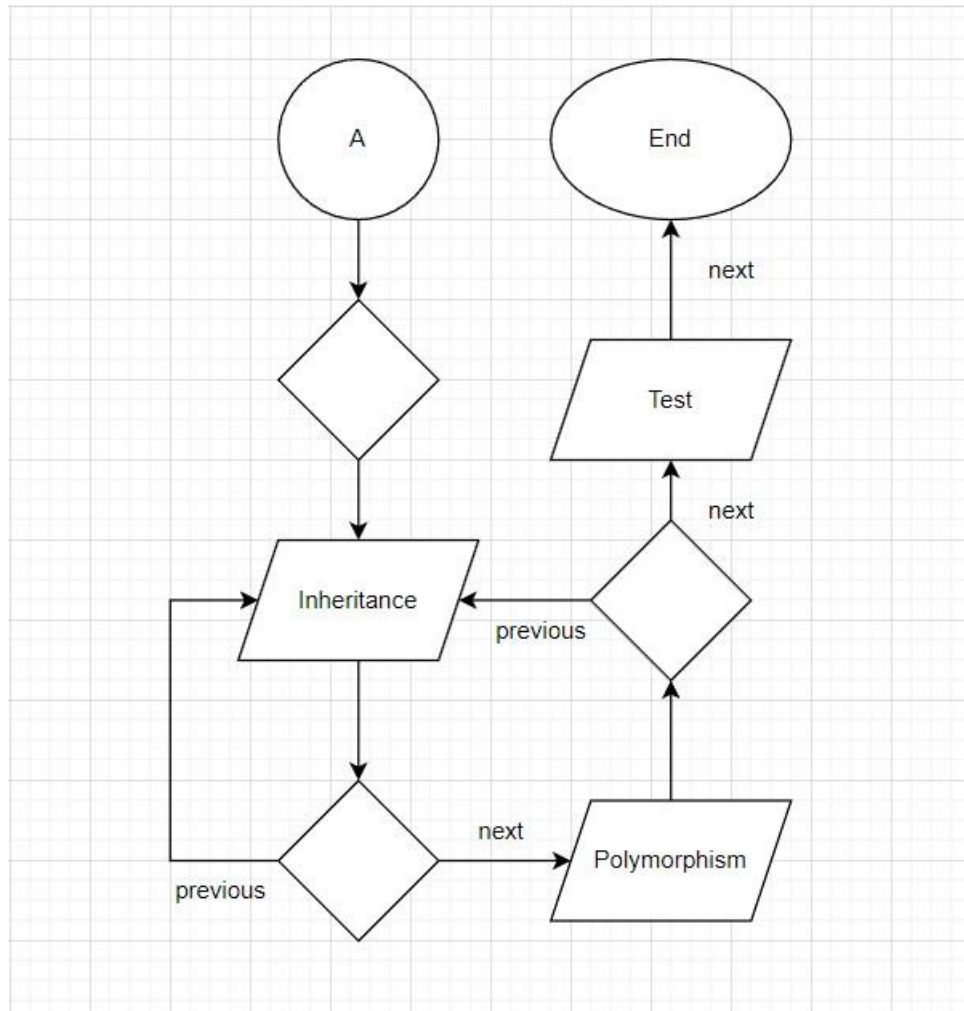
The IDE used to develop the simulator was Microsoft Visual Studio. The IDE was chosen mainly because of its accessibility. Microsoft Visual Studio can easily be found and downloaded on the internet from Microsoft`s official website. As an added bonus the software is free to use for students and is capable of being installed and used on Windows, Mac and Linux operating system. The application does not take a lot of space on your computer and it is easy to install. It is optional for you to setup an account or not to, and Microsoft offers excellent support for the software incase you may encounter some difficulties using the application. There is a vast range of tutorials and teaching documentation available on how to use the application. The application can also be used to develop software using a wide range of programming languages so it is a great tool to have in your arsenal. Errors are easily detectable and solvable when using Microsoft Visual Studio since the application alerts you to the presence of any errors in real time and provide you with possible solutions on how to solve the error and further information on how and why the error occurred in the first place.

4. PRACTICAL PART

4.1. Development of a flowchart to be programmed



Program Flowchart 1.



Program Flowchart 2.

4.2. Run manual

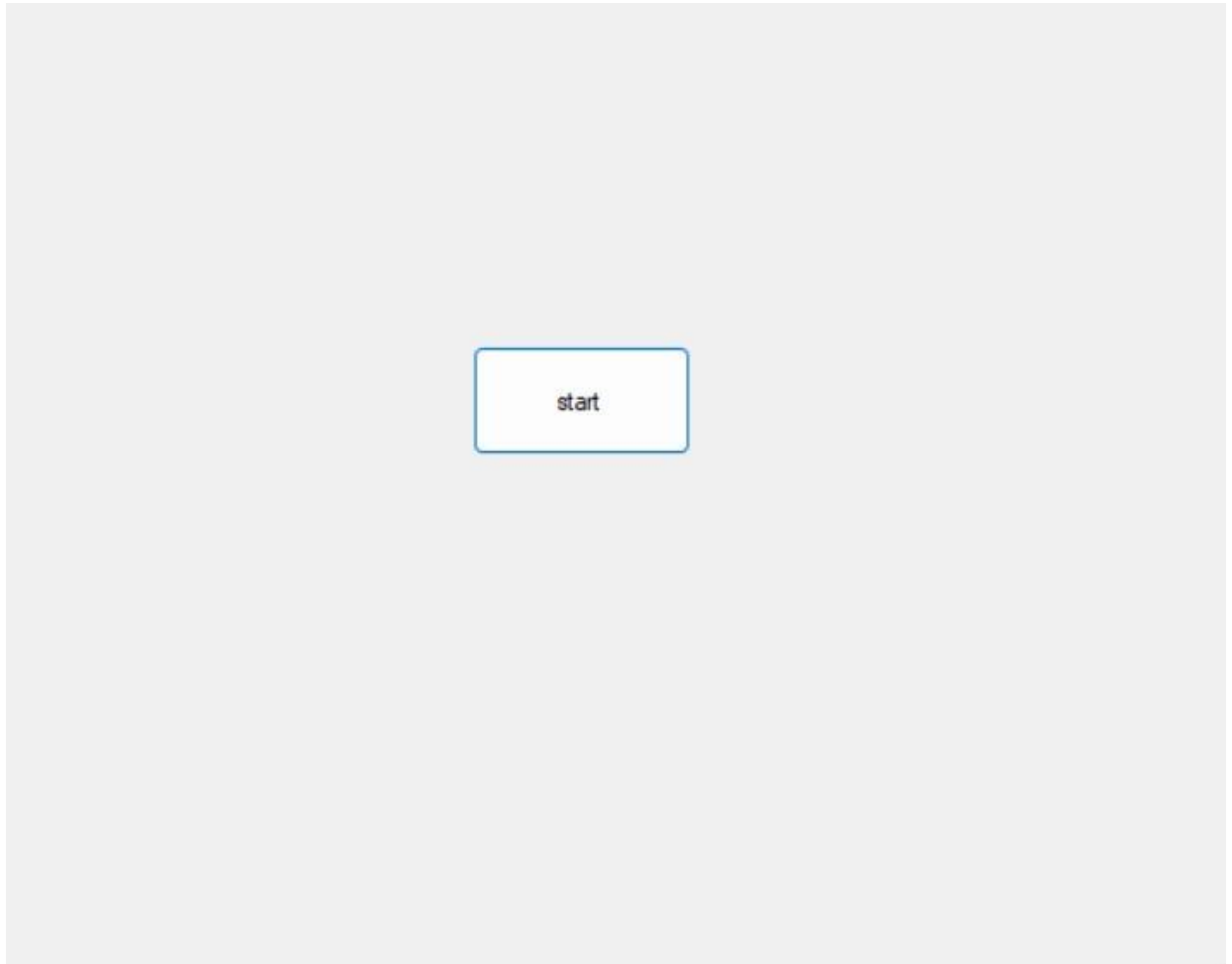


Fig 1. Program Start

To start the program the user needs to click on the “start” button.

Classes and Objects

C++ is an object-oriented programming language.

Everything in C++ is associated with classes and objects, along with its attributes and methods. For example: in real life, a car is an object. The car has attributes, such as weight and color, and methods, such as drive and brake.

Attributes and methods are basically variables and functions that belongs to the class. These are often referred to as "class members".

A class is a user-defined data type that we can use in our program, and it works as an object constructor, or a "blueprint" for creating objects.

Create a Class
To create a class, use the class keyword:

Example
Create a class called "MyClass":

```
class MyClass { // The class
public: // Access specifier
    int myNum; // Attribute (int variable)
    string myString; // Attribute (string variable)
};
```

home next

Fig 2. Classes and Objects

To see the rest of the material the user should scroll down and to go to the next topic, the user needs to press the "next" button. The user can press the "home" button to go to the home screen.

Methods

Methods are functions that belongs to the class.

There are two ways to define functions that belongs to a class:

Inside class definition
Outside class definition

In the following example, we define a function inside the class, and we name it "myMethod".

Note: You access methods just like you access attributes; by creating an object of the class and using the dot syntax (.):

Inside Example

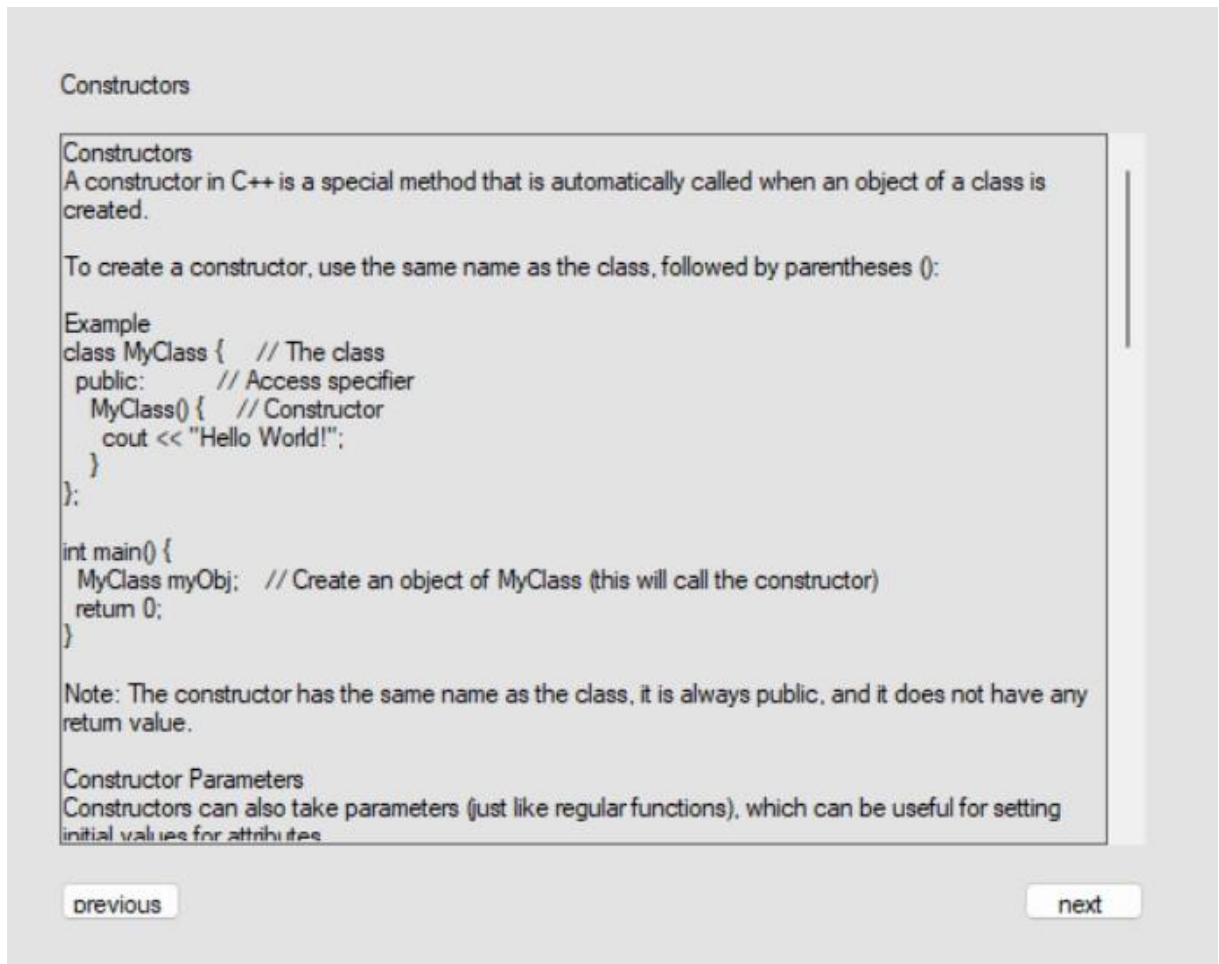
```
class MyClass { // The class
public: // Access specifier
    void myMethod() { // Method/function defined inside the class
        cout << "Hello World!";
    }
};

int main() {
    MyClass myObj; // Create an object of MyClass
    myObj.myMethod(); // Call the method
    return 0;
}
```

previous next

Fig 3. Methods

To see the rest of the material the user should scroll down and to go to the next topic, the user needs to press the “next” button. The user can press the “previous” button to go back to the previous screen.



The screenshot shows a web page with the title "Constructors". The main content is a text box containing the following text:

Constructors
A constructor in C++ is a special method that is automatically called when an object of a class is created.

To create a constructor, use the same name as the class, followed by parentheses ():

Example

```
class MyClass { // The class
public: // Access specifier
    MyClass() { // Constructor
        cout << "Hello World!";
    }
};

int main() {
    MyClass myObj; // Create an object of MyClass (this will call the constructor)
    return 0;
}
```

Note: The constructor has the same name as the class, it is always public, and it does not have any return value.

Constructor Parameters
Constructors can also take parameters (just like regular functions), which can be useful for setting initial values for attributes.

At the bottom of the page, there are two buttons: "previous" on the left and "next" on the right.

Fig 4. Constructors

To see the rest of the material the user should scroll down and to go to the next topic, the user needs to press the “next” button. The user can press the “previous” button to go back to the previous screen.

Access Specifiers

By now, you are quite familiar with the public keyword that appears in all of our class examples:

Example

```
class MyClass { // The class
public:      // Access specifier
  // class members goes here
};
```

The public keyword is an access specifier. Access specifiers define how the members (attributes and methods) of a class can be accessed. In the example above, the members are public - which means that they can be accessed and modified from outside the code.

However, what if we want members to be private and hidden from the outside world?

In C++, there are three access specifiers:

- public - members are accessible from outside the class
- private - members cannot be accessed (or viewed) from outside the class
- protected - members cannot be accessed from outside the class, however, they can be accessed in inherited classes. You will learn more about Inheritance later.

In the following example, we demonstrate the differences between public and private members:

Example

```
class MyClass {
public: // Public access specifier
  int x; // Public attribute
```

previous next

Fig 5. Access Specifiers

To see the rest of the material the user should scroll down and to go to the next topic, the user needs to press the “next” button. The user can press the “previous” button to go back to the previous screen.

Encapsulation

The meaning of Encapsulation, is to make sure that "sensitive" data is hidden from users. To achieve this, you must declare class variables/attributes as private (cannot be accessed from outside the class). If you want others to read or modify the value of a private member, you can provide public get and set methods.

Access Private Members
To access a private attribute, use public "get" and "set" methods:

Example

```
#include <iostream>
using namespace std;

class Employee {
private:
    // Private attribute
    int salary;

public:
    // Setter
    void setSalary(int s) {
        salary = s;
    }
    // Getter
    int getSalary() {
        return salary;
    }
};
```

previous next

Fig 6. Encapsulation

To see the rest of the material the user should scroll down and to go to the next topic, the user needs to press the "next" button. The user can press the "previous" button to go back to the previous screen.

Polymorphism

Polymorphism means "many forms", and it occurs when we have many classes that are related to each other by inheritance.

Like we specified in the previous chapter; Inheritance lets us inherit attributes and methods from another class. Polymorphism uses those methods to perform different tasks. This allows us to perform a single action in different ways.

For example, think of a base class called `Animal` that has a method called `animalSound()`. Derived classes of `Animals` could be `Pigs`, `Cats`, `Dogs`, `Birds` - And they also have their own implementation of an animal sound (the pig oinks, and the cat meows, etc.):

Example

```
// Base class
class Animal {
public:
    void animalSound() {
        cout << "The animal makes a sound \n";
    }
};

// Derived class
class Pig : public Animal {
public:
```

Fig 7. Polymorphism

To see the rest of the material the user should scroll down and to go to the test section, the user needs to press the "test" button. The user can press the "previous" button to go back to the previous screen.

Test

Please choose the correct answer

Classes and Objects

What keyword is used to create a class?

class Class cls

Classes and Objects

How do you create an object?

objectname classname; object name; classname objectname;

Class Methods

How many ways are there to define a method?

1 2 3

Constructors

A constructor is a special method that is automatically called when an _____ of a class is created.

method variable attribute

Fig 8. Test

The user needs to select one of the answers provided and then press the “check” button.

Test

Please choose the correct answer

Classes and Objects

What keyword is used to create a class?

class Class cls

check

Classes and Objects

How do you create an object?

objectname classname; object name; classname objectname;

check

Class Methods

How many ways are there to define a method?

1 2 3

check

Constructors

A constructor is a special method that is automatically called when an _____ of a class is created.

method variable attribute

check

correct answer

OK

The image shows a test interface with four questions. A message box with the text 'correct answer' and an 'OK' button is overlaid on the first question. The first question asks 'What keyword is used to create a class?' with options 'class', 'Class', and 'cls'. The 'class' option is selected. The second question asks 'How do you create an object?' with three options: 'objectname classname;', 'object name;', and 'classname objectname;'. The third question asks 'How many ways are there to define a method?' with options '1', '2', and '3'. The fourth question asks 'A constructor is a special method that is automatically called when an _____ of a class is created.' with options 'method', 'variable', and 'attribute'.

Fig 9. Correct answer

If the answer is correct a message box appears on the screen and the user should press “ok” in order to continue with the rest of the tasks.

Test

Please choose the correct answer

Classes and Objects

What keyword is used to create a class?

class Class cls

Classes and Objects

How do you create an object?

objectname classname; wrong answer classname objectname;

Class Methods

How many ways are there to define a

1 2 3

Constructors

A constructor is a special method that is automatically called when an _____ of a class is created.

method variable attribute

Fig 10. Wrong answer

If the answer is correct a message box appears on the screen and the user should press “ok” in order to continue with the rest of the tasks.

4.3. Testing

The program successfully runs and works as intended. The pictures in the “Run manual” show that the program was able to successfully start and present the study material on the screen. After pressing the start button the program was able to advance to the next screen and show the relevant material for the first topic. The “home” button works as intended and when pressed returned the screen to the start screen. The “next” button when pressed made the program present the next screen. The scroll feature that is present when there is a lot of material on the screen to fit on a single page works flawlessly. The subsequent pages that follow all successfully present the study material on the screen with no issues or problems. The “next” buttons on all the screens with the reading material all successfully work and always present the user with the relevant next screen in the cycle. Also, all the “home” buttons all work and manage to show the start screen when they are clicked. On the very last page that shows the reading material there is a button labelled “test”, and when clicked the button was successfully able to present the test section on screen with no issues. All the individual “check” buttons on the test section of the program successfully managed to check whether the answer that had been selected by the user was correct or not. The check feature works for all the questions on the test section of the program. The program successfully showed the appropriate dialog message on screen after clicking the “check” button. When the answer that was selected was wrong, a dialog message saying that the answer selected is wrong appeared on screen and when the answer selected was correct a prompt dialog message notifying the user that the answer is correct appeared on screen. The dialog feature worked for all instances that it is meant to. The “end” button to close the program and the “home” button to present the start screen on the testing section of the program both work as intended.

4.4 Program Description

When the program starts the user needs to press the “start” button on the centre of the screen to indicate that they are ready to begin.

The user is then presented with the reading material and examples for the first topic. When the user is finished with the material on the screen and intend to progress to the next stage in the program they need to press the “next” button. The “next” button is designed to promptly change the information shown on screen and show the material of the next topic. When the user intends to return to the the beginning of the program i.e the start screen, pressing the “home” button will allow for that to happen.

After going through all the pages with the reading material, the user is presented with the test section of the program. In order to access the test section of the program the user needs to click on the “test” button that is located on the very last page of the program that shows the reading material. This part of the program show multiple questions and answers for those questions. The user needs to select one answer for each question and if that answer is correct a prompt message appears on the screen notifying the user that the answer is correct and the user can press the close button in the corner or the ok button to close the message box. If the answer is wrong the same process takes place notifying the user that the answer that they selected is wrong.

When the user is finished they can close the program or return to the home screen and start the whole thing all over again.

CONCLUSION

Relatively extensive research on the topic, previous works and similar topics was carried out in order to find the necessity and relevance of the topic. Both positive and negative aspects of the reviewed works were taken into consideration and greatly contributed to the decisions made during the construction of the simulator. The algorithm of the problem for the topic of the work was developed. The simulator was successfully created as described and the works as intended. Several tests were done to make sure that the simulator works. Justification of the choice of software for the implementation of the task was highlighted as we as the necessity and relevance of the topic of work.

The main objective of the work was successfully completed and a simulator for the distance course on the topic “Classes in c++” in Programming ii has been created. The simulator will be able to help students improve in the topic and help them retain the information they would have learnt in class in a simple way.

The simulator works by introducing the relevant major concepts of the topic "Classes in C++" and offering a brief description and examples of how these concepts are meant to be implemented. When the user feels that they have read enough on a particular point and wish to progress to the next step they have to click the “next” button and after that they are introduced to a new concept on the topic. At the end of the reading material, the user is presented with a short multiple question test to test their knowledge on the material that they have just studied and when they feel that they are satisfied with the results they can then click the “end” button to close the application.

The simulator works perfectly as designed and described however there is always room for improvement. The simulator can be further improved by adding more topics related to “Classes in C++” and also by designing and connecting a database that will facilitate efficient updates of the study material, the test and the simulator itself.

REFERENCES

1. Stroustrup, Bjarne (1996). "A history of C++: 1979-1991". History of programming languages---II
2. Jasmin Blanchette, Mark Summerfield (2008). "C++ GUI Programming with Qt4". ISBN 0132703009, 9780132703000
3. Nicolai M. Josuttis: The C++ Standard Library, 2nd Edition. ISBN 978-0-321-62321-8
4. Julian Templeman. Microsoft® Visual C++® .NET Step by Step--Version 2003 (Step by Step (Microsoft)). EAN / ISBN : 9780735619074
5. Kaplan, Andreas M.; Haenlein, Michael (2016). "Higher education and the digital revolution: About MOOCs, SPOCs, social media, and the Cookie Monster". *Business Horizons*. 59 (4): 441–50. doi:10.1016/j.bushor.2016.03.008
6. Bennane, Abdellah (December 2012). "Adaptive Educational Software by Applying Reinforcement Learning" (PDF). *Informatics in Education*. 12 – via EBSCOhost.
7. Stroustrup, Bjarne (16 December 2021). "C++20: Reaching for the Aims of C++ - Bjarne Stroustrup - CppCon 2021". CppCon. Archived from the original on 30 December 2021. Retrieved 30 December 2021
8. Abrahams, David; Gurtovoy, Aleksey (2005). *C++ Template Metaprogramming: Concepts, Tools, and Techniques from Boost and Beyond*. Addison-Wesley. ISBN 0-321-22725-5
9. Lippman, Stanley B. (1996). *Inside the C++ Object Model*. Addison-Wesley. ISBN 0-201-83454-5
10. Information Technology Industry Council (15 October 2003). *Programming languages – C++ (Second ed.)*. Geneva: ISO/IEC. 14882:2003(E)
11. Stanton, S. (2001). Going the distance; Developing shared web-based learning programmes. *Occupational Therapy International*, 8(2), 96-106

12. Mukhtar, Khadijah; Javed, Kainat; Arooj, Mahwish; Sethi, Ahsan (May 2020). "Advantages, Limitations and Recommendations for online learning during COVID-19 pandemic era". *Pakistan Journal of Medical Sciences*. 36 (COVID19–S4): S27–S31. doi:10.12669/pjms.36.COVID19-S4.2785. ISSN 1682-024X. PMC 7306967. PMID 32582310

APPENDEX A

Algorithm Translation

Алгоритм побудови тренажера для дистанційного курсу на тему “ класів у C ++ ” <TAG1>

Крок 1. Користувачеві потрібно натиснути кнопку “ start ”, щоб запустити програму.

Перейдіть до кроку 2.

Крок 2: Класи та об’єкти. Користувачеві представлено визначення та приклад класу та об’єкта. Щоб перейти до наступного етапу, користувач повинен натиснути кнопку “ далі ”.

Крок 3: Методи класу. Користувачеві представлено визначення та приклад методів класу. Вони повинні натиснути “ наступний ”, щоб перейти до наступного екрана.

Крок 4: Конструктори. Користувачеві представлено коротке визначення конструкторів та приклад. Користувач повинен натиснути наступну кнопку, щоб перейти до наступного екрана.

Крок 5: Специфікатори доступу. На екрані з’являється опис специфікаторів доступу, і користувач повинен натиснути наступну кнопку, щоб перейти до наступного кроку.

Крок 6: Інкапсуляція. Користувачеві представлено визначення та опис інкапсуляції.

Крок 7: Успадкування. Користувачеві представлена детальна інформація про спадщину в класах C ++. Користувачеві потрібно прочитати та зрозуміти

інформацію про спадщину та натиснути наступну кнопку, щоб перейти до наступного екрана,

Крок 8: Багаторівнева спадковість. Користувачеві представлено визначення та опис багатокористувацької спадщини. Користувач повинен зрозуміти інформацію та наданий приклад. Прочитавши та зрозумівши, користувач повинен натиснути наступну кнопку, щоб перейти до наступного етапу.

Крок 9: Багатоспадкова спадковість. Користувачеві представлений короткий опис багатопопулярної спадщини та приклад. Щоб перейти до наступного етапу, користувач повинен натиснути наступну кнопку.

Крок 10: Доступ до спадкування. Користувачеві представлена інформація та приклад доступу до спадкування. Користувач повинен прочитати та зрозуміти матеріал, а потім натиснути наступну кнопку, щоб перейти до наступного етапу.

Крок 11: Поліморфізм. Користувачеві представлений короткий опис поліморфізму та короткий приклад синтаксису. Щоб перейти до наступного етапу, користувач повинен натиснути наступну кнопку.

Крок 12: Тест. Користувачеві пред'являються запитання з декількома варіантами, щоб перевірити свої знання про вивчений матеріал. Програма перевірятиме індивідуально, правильні відповіді чи ні. Якщо відповідь правильна, програма сповістить користувача про правильність відповіді, а якщо ні, то повідомить користувача, що відповідь неправильна. Зрештою, програма покаже користувачеві їх кінцевий результат.

Кінець.

APPENDEX B

Program Code

```
#pragma once

namespace Project1 {

    using namespace System;
    using namespace System::ComponentModel;
    using namespace System::Collections;
    using namespace System::Windows::Forms;
    using namespace System::Data;
    using namespace System::Drawing;

    /// <summary>
    /// Summary for Start
    /// </summary>
    public ref class Start : public System::Windows::Forms::Form
    {
    public:
        Start(void)
        {
            InitializeComponent();
            pnlClassesAndObjects->Hide();
            pnlMethods->Hide();
            pnlConstructors->Hide();
            pnlAccessSpecifiers->Hide();
            pnlEncapsulation->Hide();
            pnlInheritance->Hide();
            pnlTest->Hide();
            //
```

```

        //TODO: Add the constructor code here
        //
    }

```

protected:

```

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>

```

```

~Start()
{
    if (components)
    {
        delete components;
    }
}

```

```

private: System::Windows::Forms::Button^ startbutton;
private: System::Windows::Forms::Panel^ pnlClassesAndObjects;
private: System::Windows::Forms::Button^ home;
private: System::Windows::Forms::TextBox^ textBox1;
private: System::Windows::Forms::Label^ label1;
private: System::Windows::Forms::Button^ MethodsButton;
private: System::Windows::Forms::Panel^ pnlMethods;
private: System::Windows::Forms::Button^ Constructors;
private: System::Windows::Forms::Button^ BackToClassesAndObjects;

```

```

private: System::Windows::Forms::TextBox^ textBox2;
private: System::Windows::Forms::Label^ Methods;
private: System::Windows::Forms::Panel^ pnlConstructors;
private: System::Windows::Forms::TextBox^ textBox3;

```

```
private: System::Windows::Forms::Label^ label2;
private: System::Windows::Forms::Button^ BackToMethods;
private: System::Windows::Forms::Button^ ToAccessSpecifiers;
private: System::Windows::Forms::Panel^ pnlAccessSpecifiers;
private: System::Windows::Forms::Button^ BackToConstructors;
private: System::Windows::Forms::TextBox^ textBox4;
private: System::Windows::Forms::Label^ label3;
private: System::Windows::Forms::Button^ ToEncapsulation;
private: System::Windows::Forms::Panel^ pnlEncapsulation;
private: System::Windows::Forms::Button^ BackToAccessSpecifiers;

private: System::Windows::Forms::TextBox^ textBox5;
private: System::Windows::Forms::Label^ label4;
private: System::Windows::Forms::Button^ ToInheritance;
private: System::Windows::Forms::Panel^ pnlInheritance;
private: System::Windows::Forms::Button^ BackToEncapsulation;
private: System::Windows::Forms::TextBox^ textBox6;
private: System::Windows::Forms::Label^ label5;
private: System::Windows::Forms::Button^ ToPolymorphism;
private: System::Windows::Forms::Panel^ pnlPolymorphism;
private: System::Windows::Forms::Button^ BackToPolymorphism;
private: System::Windows::Forms::TextBox^ textBox7;
private: System::Windows::Forms::Label^ label6;
private: System::Windows::Forms::Button^ test;
private: System::Windows::Forms::Panel^ pnlTest;
private: System::Windows::Forms::Label^ label8;
private: System::Windows::Forms::Label^ label7;
private: System::Windows::Forms::GroupBox^ groupBox1;
private: System::Windows::Forms::Button^ Check1;
```

```
private: System::Windows::Forms::RadioButton^ radioButton3;  
private: System::Windows::Forms::RadioButton^ radioButton2;  
private: System::Windows::Forms::RadioButton^ radioButton1;  
private: System::Windows::Forms::Label^ label9;  
private: System::Windows::Forms::GroupBox^ groupBox2;  
private: System::Windows::Forms::Button^ Check2;  
private: System::Windows::Forms::RadioButton^ radioButton6;  
private: System::Windows::Forms::RadioButton^ radioButton5;  
private: System::Windows::Forms::RadioButton^ radioButton4;  
private: System::Windows::Forms::Label^ label10;  
private: System::Windows::Forms::VScrollBar^ vScrollBar1;  
private: System::Windows::Forms::GroupBox^ groupBox3;  
private: System::Windows::Forms::Button^ Check3;  
private: System::Windows::Forms::RadioButton^ radioButton7;  
private: System::Windows::Forms::RadioButton^ radioButton8;  
private: System::Windows::Forms::RadioButton^ radioButton9;  
private: System::Windows::Forms::Label^ label11;  
private: System::Windows::Forms::GroupBox^ groupBox4;  
private: System::Windows::Forms::Button^ Check4;  
private: System::Windows::Forms::RadioButton^ radioButton10;  
private: System::Windows::Forms::RadioButton^ radioButton11;  
private: System::Windows::Forms::RadioButton^ radioButton12;  
private: System::Windows::Forms::Label^ label12;
```

protected:

protected:

private:

```

    /// <summary>
    /// Required designer variable.
    /// </summary>
    System::ComponentModel::Container ^components;

#pragma region Windows Form Designer generated code
    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    void InitializeComponent(void)
    {
        System::ComponentModel::ComponentResourceManager^
resources = (gcnew
System::ComponentModel::ComponentResourceManager(Start::typeid));
        this->startbutton = (gcnew
System::Windows::Forms::Button());
        this->pnlClassesAndObjects = (gcnew
System::Windows::Forms::Panel());
        this->pnlMethods = (gcnew
System::Windows::Forms::Panel());
        this->pnlConstructors = (gcnew
System::Windows::Forms::Panel());
        this->pnlAccessSpecifiers = (gcnew
System::Windows::Forms::Panel());
        this->pnlEncapsulation = (gcnew
System::Windows::Forms::Panel());
        this->pnlInheritance = (gcnew
System::Windows::Forms::Panel());

```

```
        this->pnlPolymorphism = (gcnew
System::Windows::Forms::Panel());
        this->pnlTest = (gcnew System::Windows::Forms::Panel());
        this->groupBox4 = (gcnew
System::Windows::Forms::GroupBox());
        this->Check4 = (gcnew System::Windows::Forms::Button());
        this->radioButton10 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton11 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton12 = (gcnew
System::Windows::Forms::RadioButton());
        this->label12 = (gcnew System::Windows::Forms::Label());
        this->groupBox3 = (gcnew
System::Windows::Forms::GroupBox());
        this->Check3 = (gcnew System::Windows::Forms::Button());
        this->radioButton7 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton8 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton9 = (gcnew
System::Windows::Forms::RadioButton());
        this->label11 = (gcnew System::Windows::Forms::Label());
        this->groupBox2 = (gcnew
System::Windows::Forms::GroupBox());
        this->Check2 = (gcnew System::Windows::Forms::Button());
        this->radioButton6 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton5 = (gcnew
System::Windows::Forms::RadioButton());
```

```
        this->radioButton4 = (gcnew
System::Windows::Forms::RadioButton());
        this->label10 = (gcnew System::Windows::Forms::Label());
        this->vScrollBar1 = (gcnew
System::Windows::Forms::VScrollBar());
        this->groupBox1 = (gcnew
System::Windows::Forms::GroupBox());
        this->Check1 = (gcnew System::Windows::Forms::Button());
        this->radioButton3 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton2 = (gcnew
System::Windows::Forms::RadioButton());
        this->radioButton1 = (gcnew
System::Windows::Forms::RadioButton());
        this->label9 = (gcnew System::Windows::Forms::Label());
        this->label8 = (gcnew System::Windows::Forms::Label());
        this->label7 = (gcnew System::Windows::Forms::Label());
        this->test = (gcnew System::Windows::Forms::Button());
        this->BackToPolymorphism = (gcnew
System::Windows::Forms::Button());
        this->textBox7 = (gcnew
System::Windows::Forms::TextBox());
        this->label6 = (gcnew System::Windows::Forms::Label());
        this->ToPolymorphism = (gcnew
System::Windows::Forms::Button());
        this->BackToEncapsulation = (gcnew
System::Windows::Forms::Button());
        this->textBox6 = (gcnew
System::Windows::Forms::TextBox());
        this->label5 = (gcnew System::Windows::Forms::Label());
```



```
        this->ToInheritance = (gcnew
System::Windows::Forms::Button());
        this->BackToAccessSpecifiers = (gcnew
System::Windows::Forms::Button());
        this->textBox5 = (gcnew
System::Windows::Forms::TextBox());
        this->label4 = (gcnew System::Windows::Forms::Label());
        this->ToEncapsulation = (gcnew
System::Windows::Forms::Button());
        this->BackToConstructors = (gcnew
System::Windows::Forms::Button());
        this->textBox4 = (gcnew
System::Windows::Forms::TextBox());
        this->label3 = (gcnew System::Windows::Forms::Label());
        this->ToAccessSpecifiers = (gcnew
System::Windows::Forms::Button());
        this->BackToMethods = (gcnew
System::Windows::Forms::Button());
        this->textBox3 = (gcnew
System::Windows::Forms::TextBox());
        this->label2 = (gcnew System::Windows::Forms::Label());
        this->Constructors = (gcnew
System::Windows::Forms::Button());
        this->BackToClassesAndObjects = (gcnew
System::Windows::Forms::Button());
        this->textBox2 = (gcnew
System::Windows::Forms::TextBox());
        this->Methods = (gcnew System::Windows::Forms::Label());
        this->MethodsButton = (gcnew
System::Windows::Forms::Button());
```

```

this->home = (gcnew System::Windows::Forms::Button());
this->textBox1 = (gcnew
System::Windows::Forms::TextBox());
this->label1 = (gcnew System::Windows::Forms::Label());
this->pnlClassesAndObjects->SuspendLayout();
this->pnlMethods->SuspendLayout();
this->pnlConstructors->SuspendLayout();
this->pnlAccessSpecifiers->SuspendLayout();
this->pnlEncapsulation->SuspendLayout();
this->pnlInheritance->SuspendLayout();
this->pnlPolymorphism->SuspendLayout();
this->pnlTest->SuspendLayout();
this->groupBox4->SuspendLayout();
this->groupBox3->SuspendLayout();
this->groupBox2->SuspendLayout();
this->groupBox1->SuspendLayout();
this->SuspendLayout();
//
// startbutton
//
this->startbutton->Cursor =
System::Windows::Forms::Cursors::Hand;
this->startbutton->Location = System::Drawing::Point(298,
199);
this->startbutton->Name = L"startbutton";
this->startbutton->Size = System::Drawing::Size(136, 63);
this->startbutton->TabIndex = 0;
this->startbutton->Text = L"start";
this->startbutton->UseVisualStyleBackColor = true;

```

```

        this->startbutton->Click += gcnew System::EventHandler(this,
&Start::startbutton_Click);
        //
        // pnlClassesAndObjects
        //
        this->pnlClassesAndObjects->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->pnlClassesAndObjects->Controls->Add(this-
>pnlMethods);
        this->pnlClassesAndObjects->Controls->Add(this-
>MethodsButton);
        this->pnlClassesAndObjects->Controls->Add(this->home);
        this->pnlClassesAndObjects->Controls->Add(this->textBox1);
        this->pnlClassesAndObjects->Controls->Add(this->label1);
        this->pnlClassesAndObjects->Dock =
System::Windows::Forms::DockStyle::Fill;
        this->pnlClassesAndObjects->Location =
System::Drawing::Point(0, 0);
        this->pnlClassesAndObjects->Name =
L"pnlClassesAndObjects";
        this->pnlClassesAndObjects->Size =
System::Drawing::Size(770, 559);
        this->pnlClassesAndObjects->TabIndex = 1;
        //
        // pnlMethods
        //
        this->pnlMethods->Controls->Add(this->pnlConstructors);
        this->pnlMethods->Controls->Add(this->Constructors);
        this->pnlMethods->Controls->Add(this-
>BackToClassesAndObjects);

```

```

this->pnlMethods->Controls->Add(this->textBox2);
this->pnlMethods->Controls->Add(this->Methods);
this->pnlMethods->Dock =
System::Windows::Forms::DockStyle::Fill;
this->pnlMethods->Location = System::Drawing::Point(0, 0);
this->pnlMethods->Name = L"pnlMethods";
this->pnlMethods->Size = System::Drawing::Size(770, 559);
this->pnlMethods->TabIndex = 4;
//
// pnlConstructors
//
this->pnlConstructors->Controls->Add(this-
>pnlAccessSpecifiers);
this->pnlConstructors->Controls->Add(this-
>ToAccessSpecifiers);
this->pnlConstructors->Controls->Add(this->BackToMethods);
this->pnlConstructors->Controls->Add(this->textBox3);
this->pnlConstructors->Controls->Add(this->label2);
this->pnlConstructors->Dock =
System::Windows::Forms::DockStyle::Fill;
this->pnlConstructors->Location = System::Drawing::Point(0,
0);
this->pnlConstructors->Name = L"pnlConstructors";
this->pnlConstructors->Size = System::Drawing::Size(770,
559);
this->pnlConstructors->TabIndex = 4;
//
// pnlAccessSpecifiers
//

```

```

        this->pnlAccessSpecifiers->Controls->Add(this-
>pnlEncapsulation);
        this->pnlAccessSpecifiers->Controls->Add(this-
>ToEncapsulation);
        this->pnlAccessSpecifiers->Controls->Add(this-
>BackToConstructors);
        this->pnlAccessSpecifiers->Controls->Add(this->textBox4);
        this->pnlAccessSpecifiers->Controls->Add(this->label3);
        this->pnlAccessSpecifiers->Dock =
System::Windows::Forms::DockStyle::Fill;
        this->pnlAccessSpecifiers->Location =
System::Drawing::Point(0, 0);
        this->pnlAccessSpecifiers->Name = L"pnlAccessSpecifiers";
        this->pnlAccessSpecifiers->Size = System::Drawing::Size(770,
559);

        this->pnlAccessSpecifiers->TabIndex = 4;
        //
        // pnlEncapsulation
        //
        this->pnlEncapsulation->Controls->Add(this->pnlInheritance);
        this->pnlEncapsulation->Controls->Add(this->ToInheritance);
        this->pnlEncapsulation->Controls->Add(this-
>BackToAccessSpecifiers);
        this->pnlEncapsulation->Controls->Add(this->textBox5);
        this->pnlEncapsulation->Controls->Add(this->label4);
        this->pnlEncapsulation->Dock =
System::Windows::Forms::DockStyle::Fill;
        this->pnlEncapsulation->Location = System::Drawing::Point(0,
0);
        this->pnlEncapsulation->Name = L"pnlEncapsulation";

```

```

this->pnlEncapsulation->Size = System::Drawing::Size(770,
559);

this->pnlEncapsulation->TabIndex = 4;
//
// pnlInheritance
//
this->pnlInheritance->Controls->Add(this->pnlPolymorphism);
this->pnlInheritance->Controls->Add(this->ToPolymorphism);
this->pnlInheritance->Controls->Add(this-
>BackToEncapsulation);

this->pnlInheritance->Controls->Add(this->textBox6);
this->pnlInheritance->Controls->Add(this->label5);
this->pnlInheritance->Dock =
System::Windows::Forms::DockStyle::Fill;

this->pnlInheritance->Location = System::Drawing::Point(0, 0);
this->pnlInheritance->Name = L"pnlInheritance";
this->pnlInheritance->Size = System::Drawing::Size(770, 559);
this->pnlInheritance->TabIndex = 4;
//
// pnlPolymorphism
//
this->pnlPolymorphism->Controls->Add(this->pnlTest);
this->pnlPolymorphism->Controls->Add(this->test);
this->pnlPolymorphism->Controls->Add(this-
>BackToPolymorphism);

this->pnlPolymorphism->Controls->Add(this->textBox7);
this->pnlPolymorphism->Controls->Add(this->label6);
this->pnlPolymorphism->Dock =
System::Windows::Forms::DockStyle::Fill;

```

```

this->pnlPolymorphism->Location = System::Drawing::Point(0,
0);

this->pnlPolymorphism->Name = L"pnlPolymorphism";
this->pnlPolymorphism->Size = System::Drawing::Size(770,
559);

this->pnlPolymorphism->TabIndex = 4;
//
// pnlTest
//
this->pnlTest->Controls->Add(this->groupBox4);
this->pnlTest->Controls->Add(this->groupBox3);
this->pnlTest->Controls->Add(this->groupBox2);
this->pnlTest->Controls->Add(this->vScrollBar1);
this->pnlTest->Controls->Add(this->groupBox1);
this->pnlTest->Controls->Add(this->label8);
this->pnlTest->Controls->Add(this->label7);
this->pnlTest->Dock =
System::Windows::Forms::DockStyle::Fill;

this->pnlTest->Location = System::Drawing::Point(0, 0);
this->pnlTest->Name = L"pnlTest";
this->pnlTest->Size = System::Drawing::Size(770, 559);
this->pnlTest->TabIndex = 4;
//
// groupBox4
//
this->groupBox4->Controls->Add(this->Check4);
this->groupBox4->Controls->Add(this->radioButton10);
this->groupBox4->Controls->Add(this->radioButton11);
this->groupBox4->Controls->Add(this->radioButton12);
this->groupBox4->Controls->Add(this->label12);

```

```

this->groupBox4->Location = System::Drawing::Point(51,
425);

this->groupBox4->Name = L"groupBox4";
this->groupBox4->Size = System::Drawing::Size(673, 100);
this->groupBox4->TabIndex = 6;
this->groupBox4->TabStop = false;
this->groupBox4->Text = L"Constructors";
//
// Check4
//
this->Check4->Cursor =
System::Windows::Forms::Cursors::Hand;
this->Check4->Location = System::Drawing::Point(598, 61);
this->Check4->Name = L"Check4";
this->Check4->Size = System::Drawing::Size(75, 23);
this->Check4->TabIndex = 4;
this->Check4->Text = L"check";
this->Check4->UseVisualStyleBackColor = true;
this->Check4->Click += gcnew System::EventHandler(this,
&Start::Check4_Click);
//
// radioButton10
//
this->radioButton10->AutoSize = true;
this->radioButton10->Location = System::Drawing::Point(385,
61);

this->radioButton10->Name = L"radioButton10";
this->radioButton10->Size = System::Drawing::Size(75, 20);
this->radioButton10->TabIndex = 3;
this->radioButton10->TabStop = true;

```



```

this->radioButton10->Text = L"attribute";
this->radioButton10->UseVisualStyleBackColor = true;
//
// radioButton11
//
this->radioButton11->AutoSize = true;
this->radioButton11->Location = System::Drawing::Point(227,
61);

this->radioButton11->Name = L"radioButton11";
this->radioButton11->Size = System::Drawing::Size(77, 20);
this->radioButton11->TabIndex = 2;
this->radioButton11->TabStop = true;
this->radioButton11->Text = L"variable";
this->radioButton11->UseVisualStyleBackColor = true;
//
// radioButton12
//
this->radioButton12->AutoSize = true;
this->radioButton12->Location = System::Drawing::Point(14,
61);

this->radioButton12->Name = L"radioButton12";
this->radioButton12->Size = System::Drawing::Size(73, 20);
this->radioButton12->TabIndex = 1;
this->radioButton12->TabStop = true;
this->radioButton12->Text = L"method";
this->radioButton12->UseVisualStyleBackColor = true;
//
// label12
//
this->label12->AutoSize = true;

```

```

this->label12->Location = System::Drawing::Point(11, 30);
this->label12->Name = L"label12";
this->label12->Size = System::Drawing::Size(587, 16);
this->label12->TabIndex = 0;
this->label12->Text = L"A constructor is a special method that
is automatically called when an _____ of"
        L" a class is created.\r\n";
//
// groupBox3
//
this->groupBox3->Controls->Add(this->Check3);
this->groupBox3->Controls->Add(this->radioButton7);
this->groupBox3->Controls->Add(this->radioButton8);
this->groupBox3->Controls->Add(this->radioButton9);
this->groupBox3->Controls->Add(this->label11);
this->groupBox3->Location = System::Drawing::Point(48,
319);

this->groupBox3->Name = L"groupBox3";
this->groupBox3->Size = System::Drawing::Size(673, 100);
this->groupBox3->TabIndex = 5;
this->groupBox3->TabStop = false;
this->groupBox3->Text = L"Class Methods";
//
// Check3
//
this->Check3->Cursor =
System::Windows::Forms::Cursors::Hand;
this->Check3->Location = System::Drawing::Point(598, 61);
this->Check3->Name = L"Check3";
this->Check3->Size = System::Drawing::Size(75, 23);

```

```

this->Check3->TabIndex = 4;
this->Check3->Text = L"check";
this->Check3->UseVisualStyleBackColor = true;
this->Check3->Click += gcnew System::EventHandler(this,
&Start::Check3_Click);
//
// radioButton7
//
this->radioButton7->AutoSize = true;
this->radioButton7->Location = System::Drawing::Point(385,
61);

this->radioButton7->Name = L"radioButton7";
this->radioButton7->Size = System::Drawing::Size(35, 20);
this->radioButton7->TabIndex = 3;
this->radioButton7->TabStop = true;
this->radioButton7->Text = L"3";
this->radioButton7->UseVisualStyleBackColor = true;
//
// radioButton8
//
this->radioButton8->AutoSize = true;
this->radioButton8->Location = System::Drawing::Point(227,
61);

this->radioButton8->Name = L"radioButton8";
this->radioButton8->Size = System::Drawing::Size(35, 20);
this->radioButton8->TabIndex = 2;
this->radioButton8->TabStop = true;
this->radioButton8->Text = L"2";
this->radioButton8->UseVisualStyleBackColor = true;
//

```

```

// radioButton9
//
this->radioButton9->AutoSize = true;
this->radioButton9->Location = System::Drawing::Point(14,
61);

this->radioButton9->Name = L"radioButton9";
this->radioButton9->Size = System::Drawing::Size(35, 20);
this->radioButton9->TabIndex = 1;
this->radioButton9->TabStop = true;
this->radioButton9->Text = L"1";
this->radioButton9->UseVisualStyleBackColor = true;
//
// label11
//
this->label11->AutoSize = true;
this->label11->Location = System::Drawing::Point(11, 30);
this->label11->Name = L"label11";
this->label11->Size = System::Drawing::Size(280, 16);
this->label11->TabIndex = 0;
this->label11->Text = L"How many ways are there to define a
method\?";

//
// groupBox2
//
this->groupBox2->Controls->Add(this->Check2);
this->groupBox2->Controls->Add(this->radioButton6);
this->groupBox2->Controls->Add(this->radioButton5);
this->groupBox2->Controls->Add(this->radioButton4);
this->groupBox2->Controls->Add(this->label10);

```

```

this->groupBox2->Location = System::Drawing::Point(51,
213);

this->groupBox2->Name = L"groupBox2";
this->groupBox2->Size = System::Drawing::Size(673, 100);
this->groupBox2->TabIndex = 4;
this->groupBox2->TabStop = false;
this->groupBox2->Text = L"Classes and Objects";
//
// Check2
//
this->Check2->Cursor =
System::Windows::Forms::Cursors::Hand;
this->Check2->Location = System::Drawing::Point(598, 61);
this->Check2->Name = L"Check2";
this->Check2->Size = System::Drawing::Size(75, 23);
this->Check2->TabIndex = 4;
this->Check2->Text = L"check";
this->Check2->UseVisualStyleBackColor = true;
this->Check2->Click += gcnew System::EventHandler(this,
&Start::Check2_Click);
//
// radioButton6
//
this->radioButton6->AutoSize = true;
this->radioButton6->Location = System::Drawing::Point(385,
61);

this->radioButton6->Name = L"radioButton6";
this->radioButton6->Size = System::Drawing::Size(171, 20);
this->radioButton6->TabIndex = 3;
this->radioButton6->TabStop = true;

```

```

this->radioButton6->Text = L"classname objectname;";
this->radioButton6->UseVisualStyleBackColor = true;
//
// radioButton5
//
this->radioButton5->AutoSize = true;
this->radioButton5->Location = System::Drawing::Point(227,
61);

this->radioButton5->Name = L"radioButton5";
this->radioButton5->Size = System::Drawing::Size(105, 20);
this->radioButton5->TabIndex = 2;
this->radioButton5->TabStop = true;
this->radioButton5->Text = L"object name;";
this->radioButton5->UseVisualStyleBackColor = true;
//
// radioButton4
//
this->radioButton4->AutoSize = true;
this->radioButton4->Location = System::Drawing::Point(14,
61);

this->radioButton4->Name = L"radioButton4";
this->radioButton4->Size = System::Drawing::Size(171, 20);
this->radioButton4->TabIndex = 1;
this->radioButton4->TabStop = true;
this->radioButton4->Text = L"objectname classname;";
this->radioButton4->UseVisualStyleBackColor = true;
//
// label10
//
this->label10->AutoSize = true;

```

```

this->label10->Location = System::Drawing::Point(11, 30);
this->label10->Name = L"label10";
this->label10->Size = System::Drawing::Size(184, 16);
this->label10->TabIndex = 0;
this->label10->Text = L"How do you create an object\?";
//
// vScrollBar1
//
this->vScrollBar1->Location = System::Drawing::Point(740,
75);

this->vScrollBar1->Name = L"vScrollBar1";
this->vScrollBar1->Size = System::Drawing::Size(21, 413);
this->vScrollBar1->TabIndex = 3;
//
// groupBox1
//
this->groupBox1->Controls->Add(this->Check1);
this->groupBox1->Controls->Add(this->radioButton3);
this->groupBox1->Controls->Add(this->radioButton2);
this->groupBox1->Controls->Add(this->radioButton1);
this->groupBox1->Controls->Add(this->label9);
this->groupBox1->Location = System::Drawing::Point(48,
105);

this->groupBox1->Name = L"groupBox1";
this->groupBox1->Size = System::Drawing::Size(676, 102);
this->groupBox1->TabIndex = 2;
this->groupBox1->TabStop = false;
this->groupBox1->Text = L"Classes and Objects";
//
// Check1

```

```

//
this->Check1->Cursor =
System::Windows::Forms::Cursors::Hand;
this->Check1->Location = System::Drawing::Point(601, 60);
this->Check1->Name = L"Check1";
this->Check1->Size = System::Drawing::Size(75, 23);
this->Check1->TabIndex = 5;
this->Check1->Text = L"check";
this->Check1->UseVisualStyleBackColor = true;
this->Check1->Click += gnew System::EventHandler(this,
&Start::Check1_Click);
//
// radioButton3
//
this->radioButton3->AutoSize = true;
this->radioButton3->Location = System::Drawing::Point(340,
65);
this->radioButton3->Name = L"radioButton3";
this->radioButton3->Size = System::Drawing::Size(45, 20);
this->radioButton3->TabIndex = 3;
this->radioButton3->TabStop = true;
this->radioButton3->Text = L"cls";
this->radioButton3->UseVisualStyleBackColor = true;
//
// radioButton2
//
this->radioButton2->AutoSize = true;
this->radioButton2->Location = System::Drawing::Point(170,
65);
this->radioButton2->Name = L"radioButton2";

```



```

this->radioButton2->Size = System::Drawing::Size(62, 20);
this->radioButton2->TabIndex = 2;
this->radioButton2->TabStop = true;
this->radioButton2->Text = L"Class";
this->radioButton2->UseVisualStyleBackColor = true;
//
// radioButton1
//
this->radioButton1->AutoSize = true;
this->radioButton1->Location = System::Drawing::Point(17,
65);

this->radioButton1->Name = L"radioButton1";
this->radioButton1->Size = System::Drawing::Size(60, 20);
this->radioButton1->TabIndex = 1;
this->radioButton1->TabStop = true;
this->radioButton1->Text = L"class";
this->radioButton1->UseVisualStyleBackColor = true;
//
// label9
//
this->label9->AutoSize = true;
this->label9->Location = System::Drawing::Point(6, 28);
this->label9->Name = L"label9";
this->label9->Size = System::Drawing::Size(246, 16);
this->label9->TabIndex = 0;
this->label9->Text = L"What keyword is used to create a
class\?";

//
// label8
//

```

```

this->label8->AutoSize = true;
this->label8->Location = System::Drawing::Point(48, 64);
this->label8->Name = L"label8";
this->label8->Size = System::Drawing::Size(209, 16);
this->label8->TabIndex = 1;
this->label8->Text = L"Please choose the correct answer";
//
// label7
//
this->label7->AutoSize = true;
this->label7->Location = System::Drawing::Point(45, 38);
this->label7->Name = L"label7";
this->label7->Size = System::Drawing::Size(34, 16);
this->label7->TabIndex = 0;
this->label7->Text = L"Test";
//
// test
//
this->test->Cursor = System::Windows::Forms::Cursors::Hand;
this->test->Location = System::Drawing::Point(649, 495);
this->test->Name = L"test";
this->test->Size = System::Drawing::Size(75, 23);
this->test->TabIndex = 3;
this->test->Text = L"test";
this->test->UseVisualStyleBackColor = true;
this->test->Click += gcnew System::EventHandler(this,
&Start::test_Click);
//
// BackToPolymorphism
//

```

```

        this->BackToPolymorphism->Cursor =
System::Windows::Forms::Cursors::Hand;
        this->BackToPolymorphism->Location =
System::Drawing::Point(48, 495);
        this->BackToPolymorphism->Name =
L"BackToPolymorphism";
        this->BackToPolymorphism->Size =
System::Drawing::Size(75, 23);
        this->BackToPolymorphism->TabIndex = 2;
        this->BackToPolymorphism->Text = L"previous";
        this->BackToPolymorphism->UseVisualStyleBackColor = true;
        this->BackToPolymorphism->Click += gnew
System::EventHandler(this, &Start::BackToPolymorphism_Click);
        //
        // textBox7
        //
        this->textBox7->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox7->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox7->Location = System::Drawing::Point(51, 83);
        this->textBox7->Multiline = true;
        this->textBox7->Name = L"textBox7";
        this->textBox7->ReadOnly = true;
        this->textBox7->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
        this->textBox7->Size = System::Drawing::Size(673, 391);
        this->textBox7->TabIndex = 1;
        this->textBox7->Text = resources-
>GetString(L"textBox7.Text");

```

```

//
// label6
//
this->label6->AutoSize = true;
this->label6->Location = System::Drawing::Point(45, 38);
this->label6->Name = L"label6";
this->label6->Size = System::Drawing::Size(93, 16);
this->label6->TabIndex = 0;
this->label6->Text = L"Polymorphism";
//
// ToPolymorphism
//
this->ToPolymorphism->Cursor =
System::Windows::Forms::Cursors::Hand;
this->ToPolymorphism->Location =
System::Drawing::Point(625, 495);
this->ToPolymorphism->Name = L"ToPolymorphism";
this->ToPolymorphism->Size = System::Drawing::Size(75, 23);
this->ToPolymorphism->TabIndex = 3;
this->ToPolymorphism->Text = L"next";
this->ToPolymorphism->UseVisualStyleBackColor = true;
this->ToPolymorphism->Click += gcnew
System::EventHandler(this, &Start::ToPolymorphism_Click);
//
// BackToEncapsulation
//
this->BackToEncapsulation->Cursor =
System::Windows::Forms::Cursors::Hand;
this->BackToEncapsulation->Location =
System::Drawing::Point(51, 494);

```

```

        this->BackToEncapsulation->Name =
L"BackToEncapsulation";
        this->BackToEncapsulation->Size = System::Drawing::Size(75,
23);

        this->BackToEncapsulation->TabIndex = 2;
        this->BackToEncapsulation->Text = L"previous";
        this->BackToEncapsulation->UseVisualStyleBackColor = true;
        this->BackToEncapsulation->Click += gcnew
System::EventHandler(this, &Start::BackToEncapsulation_Click);
        //
        // textBox6
        //
        this->textBox6->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox6->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox6->CharacterCasing =
System::Windows::Forms::CharacterCasing::Upper;
        this->textBox6->Location = System::Drawing::Point(51, 83);
        this->textBox6->Multiline = true;
        this->textBox6->Name = L"textBox6";
        this->textBox6->ReadOnly = true;
        this->textBox6->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
        this->textBox6->Size = System::Drawing::Size(658, 391);
        this->textBox6->TabIndex = 1;
        this->textBox6->Text = resources-
>GetString(L"textBox6.Text");
        //
        // label5

```

```

//
this->label5->AutoSize = true;
this->label5->Location = System::Drawing::Point(45, 39);
this->label5->Name = L"label5";
this->label5->Size = System::Drawing::Size(72, 16);
this->label5->TabIndex = 0;
this->label5->Text = L"Inheritance";
//
// ToInheritance
//
this->ToInheritance->Cursor =
System::Windows::Forms::Cursors::Hand;
this->ToInheritance->Location = System::Drawing::Point(625,
494);
this->ToInheritance->Name = L"ToInheritance";
this->ToInheritance->Size = System::Drawing::Size(75, 23);
this->ToInheritance->TabIndex = 3;
this->ToInheritance->Text = L"next";
this->ToInheritance->UseVisualStyleBackColor = true;
this->ToInheritance->Click += gcnw
System::EventHandler(this, &Start::ToInheritance_Click);
//
// BackToAccessSpecifiers
//
this->BackToAccessSpecifiers->Cursor =
System::Windows::Forms::Cursors::Hand;
this->BackToAccessSpecifiers->Location =
System::Drawing::Point(51, 494);
this->BackToAccessSpecifiers->Name =
L"BackToAccessSpecifiers";

```

```

        this->BackToAccessSpecifiers->Size =
System::Drawing::Size(75, 23);
        this->BackToAccessSpecifiers->TabIndex = 2;
        this->BackToAccessSpecifiers->Text = L"previous";
        this->BackToAccessSpecifiers->UseVisualStyleBackColor =
true;
        this->BackToAccessSpecifiers->Click += gnew
System::EventHandler(this, &Start::BackToAccessSpecifiers_Click);
        //
        // textBox5
        //
        this->textBox5->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox5->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox5->Location = System::Drawing::Point(48, 74);
        this->textBox5->Multiline = true;
        this->textBox5->Name = L"textBox5";
        this->textBox5->ReadOnly = true;
        this->textBox5->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
        this->textBox5->Size = System::Drawing::Size(661, 400);
        this->textBox5->TabIndex = 1;
        this->textBox5->Text = resources-
>GetString(L"textBox5.Text");
        //
        // label4
        //
        this->label4->AutoSize = true;
        this->label4->Location = System::Drawing::Point(45, 39);

```

```

this->label4->Name = L"label4";
this->label4->Size = System::Drawing::Size(92, 16);
this->label4->TabIndex = 0;
this->label4->Text = L"Encapsulation";
//
// ToEncapsulation
//
this->ToEncapsulation->Cursor =
System::Windows::Forms::Cursors::Hand;
this->ToEncapsulation->Location =
System::Drawing::Point(625, 510);
this->ToEncapsulation->Name = L"ToEncapsulation";
this->ToEncapsulation->Size = System::Drawing::Size(75, 23);
this->ToEncapsulation->TabIndex = 3;
this->ToEncapsulation->Text = L"next";
this->ToEncapsulation->UseVisualStyleBackColor = true;
this->ToEncapsulation->Click += gcnew
System::EventHandler(this, &Start::ToEncapsulation_Click);
//
// BackToConstructors
//
this->BackToConstructors->Cursor =
System::Windows::Forms::Cursors::Hand;
this->BackToConstructors->Location =
System::Drawing::Point(48, 510);
this->BackToConstructors->Name = L"BackToConstructors";
this->BackToConstructors->Size = System::Drawing::Size(75,
23);
this->BackToConstructors->TabIndex = 2;
this->BackToConstructors->Text = L"previous";

```



```

        this->BackToConstructors->UseVisualStyleBackColor = true;
        this->BackToConstructors->Click += gnew
System::EventHandler(this, &Start::BackToConstructors_Click);
        //
        // textBox4
        //
        this->textBox4->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox4->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox4->Location = System::Drawing::Point(51, 74);
        this->textBox4->Multiline = true;
        this->textBox4->Name = L"textBox4";
        this->textBox4->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
        this->textBox4->Size = System::Drawing::Size(649, 414);
        this->textBox4->TabIndex = 1;
        this->textBox4->Text = resources-
>GetString(L"textBox4.Text");
        //
        // label3
        //
        this->label3->AutoSize = true;
        this->label3->Location = System::Drawing::Point(48, 38);
        this->label3->Name = L"label3";
        this->label3->Size = System::Drawing::Size(115, 16);
        this->label3->TabIndex = 0;
        this->label3->Text = L"Access Specifiers";
        //
        // ToAccessSpecifiers

```

```

//
this->ToAccessSpecifiers->Cursor =
System::Windows::Forms::Cursors::Hand;
this->ToAccessSpecifiers->Location =
System::Drawing::Point(634, 494);
this->ToAccessSpecifiers->Name = L"ToAccessSpecifiers";
this->ToAccessSpecifiers->Size = System::Drawing::Size(75,
23);

this->ToAccessSpecifiers->TabIndex = 3;
this->ToAccessSpecifiers->Text = L"next";
this->ToAccessSpecifiers->TextAlign =
System::Drawing::ContentAlignment::TopCenter;
this->ToAccessSpecifiers->UseVisualStyleBackColor = true;
this->ToAccessSpecifiers->Click += gnew
System::EventHandler(this, &Start::ToAccessSpecifiers_Click);
//
// BackToMethods
//
this->BackToMethods->Cursor =
System::Windows::Forms::Cursors::Hand;
this->BackToMethods->Location = System::Drawing::Point(48,
494);
this->BackToMethods->Name = L"BackToMethods";
this->BackToMethods->Size = System::Drawing::Size(75, 23);
this->BackToMethods->TabIndex = 2;
this->BackToMethods->Text = L"previous";
this->BackToMethods->UseVisualStyleBackColor = true;
this->BackToMethods->Click += gnew
System::EventHandler(this, &Start::BackToMethods_Click);
//

```

```

// textBox3
//
this->textBox3->BackColor =
System::Drawing::SystemColors::ControlLight;
this->textBox3->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
this->textBox3->Location = System::Drawing::Point(48, 74);
this->textBox3->Multiline = true;
this->textBox3->Name = L"textBox3";
this->textBox3->ReadOnly = true;
this->textBox3->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
this->textBox3->Size = System::Drawing::Size(661, 400);
this->textBox3->TabIndex = 1;
this->textBox3->Text = resources-
>GetString(L"textBox3.Text");
//
// label2
//
this->label2->AutoSize = true;
this->label2->Location = System::Drawing::Point(45, 38);
this->label2->Name = L"label2";
this->label2->Size = System::Drawing::Size(81, 16);
this->label2->TabIndex = 0;
this->label2->Text = L"Constructors";
//
// Constructors
//
this->Constructors->Cursor =
System::Windows::Forms::Cursors::Hand;

```

```

this->Constructors->Location = System::Drawing::Point(634,
480);

this->Constructors->Name = L"Constructors";
this->Constructors->Size = System::Drawing::Size(75, 23);
this->Constructors->TabIndex = 3;
this->Constructors->Text = L"next";
this->Constructors->UseVisualStyleBackColor = true;
this->Constructors->Click += gnew
System::EventHandler(this, &Start::Constructors_Click);
//
// BackToClassesAndObjects
//
this->BackToClassesAndObjects->Cursor =
System::Windows::Forms::Cursors::Hand;
this->BackToClassesAndObjects->Location =
System::Drawing::Point(48, 480);
this->BackToClassesAndObjects->Name =
L"BackToClassesAndObjects";
this->BackToClassesAndObjects->Size =
System::Drawing::Size(75, 23);
this->BackToClassesAndObjects->TabIndex = 2;
this->BackToClassesAndObjects->Text = L"previous";
this->BackToClassesAndObjects->UseVisualStyleBackColor =
true;
this->BackToClassesAndObjects->Click += gnew
System::EventHandler(this, &Start::BackToClassesAndObjects_Click);
//
// textBox2
//

```

```

        this->textBox2->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox2->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox2->Location = System::Drawing::Point(48, 74);
        this->textBox2->Multiline = true;
        this->textBox2->Name = L"textBox2";
        this->textBox2->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;
        this->textBox2->Size = System::Drawing::Size(661, 383);
        this->textBox2->TabIndex = 1;
        this->textBox2->Text = resources-
>GetString(L"textBox2.Text");
        //
        // Methods
        //
        this->Methods->AutoSize = true;
        this->Methods->Location = System::Drawing::Point(45, 38);
        this->Methods->Name = L"Methods";
        this->Methods->Size = System::Drawing::Size(59, 16);
        this->Methods->TabIndex = 0;
        this->Methods->Text = L"Methods";
        //
        // MethodsButton
        //
        this->MethodsButton->Location =
System::Drawing::Point(634, 480);
        this->MethodsButton->Name = L"MethodsButton";
        this->MethodsButton->Size = System::Drawing::Size(75, 23);
        this->MethodsButton->TabIndex = 3;

```

```

        this->MethodsButton->Text = L"next";
        this->MethodsButton->UseVisualStyleBackColor = true;
        this->MethodsButton->Click += gcnew
System::EventHandler(this, &Start::MethodsButton_Click);
        //
        // home
        //
        this->home->Cursor =
System::Windows::Forms::Cursors::Hand;
        this->home->Location = System::Drawing::Point(65, 480);
        this->home->Name = L"home";
        this->home->Size = System::Drawing::Size(75, 23);
        this->home->TabIndex = 2;
        this->home->Text = L"home";
        this->home->UseVisualStyleBackColor = true;
        this->home->Click += gcnew System::EventHandler(this,
&Start::home_Click);
        //
        // textBox1
        //
        this->textBox1->BackColor =
System::Drawing::SystemColors::ControlLight;
        this->textBox1->BorderStyle =
System::Windows::Forms::BorderStyle::FixedSingle;
        this->textBox1->Location = System::Drawing::Point(65, 74);
        this->textBox1->Multiline = true;
        this->textBox1->Name = L"textBox1";
        this->textBox1->ReadOnly = true;
        this->textBox1->ScrollBars =
System::Windows::Forms::ScrollBars::Vertical;

```

```

this->textBox1->Size = System::Drawing::Size(644, 383);
this->textBox1->TabIndex = 1;
this->textBox1->Text = resources-
>GetString(L"textBox1.Text");
//
// label1
//
this->label1->AutoSize = true;
this->label1->Location = System::Drawing::Point(62, 39);
this->label1->Name = L"label1";
this->label1->Size = System::Drawing::Size(131, 16);
this->label1->TabIndex = 0;
this->label1->Text = L"Classes and Objects";
//
// Start
//
this->AutoScaleDimensions = System::Drawing::SizeF(8, 16);
this->AutoScaleMode =
System::Windows::Forms::AutoScaleMode::Font;
this->ClientSize = System::Drawing::Size(770, 559);
this->Controls->Add(this->pnlClassesAndObjects);
this->Controls->Add(this->startbutton);
this->Name = L"Start";
this->Text = L"Start";
this->Load += gcnew System::EventHandler(this,
&Start::Start_Load);
this->pnlClassesAndObjects->ResumeLayout(false);
this->pnlClassesAndObjects->PerformLayout();
this->pnlMethods->ResumeLayout(false);
this->pnlMethods->PerformLayout();

```

```
this->pnlConstructors->ResumeLayout(false);
this->pnlConstructors->PerformLayout();
this->pnlAccessSpecifiers->ResumeLayout(false);
this->pnlAccessSpecifiers->PerformLayout();
this->pnlEncapsulation->ResumeLayout(false);
this->pnlEncapsulation->PerformLayout();
this->pnlInheritance->ResumeLayout(false);
this->pnlInheritance->PerformLayout();
this->pnlPolymorphism->ResumeLayout(false);
this->pnlPolymorphism->PerformLayout();
this->pnlTest->ResumeLayout(false);
this->pnlTest->PerformLayout();
this->groupBox4->ResumeLayout(false);
this->groupBox4->PerformLayout();
this->groupBox3->ResumeLayout(false);
this->groupBox3->PerformLayout();
this->groupBox2->ResumeLayout(false);
this->groupBox2->PerformLayout();
this->groupBox1->ResumeLayout(false);
this->groupBox1->PerformLayout();
this->ResumeLayout(false);

    }
#pragma endregion

    private: System::Void startbutton_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlClassesAndObjects->Show();
    }
```



```
private: System::Void Start_Load(System::Object^ sender,
System::EventArgs^ e) {

    }

private: System::Void home_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlClassesAndObjects->Hide();
}

private: System::Void MethodsButton_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlMethods->Show();
}

private: System::Void BackToClassesAndObjects_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlMethods->Hide();
}

private: System::Void Constructors_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlConstructors->Show();
}

private: System::Void BackToMethods_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlConstructors->Hide();
}

private: System::Void ToAccessSpecifiers_Click(System::Object^ sender,
System::EventArgs^ e) {
    pnlAccessSpecifiers->Show();
}

private: System::Void BackToConstructors_Click(System::Object^ sender,
System::EventArgs^ e) {
```

```
        pnlAccessSpecifiers->Hide();
    }
private: System::Void ToEncapsulation_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlEncapsulation->Show();
    }
private: System::Void BackToAccessSpecifiers_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlEncapsulation->Hide();
    }
private: System::Void ToInheritance_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlInheritance->Show();
    }
private: System::Void BackToEncapsulation_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlInheritance->Hide();
    }
private: System::Void ToPolymorphism_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlPolymorphism->Show();
    }
private: System::Void BackToPolymorphism_Click(System::Object^ sender,
System::EventArgs^ e) {
        pnlPolymorphism->Hide();
    }
private: System::Void test_Click(System::Object^ sender, System::EventArgs^ e)
{
        pnlTest->Show();
    }
}
```

```
private: System::Void Check1_Click(System::Object^ sender, System::EventArgs^
e) {
    if (radioButton1->Checked == true) {
        MessageBox::Show("correct answer");
    }
    else {
        MessageBox::Show("wrong answer");
    }
}

private: System::Void Check2_Click(System::Object^ sender,
System::EventArgs^ e) {
    if (radioButton6->Checked == true) {
        MessageBox::Show("correct answer");
    }
    else {
        MessageBox::Show("wrong answer");
    }
}

private: System::Void Check3_Click(System::Object^ sender, System::EventArgs^
e) {
    if (radioButton7->Checked == true) {
        MessageBox::Show("correct answer");
    }
    else {
        MessageBox::Show("wrong answer");
    }
}

private: System::Void Check4_Click(System::Object^ sender, System::EventArgs^
e) {
    if (radioButton12->Checked == true) {
```

```
        MessageBox::Show("correct answer");  
    }  
    else {  
        MessageBox::Show("wrong answer");  
    }  
}  
};  
}
```