

**HIMALAYAN INSTITUTE OF TECHNOLOGY
DEHRADUN
(AFFILIATED TO H.N.B. GARHWAL CENTRAL UNIVERSITY,
SRINAGAR)**



**PROJECT REPORT ON
“E-LEARNING PLATFORM FOR PROGRAMMERS”
SUBMITTED FOR PARTIAL FULFILLMENT OF
BACHELOR OF COMPUTER APPLICATION
H.N.B. GARHWAL CENTRAL UNIVERSITY, SRINAGAR
2018-2021**



Submitted By:

Geeta Negi

BCA (6thSem.)

Enrollment No.-:G182200002

Submitted To:

Mrs. Shalini kuchhal

HIMALAYAN INSTITUTE OF TECHNOLOGY DEHRADUN
(AFFILIATED TO H.N.B. GARHWAL CENTRAL UNIVERSITY, SRINAGAR)

CERTIFICATE

This is to certify that project report entitled “**E-LEARNING PLATFORM FOR PROGRAMMERS**” submitted by **Geeta Negi** in the partial fulfillment of the requirement for the award of the degree “**Bachelor of Computer Applications**” is a record of bonafide work carried out in this organization under my supervision has not been submitted anywhere else for any other purpose.

Mrs. Shalini Kuchhal
(Subject In charge)

Prof. Arrchana Jasola
(Director)

ACKNOWLEDGEMENT

I am grateful to my college **HIMALAYAN INSTITUTE OF TECHNOLOGY**, Dehradun and our director Prof. Arrchana Jasola for giving me an opportunity to undergo training program which is a part of course curriculum in BCA. I have prepared a project entitled “**E-LEARNING PLATFORM FOR PROGRAMMERS**”.

I am also grateful to my faculty **Mrs. Shalini Kuchhal** for guiding me and supervising to complete the project report.

Geeta Negi

BCA-6th Sem.

DECLARATION

I hereby declare that the project report entitled, “**E-LEARNING PLATFORM FOR PROGRAMMERS**” has been written and submitted by me, under guidance of computer science faculty in my original work.

The finding in the report is based on the task done by me in the Himalayan Institute of Technology, Dehradun while preparing the report, I have not copied anything from any source or other project submitted for similar purpose.

Place: Dehradun

Date: 09-08-2021

Geeta Negi

BCA-6thSem.

TABLE OF CONTENTS

1. Introduction
 - a) About Project
 - b) Aims, Objective & vision
2. System Study
 - a) System Objectives
 - b) Feasibility Study
3. System Analysis
 - a) Software/Hardware Requirements Specifications
4. System Design
 - a) E-R Diagram
 - b) Data Flow Diagram
 - c) Database Design
5. System Coding
6. Input/output form Screen Shots
7. Testing
8. Maintenance
9. Conclusion
- 10.Future Scope
- 11.References

INTRODUCTION

ABOUT PROJECT

“E-LEARNING PLATFORM FOR PROGRAMMERS”

A learning system based on formalized teaching but with the help of electronic resources is known as E-learning. While teaching can be based in or out of the classrooms, the use of computers and the Internet forms the major component of E-learning. E-learning can also be termed as a network-enabled transfer of skills and knowledge, and the delivery of education is made to a large number of recipients at the same or different times. Earlier, it was not accepted wholeheartedly as it was assumed that this system lacked the human element required in learning. However, with the rapid progress in technology and the advancement in learning systems, it is now embraced by the masses.

The website “E-LEARNING PLATFORM FOR PROGRAMMERS” provides innovative courses directly to people at free of cost. This website E-LEARNING PLATFORM FOR PROGRAMMERS can be helpful to beginner to advanced level programmers to improve their knowledge and keep in touch what they have learned so far. People can study at their pace and access the course when it is convenient for them. Even school going students can use this website to learn programming from beginning.

Following are the tasks of the project:

- 1) Home page
- 2) About
- 3) Services
 - a) Programming courses
 - b) Algorithms
 - c) Quizzes
- 4) Contact form

AIMS, OBJECTIVES & VISIONS

The main Objective of the project is to enhance the quality of learning. Meet the learning style or needs of users. Improve the efficiency and effectiveness. Improve user-accessibility and time flexibility to engage learners in the learning process.

The main Aim is to offer flexible learning possibilities to everyone across the globe who wants to learn programming and prepare them for job interviews. They will also learn algorithmic techniques for solving various problems in easy and convenient way with full flexibility of time.

In brief, the following points describe the main objectives to be achieved through this website:

- To fully computerize all the aspects of E-learning platform for Programmers.
- To design a system that gives the better learning and is more efficient.
- To design a system that is reliable.
- System could add new records maintain them in database.
- To develop a website that will surely satisfied the service.

SYSTEM STUDY

SYSTEM OBJECTIVE

“E-LEARNING PLATFORM FOR PROGRAMMERS” is a requirement you have probably read a thousand times when reading about how to design effective eLearning courses. However, “clear” may have different meanings for different people; what is absolutely clear to you may not be as clear as you think to your audience. In this article, I’ll show you specific ways to ensure that your learning objectives are crystal clear for everyone, and I’ll also explain why this is so important for the success of your eLearning deliverable. Furthermore, I’ll share all the critical information you need to know about developing high-quality learning objectives for eLearning.

There are certain goals when it comes to eLearning and some of these are to:

- Enhance the quality of learning and teaching
- Meet the learning style or needs of students
- Improve the efficiency and effectiveness
- Improve user-accessibility and time flexibility to engage learners in the learning process

eLearning is vast and an expanding platform with huge prospective in higher education. Since there are many challenges in making eLearning effective, it is important to know how to manage it and access to the resources. Take a minute and just imagine if one is not having the roadmap to guide from start to finish is actually like plunging into eLearning without an effective strategy because learners would be lost in the learning content.

FEASIBILITY STUDY

Feasibility is the determination of whether or not a project is worth doing. The process followed in making this determination is called a Feasibility Study. Generally, feasibility studies are undertaken within tight time constraints and normally culminate in a written and oral feasibility report.

DIFFERENT TYPES OF FEASIBILITY:

In the conduct of the feasibility study, the analyst will usually consider seven distinct, but inter-related types of feasibility. They are:

Technical Feasibility:

This is concerned with specifying equipment and software that will successfully satisfy the user requirements. The technical needs of the system may vary considerably, but might include:

- The facility to produce outputs in a given time.
- Response time under certain conditions.
- Ability to process a certain volume of transaction at a particular speed.
- Facility to communicate data to distant location.

At the feasibility stage, it is desirable that two or three different configurations will be pursued that satisfy the key technical requirements but which represent different levels of ambition and cost. Out of all types of feasibility, technical feasibility generally is the most difficult to determine.

Operational Feasibility:

It is mainly related to human organizational and political aspects. The points to be considered are:

- What changes will be brought with the system?
- What organizational structures are disturbed?
- What new skills will be required?

Do the existing staff members have these skills? If not, can they be trained in due course of time?

People are inherent to change. An estimate should be made of how strong a reaction the user staff is likely to have towards the development of a computerized system. Therefore it is understandable that the introduction of a candidate system requires a special effort to educate, sell, and train the staff on new ways of conducting business.

Economic Feasibility:

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposal system. More commonly known as Cost / Benefit analysis; the procedure is to determine the benefits and savings that are expected from a proposed system and compare them with costs.

Among the most important information contained in feasibility study is cost benefit analysis-an assessment of the economic justification for a computer based system project. Cost-benefit analysis delineates costs for

project development and weights them against tangible and intangible benefits of a system.

Social Feasibility:

Social feasibility is a determination of whether a proposed project will be acceptable to the people or not. This determination typically examines the probability of the project being accepted by the group directly affected by the proposed system change. This project is proved to be socially acceptable and thus social feasible.

Management Feasibility:

It is a determination of whether a proposed project will be acceptable to management. This is also accepted by the management therefore Management Feasibility is present here.

Legal Feasibility:

Legal feasibility is a determination of whether a proposed project infringes on known Acts, Statues, as well as any pending legislation.

SYSTEM ANALYSIS

SOFTWARE AND HARDWARE REQUIREMENTS SPECIFICATION

A software requirements specification (SRS) is a detailed description of a software system to be developed with its functional and non-functional requirements. The SRS is developed based the agreement between customer and contractors. It may include the use cases of how user is going to interact with software system. The software requirement specification document consistent of all necessary requirements required for project development. To develop the software system we should have clear understanding of Software system. To achieve this we need to continuous communication with customers to gather all requirements. A good SRS defines the how Software System will interact with all internal modules, hardware, communication with other programs and human user interactions with wide range of real life scenarios. Using the Software requirements specification (SRS) document on QA lead, managers creates test plan. It is very important that testers must be cleared with every detail specified in this document in order to avoid faults in test cases and its expected results.

It is highly recommended to review or test SRS documents before start writing test cases and making any plan for testing. Let's see how to test SRS and the important point to keep in mind while testing it.

Correctness of SRS should be checked. Since the whole testing phase is dependent on SRS, it is very important to check its correctness. There are some standards with which we can compare and verify.

Ambiguity should be avoided. Sometimes in SRS, some words have more than one meaning and this might confused testers making it difficult to get the exact reference. It is advisable to check for such ambiguous words and make the meaning clear for better understanding.

Requirements should be complete. When tester writes test cases, what exactly is required from the application, is the first thing which needs to be clear. For e.g. if application needs to send the specific data of some specific size then it should be clearly mentioned in SRS that how much data and what is the size limit to send. Consistent requirements.SRS should be consistent within itself and consistent to its reference documents. If you call an input "Start and Stop" in one place, don't call it "Start/Stop" in another. This sets the standard and should be followed throughout the testing phase.

Verification of expected result: SRS should not have statements like "Work as expected", it should be clearly stated that what is expected since different testers would have different thinking aspects and may draw different results from this statement.

Testing environment: some applications need specific conditions to test and also a particular environment for accurate result. SRS should have clear documentation on what type of environment is needed to set up.

Pre-conditions defined clearly: One of the most important part of test cases is pre-conditions. If they are not met properly then actual result will always be different expected result. Verify that in SRS, all the pre-conditions are mentioned clearly. **Requirements ID:** these are the base of test case template. Based on requirement Ids, test case ids are written. Also, requirements ids make it easy to categorize modules so just

by looking at them, tester will know which module to refer. SRS must have them such as id defines a particular module.

Security and Performance criteria: Security is priority when a software is tested especially when it is built in such a way that it contains some crucial information when leaked can cause harm to business. Tester should check that all the security related requirements are properly defined and are clear to him. Also, when we talk about performance of a software, it plays a very important role in business so all the requirements related to performance must be clear to the tester and he must also know when and how much stress or load testing should be done to test the performance.

Assumption should be avoided: Sometimes when requirement is not cleared to tester, he tends to make some assumptions related to it, which is not a right way to do testing as assumptions could go wrong and hence, test results may vary. It is better to avoid assumptions and ask clients about all the “missing requirements” to have a better understanding of expected results.

Deletion of irrelevant requirements: There are more than one team who work on SRS so it might be possible that some irrelevant requirements are included in SRS. Based on the understanding of the software, tester can find out which are these requirements and remove them to avoid confusions and reduce work load.

Freeze requirements: When an ambiguous or incomplete requirement is sent to client to analyze and tester gets a reply, that requirement result will be updated in the next SRS version and client will freeze that requirement. Freezing here means that result will not change again until and unless some major addition or modification is introduced in the software.

SYSTEM DESIGN

SYSTEM DESIGN

The design of a system produces details that state how a system will meet the requirements identified during system analysis. In this phase, the detailed specifications for the new system will be formulated. They will describe its features: the outputs, inputs, files and databases, procedures—all in manner that meets project requirements.

In analysis of the system, we have seen what a system should do. In system design phase the emphasis will be on how to do what a system should do.

There are two main approaches to design:

- Data Centered Approach
- Process Centered Approach.

The present project is designed based on Data Centered Approach as the modern school of thinking on this subject is that if data is organized effectively the processes can always be designed in such a way that the data is made available to them.

The principle of Object Oriented Design (OOD) is adapted where designing is defined as a *collection of data and its associated characteristics (processes) as object. These objects are in line with real life objects.

Logical design:

Data Structure approach is being adapted since data can be associated with physical structure which can be seen and felt it is therefore logical to start with data rather than processes which are invisible. They are there, but processes cannot be implemented on any operating environment i.e. one need not know on which machine or operating system or database the system is going to be working.

Physical design:

In physical design, the outside of logical design is implemented using the features of a particular Environment.

The following are the contents of the data stores:

- Form for Entering new courses in the course calendar
- Form for entering timetable.
- Form for Assignment of course coordinator.

Design of procedure

Software design is both a process and a model. The design process is a sequence of steps

that's enable the designers to all aspects of the software to be built. Basic design principles enable the software engineer to navigate the design process. There are some principles of software design,

- The design process should not suffer from “tunnel vision”.
- The design should be traceable to the analysis.
- The design should not reinvent the wheel.
- The design should “minimize the intellectual distance” between the software and the problem as it exist in the world.
- The design should exhibit uniformity and integration.
- The design should be structured to accommodate change.
- The design should be structured to degrade gently, even when aberrant data, events, or operating conditions are encountered.
- Design is not coding, is not design.
- The design should be assessed for quality as it is being created, not after the fact.
- The design should reviewed to minimize conceptual (semantic).

Independence is measured using two qualitative criteria: Cohesion and Coupling. Cohesion is a measure of the relative functional strength of a module. Coupling is a measure of the relative interdependence among modules.

ARCHITECTURAL DESIGN

Architectural Design represents the structure of data and program components that are required to build this computer based system.

The software requirement can be mapped into various representation of the design model.

Structure design is often characterized often as a dataflow oriented design method because it provides a convenient transition from a DFD to a software.

Data Design of Accounts Managing Package creates a model of data and information that is represented at a high level of abstraction.

Architectural design is represented through ERDs, DFDs, Data Structure used.

In this section discussed about data flow diagram, Entity relationship diagram. these things are represented as diagrams with proper notation.

E-R DIAGRAM

An entity relationship diagram (ERD) shows the relationships of entity sets stored in a database.

An entity in this context is a component of data. In other words, ER diagrams illustrate the logical structure of databases.

At first glance an entity relationship diagram looks very much like a flowchart. It is the specialized symbols, and the meanings of those symbols, that make it unique.

Peter Chen developed ERDs in 1976. Since then Charles Bachman and James Martin have added some slight refinements to the basic ERD principles.

Common Entity Relationship Diagram Symbols

An ER diagram is a means of visualizing how the information a system produces is related.

There are five main components of an ERD:

- **Entities**, which are represented by rectangles. An entity is an object or concept about which you want to store information.
- A weak entity is an entity that must be defined by a foreign key relationship with another entity as it cannot be uniquely identified by its own attributes alone
- **Actions**, which are represented by diamond shapes, show how two entities share information in the database.
- In some cases, entities can be self-linked. For example, employees can supervise other employees.
- **Attributes**, which are represented by ovals. A key attribute is the unique, distinguishing characteristic of the entity. For example, an employee's social security number might be the employee's key attribute.
- A multivalued attribute can have more than one value. For example, an employee entity can have multiple skill values.
- **Connecting lines**, solid lines that connect attributes to show the relationships of entities in the diagram.

DATA FLOW DIAGRAM

The data flow diagram is one of the most improvement tools used by the system analyst DE Macro(1978) NadGandSarson (1979) popularized the use of the data flow diagram as modeling tools through their structured system analysis methodologies.

A data flow diagram should be the first tool used by system analyst to model system

components.

These components are the system processes; the data used by this processes and external entities that interact with the system and the information flows in the system.

There are four kinds of system components:

Process

Process show what system does. Each process has one or more data inputs and produce one or more data output, Circles in a data flow diagram represent process. Each process has unique name and number. This name and number appear inside the circle that represents the processes in a data flow diagram. The process is represented as circle

Data Stores

File or data store is depositary of data. They contain data that is retained in the system. Processes can enter the data into a data store or retrieve data from the data store. Each data store is represented by thin line in the data flow diagram and each data store has a unique name. The data store is represented in form of a line

External Entities

External entities are outside the system but they either supply input data into the system or use the system output, they are entities which the designer has no control. Square or rectangle may represent external entities that supply data into a system or sometimes called sources.

Data Flows

Dataflow model the passage of data in the system and are represented lines joining system components. An arrow indicates the direction of the flow and the line labeled by the name of the data flow.

Storage

It is represented by an open ended narrow rectangle. Data stores may be long-term files such as sales ledgers, or may be short-term accumulations: for example batches of documents that are waiting to be processed. Each data store should be given a reference followed by an arbitrary number.

DATABASE DESIGN

TABLES USED IN PROJECT (as Back End):

The tables used in this project are designed in Mysql. Mysql is the basic tool used in this project to store the data. There are one tables used named details having values “fname”, ”phone”, ”email”, ”umessage”.

DATABASE MANAGEMENT SYSTEM:

A database management system is probably best defined as a software system that allows access to data contained in a database. A database management system consists of a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred as the database, contains information about one particular enterprise. The primary goal of DBMS is to provide an environment that is both convenient and efficient to use in retrieving and storing database information.

COMPONENTS OF DATABASE:

1. Records :- A record is a collection of fields containing alphanumeric characters.
- 2.Fields :- A field is an item, which distinguishes one record, from another.
3. Key Fields:- A key field is a field that is unique to a particular record and is used distinguish a file from another in one or another respect.
4. Entity :- A person ,place ,thing or event about which information must be kept.
5. Attribute :-Piece of information describing a particular entity.

SYSTEM CODING

```

<!DOCTYPE html>
<html>
<head>
    <meta charset="utf-8">
    <meta http-equiv="X-UA-Comaptible" content="IE=edge">
    <title>iEdlearn</title>
    <meta name="description" content="">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link rel="stylesheet" type="text/css" href="css/style.css">
    <link href="form.php">
</head>
<body>
    <nav class="navbar background h-nav-resp">
        <ul class="nav-list v-class-resp">
            <div class="logo">iEdlearn</div>
            <li><a href="#home">Home</a> </li>
            <li><a href="#about">About</a></li>
            <li><a href="#services">Services</a></li>
            <li><a href="#team">Team</a></li>
            <li><a href="#contact">Contact Us</a></li>
        </ul>
        <div class="rightnav v-class-resp">
            <input type="text" name="search" id="search">
            <button class="btn btn-sm">Search</button>
        </div>
        <div class="burger">
            <div class="line"></div>
            <div class="line"></div>
            <div class="line"></div>
        </div>
    </nav>
    <section class="background firstsection" id="home">
        <div class="box-main">

```

```
<div class="firsthalf">
  <p class="text-big">
    Being the pro in Programming can be easy,
    Especially if you start with us!
  </p>
  <p class="text-small">Programming is fun! A gratifying experience that everyone
must try to engage in and learn.
```

Programming opens many doors that allow new developers to take advantage of and implement new projects.

If you are a beginner enthusiast or just looking to learn more programming languages, then you are on the right place...</p>

```
</p>
<div class="button">
  <button
                                                                    class="btn"
onclick="document.getElementById('about').scrollIntoView({scroll:'smooth'})">Know
More</button>
```

```
</div>
</div>
<div class="secondhalf">
  
</div>
</div>
</section>
```

```
<!-- about section starts -->
<section class="about" id="about">
  <div class="image">
    
  </div>
```

```
<div class="content">
  <h3>Our Mission</h3>
```

<p>We want to create a world where anyone can build something meaningful with technology, and everyone has the learning tools, resources, and opportunities to do so.

Code contains a world of possibilities — all that's required is the curiosity and drive to learn.

At iEdlearn, we are committed to empowering all people, regardless of where they are in their coding journeys, to continue to learn, grow, and make an impact on the world around them.

Programming has become much more significantly important than ever before.

Most of the programming aspirants want to develop their coding skills and effectively improve on their basics of programming.

We create simplified and interactive learning experiences.

<p>⚡ Algorithm is a procedure or formula used for solving a problem

⚡ Algorithms are essentially problem solvers -

their purpose is to solve and often automate a solution to a particular problem.

⚡ Learn algorithms for the competitive programming</p>

</div>

</div>

</div>

<div class="service-colum">

<div class="main">

<div class="service">

<div class="service-logo"> <i>

</i></div>

<h4>Quiz</h4>

<p>⚡ Quiz for practising and enjoymentment.

⚡ Check your process by attending daily quiz.

</p>

</div>

</div>

</div>

</div>

</div>

</section>

<!--services section end-->

<!--Team section start-->

<section class="team" id="team">

<center><div><h2 class="text-center">Team</h2></div></center>

<div class="container-card">

<div class="card">

<div class="picture">

</div>

```

<div class="main-content">
  <h3 class="name">GEETA NEGI</h3>
  <span> <a href="http://www.instagram.com/geeta.negi_"
    target="_blank"><i class="fab fa-instagram"></i> Geeta Negi</a></span>
  <h4 class="title">Programmer 🤖💻</h4>
</div>
<ul class="social">
  <li><a href="#" class="fa fa-github"></a></li>
  <li><a href="http://www.instagram.com/geeta.negi_"
    target="_blank" class="fa fa-instagram"></a></li>
  <li><a href="https://t.me/Geeta_Negi" target="_blank"
    class="fa fa-telegram"></a></li>
  <li><a href="https://www.facebook.com/geeta.negi.58118774" target="_blank"
    class="fa fa-facebook"></a></li>
</ul>
<div class="skills">
  <h6>Skills</h6>
  <ul>
    <li>C++</li>
    <li>C#</li>
    <li>Github</li>
    <li>Git</li>
    <li>HTML</li>
    <li>CSS</li>
    <li>PHP</li>
    <li>JavaScript</li>
  </ul>
</div>
</div>
<div class="card">
  <div class="picture">
    
  </div>

```

```

<div class="main-content">
  <h3 class="name">Ankit</h3>
  <span><i class="fab fa-instagram"></i> Ankit</span>
  <h4 class="title">Frontend developer 🧑💻</h4>
</div>
<ul class="social">
  <li><a href="#" class="fa fa-github"></a></li>
  <li><a href="#" class="fa fa-twitter"></a></li>
  <li><a href="#" class="fa fa-instagram"></a></li>
  <li><a href="#" class="fa fa-telegram"></a></li>
</ul>
<div class="skills">
  <h6>Skills</h6>
  <ul>
    <li>HTML</li>
    <li>CSS</li>
    <li>SASS</li>
    <li>JavaScript</li>
    <li>React</li>
    <li>Github</li>
    <li>Git</li>
    <li>VS code</li>
  </ul>
</div>
</div>
</div>
</section>
<script src="https://kit.fontawesome.com/dd8c49730d.js"
crossorigin="anonymous"></script>
<!--team section ends-->
<!-- contact section starts -->
<section class="contact" id="contact">
  <h1 class="text-center">Contact Us</h1>

```

```

<form action="contact.php" method="post">
<div class="form">
    <input class="form-input" type="text" placeholder="Enter your name" name="fname"
required>
    <input class="form-input" type="phone" placeholder="Enter your phone number"
name="phone">
    <input class="form-input" type="email" placeholder="Enter your email" name="email"
required>
    <textarea class="form-input" name="umessage" id="textarea" cols="30" rows="10"
placeholder="Elaborate your concern"></textarea>
    <br>
    <button class="btn btn-dark" type="submit"
name="submit">Submit</button>
</div>
</form>
</section>
<!-- contact section ends -->
<!-- footer section starts-->
<footer class="background">
    <p class="text-footer">Copyright &copy; 2021 www.iEdlearn.com -All rights
reserved</p>
</footer>
<!-- footer section ends-->
<script src="js/resp.js"></script>
</body>
</html>

```

```

<?php
$server_name="localhost";
$username="root";
$password="";
$database_name="db_contact";
$conn=mysqli_connect($server_name,$username,$password,$database_name);
//check the connection

```

```

if(!$conn)
{
    die("connection failed:".mysqli_connect_error());
}
if(isset($_POST['submit']))
{
    $fname=$_POST['fname'];
    $phone=$_POST['phone'];
    $email=$_POST['email'];
    $umessage=$_POST['umessage'];
    $sql_query = "INSERT INTO details(fname,phone,email,umessage)
    VALUES('$fname','$phone','$email','$umessage')";
    if(mysqli_query($conn,$sql_query))
    {
        echo"New details entry inserted successfully!";
    }
    else{
        echo"Error: ".$sql."".mysqli_error($conn);
    }
    mysqli_close($conn);
}
?>

```

```

/*stylesheet for homepage*/
*{
    margin: 0;
    padding: 0;
}
html{
    scroll-behavior: smooth;
}
.logo{
    width: 20%;
    display: flex;

```

```
    justify-content: center;
    align-items: center ;
    color: springgreen;
    font-size: 15pt;
    font-family: Georgia, 'Times New Roman', Times, serif;
}

.logo img{
width: 33%;
border: 3px solid white;
border-radius: 50px ;
margin: 6px;
}

.navbar{
display: flex ;
align-items: center;
justify-content: center;
position: sticky;
top: 0;
z-index: 1;
cursor: pointer ;
}

.nav-list{
    width: 70%;
    display: flex;
    align-items: center;
}

.nav-list li{
list-style: none;
padding: 20px 25px;
}

.nav-list li a{
    text-decoration: none;
    color: white;
    font-size: 15px;
```

```
font-family: 'ubuntu',sans-serif;
}
.nav-list li a:hover{
    color: grey;
}
.rightnav{
    width: 30%;
    text-align: right;
    padding: 0 23px;
}
#search{
    padding: 5px;
    font-size: 17px;
    border: 2px solid grey ;
    border-radius: 9px;
}
.background{
    background: rgba(0,0,0,0.7) url('../img/background.jpg');
    background-size: cover;
    background-blend-mode: darken;
}
.firstsection{
    height: 95vh;
}
.box-main{
    display: flex;
    justify-content: center;
    align-items: center;
    color: white;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    max-width: 53%;
margin: auto;
height: 73%;
}
```



```
.firsthalf{
  width: 80%;
  display: flex;
  flex-direction: column;
  justify-content: center;
}
.secondhalf{
  width: 30%;
}
.secondhalf img{
  width: 70%;
  border: 4px solid white;
  border-radius: 150px;
  display: block;
  margin: auto;
}
.text-big{
  font-size: 40px;
  margin-top: 85px;
}
.text-small{
  font-size: 18px;
}
.btn{
  padding: 8px 20px;
  margin: 7px 3px;
  border: 2px solid white;
  border-radius: 8px;
  background: none;
  color: white;
  font-family: 'ubuntu', sans-serif;
  font-size: 16px;
}
.btn-sm{
```

```
padding: 6px 10px;
vertical-align: middle;
font-size: 16px;
}
.btn-dark{
color: black;
border: 2px solid grey;
}
.btn-about{
color: black;
border: 2px solid grey;
}
.about{
background-color: lavender;
display: flex;
align-items: center;
justify-content: center;
flex-wrap: wrap;
}
.about .image{
flex:1 1 40rem;
}
.about .image img{
width:100%;
}
.about .content{
flex:1 1 40rem;
}
.about .content h3{
font-size: 3.5rem;
color: black;
}
.about .content p{
font-size: 1.5rem;
```

```
    color: darkblue;
    padding: 1rem 0;
}
body {
    background-color: rgba(0,0,0,0.5);
    position: relative;
}
.services{
    background-color: grey;
}
h1 {
    text-align: center;
    margin: 50px;
    color: #fff;
    text-transform: uppercase;
}
.main {
    width: 100%;
    min-height: 380px;
    text-align: center;
    position: relative;
    cursor: pointer;
}
.service {
    background: #111;
    padding: 30px;
    border-radius: 10px;
    position: absolute;
    bottom: 0;
    z-index: 0;
    box-shadow: 0 0 25px -5px rgb(253, 253, 253);
    transition: transform .8s;
    height: 372px;
    margin: 50px;
```

```
margin-bottom: 20px;
color: #fff;
}
.service-logo {
width: 130px;
height: 130px;
border-radius: 50%;
margin: -90px auto 0;
background: lightseagreen;
border: 18px solid rgb(240, 240, 20);
display: flex;
justify-content: center;
align-items: center;
}
.service h4 {
height: 35px;
width: 80%;
margin: 50px auto;
background: #111;
position: relative;
}
.service h4::after,
.service h4::before {
content: "";
width: 40px;
height: 30px;
position: absolute;
z-index: -1;
}
.service h4::after {
background: linear-gradient(to right, #fff, rgb(240, 240, 20));
right: -5px;
top: -5px;
}
```

```
.service h4::before {
  background: linear-gradient(to right, rgb(240, 240, 20), #fff);
  left: -5px;
  bottom: -5px;
}
.main:hover .service {
  transform: translateY(-40px);
}
.row {
  display: flex;
  justify-content: space-around;
  align-items: center;
  margin-top: 180px;
}
.service-colum {
  flex: 3;
  background: #111;
}
.team{
  margin: 0;
  font-family: tahoma;
  height: 95vh;
  background :lavender;
  display: flex;
  justify-content: center;
  align-items: center;
}
.team h2{
  margin: -275px -275px 70px 275px;
  color:black;
}
.container-card{
  position: relative;
  width: 700px;
```

```
display: flex;
justify-content: space-between;
flex-wrap: wrap;
}

.card {
padding: 23px 20px 20px;
margin: 63px -23px auto -210px;
margin-bottom: 30px;
background-color: lightseagreen;
text-align: center;
overflow: hidden;
position: relative;
box-shadow: 15px 13px 22px grey, -23px -13px 17px lightcyan;
width: 35%;
}

.card .picture {
display: inline-block;
height: 130px;
width: 130px;
z-index: 0;
position: relative;
border-radius: 20%;
}

.card .picture::before {
content: "";
width: 100%;
height: 0;
border-radius: 50%;
background-color: #1f1b1b;
position: absolute;
bottom: 135%;
right: 0;
left: 0;
opacity: 0.9;
```

```
    transform: scale(3);
    transition: all 0.3s linear 0s;
}
.card:hover .picture::before {
    height: 100%;
}
.card .picture::after {
    content: "";
    width: 100%;
    height: 100%;
    border-radius: 50%;
    background-color: #1f1b1b;
    position: absolute;
    top: 0;
    left: 0;
    z-index: -1;
}
.card .picture img {
    width: 100%;
    height: 100%;
    border-radius: 50%;
    transform: scale(1);
    transition: all 0.9s ease 0s;
}
.card:hover .picture img {
    box-shadow: 0 0 0 14px #f7f5ec;
    transform: scale(0.7);
}
.card .title {
    display: block;
    font-size: 15px;
    color: #1f1b1b;
    text-transform: capitalize;
    margin: 5px 37px 0;
```

```
}  
.card .main-content {  
  margin-bottom: -19px;  
}  
.card .main-content .name {  
  padding: 0;  
  margin: 8px;  
  text-transform: uppercase;  
}  
.card .social {  
  width: 100%;  
  padding: 0;  
  margin: 0;  
  background-color: #1f1b1b;  
  position: absolute;  
  bottom: -100px;  
  left: 0;  
  transition: all 0.5s ease 0s;  
}  
.card:hover .social {  
  bottom: 0;  
}  
.card .social li {  
  display: inline-block;  
}  
.card .social li a {  
  display: block;  
  padding: 10px;  
  font-size: 17px;  
  color: white;  
  transition: all 0.3s ease 0s;  
  text-decoration: none;  
}  
.card .social li a:hover {
```



```
    color: #3e3b44;
    background-color: #f7f5ec;
    transition: 0.5s;
}

.skills {
    text-align: left;
    padding: 15px;
}

.skills ul {
    list-style-type: none;
    margin: 0;
    padding: 0;
}

.skills ul li {
    border: 1px solid #1f1b1b;
    border-radius: 2px;
    display: inline-block;
    font-size: 12px;
    margin: 0 7px 7px 0;
    padding: 7px;
}

.skills ul li:hover {
    background: #1f1b1b;
    color: #fff;
    cursor: pointer;
    transition: 0.5s;
}

.contact{
    background-color:cornsilk;
    height: fit-content;
}

.text-center{
    text-align: center;
    padding-top: 5px;
```

```
    font-family:'ubuntu',sans-serif;
    font-size: 35px;
}
.contact h1{
    text-align: center;
    margin:1px;
    color: black;
    padding: 30px;
    text-transform: uppercase;
}
.form{
    max-width: 62%;
    margin:20px 0px 0px 100px;
}
.form-input{
    margin: 14px 0;
    padding: 5px 3px;
    width: 100%;
    font-size: 19px;
    border: 2px solid grey;
    border-radius: 6px;
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
}
textarea{
    height:120px;
}
.text-footer{
    text-align: center;
    padding: 20px 0;
    font-family: 'ubuntu',sans-serif;
    display: flex;
    justify-content: center;
    color: white;
}
```

```
.burger{
    display: none;
position: absolute;
cursor:pointer;
right: 5%;
top: 15px;
}
.line{
    width: 33px;
    background-color: white;
    height: 4px;
    margin: 5px 3px;
}
@media only screen and (max-width:973px){
    .nav-list{
        flex-direction: column;
        transition: all 0.7s ease-out;
    }
    .navbar{
        flex-direction: column;
        height: 470px;
    }
    .rightnav{
        text-align: center;
    }
    .box-main{
        flex-direction: column-reverse;
        max-width: 100%;
    }
    #search{
        width: 100%;
    }
    .burger{
        display:block;
```

```

    }
    .h-nav-resp{
        height:72px;
    }
    .v-class-resp{
        opacity: 0;
    }
    .section{
        flex-direction: column-reverse;
    }
    .logo img {
        width: 19px;
        height: 18px;
    }
}

//responsive javascript code
burger =document.querySelector('.burger')
navbar =document.querySelector('.navbar')
navlist =document.querySelector('.nav-list')
rightnav =document.querySelector('.rightnav')
burger.addEventListener('click', ()=>{
    rightnav.classList.toggle('v-class-resp');
    navlist.classList.toggle('v-class-resp');
    navbar.classList.toggle('h-nav-resp');
})

<!doctype html>
<head>
    <link rel="stylesheet" type="text/css" href="css/programming.css">
    <title>Programming Courses</title>
</head>
<body>
    <center><div><h1>Programming Courses</h1></div></center>
    <div class="container">

```

```

<div class="row">
  <div class="column">
    <div class="main">
      <div class="service">
        <div class="service-logo"><i></i>
        </div>
        <a href="c++.html" ><h1>C++</h1></a>
        <p>C++ is used for graphics-heavy software such as games, photo and video
editing apps.</p>
      </div>
    </div>
  </div>
  <div class="column">
    <div class="main">
      <div class="service">
        <div class="service-logo"><i></i>
        </div>
        <a href="cs.html"><h1>C#</h1></a>
        <p>C# is used for developing desktop applications, web applications,
web services and game development in Unity.</p>
      </div>
    </div>
  </div>
  <div class="column">
    <div class="main">
      <div class="service">
        <div class="service-logo"><i></i>
        </div>
        <a href="C.html" ><h1>Programming in C</h1></a>
        <p>C language is used to develop system applications that forms
major portion of operating systems such as Windows, UNIX and

```

Linux.</p>

</div>

</div>

</div>

</div>

</div>

</body>

/*stylesheet for programming course main page*/

body{

background:lightsteelblue;

padding: 0px;

margin: 0px;

}

.main {

width: 100%;

min-height: 380px;

text-align: center;

position: relative;

cursor: pointer;

}

.service {

background: #e68585;

padding: 4px 40px 21px;

border-radius: 30px;

position: absolute;

box-shadow: 0 0 25px -5px rgb(253 253 253);

transition: transform .8s;

height: 350px;

width: 190px;

margin: -52px 5px 15px -130px;

margin-bottom: 20px;

color: #fff;

}

```
.service h1 {
    height: 35px;
    width: 80%;
    margin: 50px auto;
    position: relative;
}

.row {
    display: flex;
    justify-content: space-around;
    align-items: center;
    margin-top: 180px;
}
```

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Google font cdn link -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link      href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
rel="stylesheet">
    <!-- font awesome cdn link -->
    <link      rel="stylesheet"      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <!-- Custom CSS file link -->
    <link rel="stylesheet" href="css/course.css">
    <title>C++</title>
</head>
<body>
    <div class="container-xxl my-md-4 bd-layout">
        <aside class="bd-sidebar">
```

```

<nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
  <ul class="list-unstyled mb-0 py-3 pt-md-1">
    <li class="mb-1">
      <button class="btn d-inline-flex align-items-center rounded collapsed"
data-bs-toggle="collapse"
      data-bs-target="#getting-started-collapse" aria-expanded="false">
        C++
      </button>
      <div class="collapse" id="getting-started-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">
          <li><a href="c++.html" target="_top" class="active"
class="d-inline-flex align-items-center rounded">Introduction</a></li>
          <li><a href="C++ Syntax.html" target="_top"
class="d-inline-flex align-items-center rounded">Syntax</a></li>
          <li><a href="#" class="d-inline-flex align-items-center rounded">Output
&amp;
          Variables</a></li>
          <li><a href="#" class="d-inline-flex align-items-center
rounded">String</a></li>
          <li><a href="#" class="d-inline-flex align-items-center
rounded">Loops</a></li>
          <li><a href="#" class="d-inline-flex align-items-center
rounded">Array</a></li>
          <li><a href="#" class="d-inline-flex align-items-center
rounded">Pointers</a></li>
        </ul>
      </div>
      <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
      data-bs-target="#function-collapse" aria-expanded="false">
        Function
      </button>
      <div class="collapse" id="function-collapse">

```



```

        <ul class="list-unstyled fw-normal pb-1 small">
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Function</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Function Parameters</a>
            </li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Function
                Overloading</a></li>
        </ul>
    </div>
    <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
        data-bs-target="#C++-Classes" aria-expanded="false">
        Getting started
    </button>
    <div class="collapse" id="C++-Class-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">
            <li><a href="#" class="d-inline-flex align-items-center rounded">C++
OOP</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Classes &
                Objects</a></li>
            <li><a href="#" class="d-inline-flex align-items-center rounded">Class
Methods</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Encapsulation</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Abstraction</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Polymorphism</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Inheritance</a></li>
        </ul>

```

```

        </div>
    </li>
</ul>
</nav>
</aside>
<main class="bd-main order-1" id="intro">
    <div class="bd-intro ps-lg-4">
        <div class="d-md-flex flex-md-row-reverse align-items-center justify-content-between">
            <h1 class="bd-title" id="content">"Introduction to C++"
                <a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#"
href="#content"
                    style="padding-left: 0.375em;">
            </h1>
            <div class="nextprev">
                <a class="left wntd" href="programming.html"><strong><
Home</strong></a>
                <a class="right wntd" href="C++ Syntex.html"><strong>Next ></strong></a>
            </div>
        </div>
        <p class="bd-lead">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.</p>
    </div>

```

```

<div class="bd-toc mt-4 mb-5 my-md-0 ps-xl-3 mb-lg-5 text-muted">
  <strong class="d-block h6 my-2 pb-2 border-bottom">What's On this
page</strong>
  <nav id="TableOfContents" style="font-size: 15px;">
    <ul>
      <li><a href="#whatiscplusplus">What is C++</a></li>
      <li><a href="#whyusecplusplus">Why Use C++</a></li>
      <li><a href="#getstarted">Get Started C++</a></li>
      <li><a href="#ide">C++ Install IDE</a></li>
      <li><a href="#qustart">Quick Start</a></li>
    </ul>
  </nav>
</div>
<div class="bd-content ps-lg-4">
  <h2 id="whatiscplusplus">"What is C++?"
    <a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#"
href="#whatiscplusplus"
      style="padding-left: 0.375em;">
    </a>
  </h2>
  <p>C++ is a cross-platform language that can be used to create high-performance
    applications. <br> C++ was developed by Bjarne Stroustrup, as an extension to
the C
    language. <br> C++ gives programmers a high level of control over system
resources and
    memory.</p>
  <hr>
  <h2 id="whyusecplusplus">"Why Use C++"
    <a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#"
href="#whyusecplusplus"
      style="padding-left: 0.375em;">
    </a>
  </h2>
  <p>C++ is one of the world's most popular programming languages. <br> C++ can

```

be found in today's

operating

systems, Graphical User Interfaces, and embedded systems.
 C++ is an object-oriented programming

language which gives a clear structure to programs and allows code to be reused, lowering

development

costs.
 C++ is portable and can be used to develop applications that can be adapted to multiple

platforms.
 C++ is fun and easy to learn! </p>

<hr>

<h2 id="getstarted">"Get Started C++"

</h2>

<p>To start using C++, you need two things
 A text editor, like Notepad, to write C++ code

A compiler, like GCC, to translate the C++ code into a language that the computer will understand

 There are many text editors and compilers to choose from. In this tutorial, we will use an IDE

(see below).

</p>

<hr>

<h2 id="ide">"Install IDE"

</h2>

<p>An IDE (Integrated Development Environment) is used to edit AND compile the code.
 Popular IDE's

include Sublime Text, Eclipse, and Visual Studio. These are all free, and they can be used to both

edit and debug C++ code.</p>

<p>

Note:

Web-based IDE's can work as well, but functionality is limited.
 We will use Sublime Text in our tutorial, which we believe is a good place to start.

</p>

<p>You can find the latest version of Sublime Text at <a href="https://www.sublimetext.com/download"

target="_blank">http://www.codeblocks.org/downloads/26.

Download the <code class="codespan">mingw-setup.exe</code> file, which will install the text

editor with

a compiler. </p>

<hr>

<h2 id="qustart">"Quick Start"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#qustart"

style="padding-left: 0.375em;">

</h2>

<p>Let's create our first C++ file.
 Open Codeblocks and go to File > New > Empty

File.
 Write the following C++ code and save the file as

<code>myfirstprogram.cpp </code> (File > Save File as):

</p>

<div class="example">

<h4>myfirstprogram.cpp</h4>

<div class="code notranslate javaHigh">

#include<iostream>;

```

<br>
using namespace std;
<br>
<br>
<span class="javakeywordcolor" style="color:mediumblue">int</span>
main() {
<br>
    &nbsp;
    <span class="javanumbercolor" style="color:red">
</span>
    cout<&lt;&lt;
    <span class=" javastringcolor" style="color:brown">"Hello World!"</span>
    ;
    <br>
    &nbsp;
    <span class="javakeywordcolor" style="color:mediumblue">return</span>
    <span class="javanumbercolor" style="color:red">0</span>
    ;
    <br>
    }
</span>
</div>

```

</div>

<p>In Sublime Text, The code should look like this:</p>

<p>Then, go to Tools > Build to run (execute) the program,
and The Shortcut Key to

run the program is Ctrl+Shift+B. The result will look
something to this:</p>

<p>Output of The Program.</p>

</div>

<div class="nextprev">

```

        <a class="left wntd" href="programming.html"><strong>< Home</strong></a>

        <a class="right wntd" href="C++ Syntax.html"><strong>Next ></strong></a>
    </div>
</main>
</div>
<!-- JQuery cdn link -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</body>
</html>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Google font cdn link -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link      href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
rel="stylesheet">
    <!-- font awesome cdn link -->
    <link      rel="stylesheet"      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <!-- Custom CSS file link -->
    <link rel="stylesheet" href="css/course.css">
    <title>C++</title>
</head>
<body>
    <div class="container-xxl my-md-4 bd-layout">
        <aside class="bd-sidebar">
            <nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
                <ul class="list-unstyled mb-0 py-3 pt-md-1">
                    <li class="mb-1">

```

```

        <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
        data-bs-target="#getting-started-collapse" aria-expanded="false">
        C++
    </button>
    <div class="collapse" id="getting-started-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">>
            <li><a href="c++.html" target="_top"
                class="d-inline-flex                                align-items-center
rounded">Introduction</a></li>
            <li><a href="C++ Syntax.html" target="_top" class="active"
                class="d-inline-flex align-items-center rounded">Syntax</a></li>
            <li><a href="OutIn.html" class="d-inline-flex align-items-center
rounded">Output &
                Input</a></li>
            <li><a href="var.html" class="d-inline-flex align-items-center
rounded">Variables</a>
            </li>
            <li><a href="str.html" class="d-inline-flex align-items-center
rounded">String</a></li>
            <li><a href="lop.html" class="d-inline-flex align-items-center
rounded">Loops</a></li>
            <li><a href="fun.html" class="d-inline-flex align-items-center
rounded">Function</a>
            </li>
        </ul>
    </div>
    <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
        data-bs-target="#C++-Classes" aria-expanded="false">
        Class & Object
    </button>
    <div class="collapse" id="C++-Class-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">

```



```

                <li><a href="oop.html" class="d-inline-flex align-items-center
rounded">C++ OOP</a></li>
                <li><a href="clob.html" class="d-inline-flex align-items-center
rounded">Classes &
Objects</a></li>
                <li><a href="clm.html" class="d-inline-flex align-items-center
rounded">Class
Methods</a></li>
                <li><a href="en.html" class="d-inline-flex align-items-center
rounded">Encapsulation</a>
                </li>
                <li><a href="abs.html" class="d-inline-flex align-items-center
rounded">Abstraction</a>
                </li>
                <li><a href="poly.html"
class="d-inline-flex align-items-center
rounded">Polymorphism</a></li>
                <li><a href="in.html" class="d-inline-flex align-items-center
rounded">Inheritance</a>
                </li>
        </ul>
        <hr>
        <hr>
        <hr>
        <hr>
        </div>
    </li>
</ul>
</nav>
</aside>
<main class="bd-main order-1" id="sntx">
    <div class="bd-intro ps-lg-4">
        <div class="d-md-flex flex-md-row-reverse align-items-center justify-content-
between">

```

```

<h1 class="bd-title" id="syntex">"Syntex of C++?"
  <a class="anchorjs-link" aria-label="Anchor" data-anchorjs-icon="#"
href="#syntex"
  style="padding-left: 0.375em;">
</h1>
<div class="nextprev">
  <a class="left wntd" href="c++.html"><strong>◀ Previous</strong></a>
  <a class="right wntd" href="#"><strong>Next ▶</strong></a>
</div>
</div>
<hr>
<p class="bd-lead">Learning C++ programming can be simplified into: <br>
  &nbsp; &nbsp; &nbsp; <strong>1.</strong> Writing your program in a text
editor and saving it with
  correct extension(.CPP, .C, .CP) <br>
  &nbsp; &nbsp; &nbsp; <strong>2.</strong> Compiling your program using a
compiler or online IDE. <br>
  &nbsp; &nbsp; &nbsp; <strong>3.</strong> Understanding the basic
terminologies. </p>
</div>
<div class="bd-toc mt-4 mb-5 my-md-0 ps-xl-3 mb-lg-5 text-muted">
  <strong class="d-block h6 my-2 pb-2 border-bottom">What's On this
page</strong>
  <nav id="TableOfContents" style="font-size: 15px;">
    <ul>
      <li><a href="#syntex">Syntex of C++</a></li>
      <li><a href="#c++1">Syntex</a></li>
      <li><a href="#c++2">Example Explained</a></li>
      <li><a href="#c++3">Omitting Namespace</a></li>
    </ul>
  </nav>
</div>
<div class="bd-content ps-lg-4">

```

Syntext

[Anchor](#c++1)

style="padding-left: 0.375em;">  The “Hello World” program is the first step towards learning any programming language and also one of the simplest programs you will learn. All you have to do is display the message “Hello World” on the screen. Let us now look at the program:

myfirstprogram.cpp

<div class="code notranslate javaHigh">
#include<iostream>

using namespace std;

main() {

cout<<<

;

```

        <br>
    }
</span>
</div>
</div>
<div class="op123">
    <b>Output:</b>
    <pre>Hello EveryOne</pre>
</div>
<hr>
<h2 id="c++2">"Example Explained"
    <a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#"
href="#c++2"
        style="padding-left: 0.375em;">
    </a>
</h2>
<p>Let us now understand every line and the terminologies of the above
program:</p>
<ol class="bd-leadl">
    <li><strong>// Simple C++ program to display “Hello World”</strong> : This
line is a comment line. A comment is used to display additional information about the
program. A comment does not contain any programming logic. When a comment is
encountered by a compiler, the compiler simply skips that line of code. Any line beginning
with ‘//’ without quotes OR in between /*...*/ in C++ is comment.<br></li><br>
    <li><strong>#include</strong>: In C++, &nbsp;all lines that start with pound (#)
sign are called directives and are processed by preprocessor which is a program invoked by
the compiler. The <strong>#include</strong> directive tells the compiler to include a file and
<strong>#include<i> <code><strong>#include<i> <code>iostream<i> <code>.</strong></strong></strong>. It tells the compiler to include the
standard iostream file&nbsp;which contains declarations of all the standard input/output
library
        functions.<br>
    </li><br>
    <li><strong>using namespace std</strong>: This is used to import the entirety of
the std namespace into the current namespace of the program. The statement using

```


[Anchor](#c++3)

style="padding-left: 0.375em;">

</h2>

 You might see some C++ programs that runs without the standard namespace

library. The

`using namespace std` line can be omitted and replaced with the `std`

keyword,

followed by the `::` operator

for some objects. As you can see in the examples below.

</p>

` Example 1. `

style="color:#88c999">

 #include <iostream>

<span class="javakeywordcolor"

style="color:#88c999">int main() {
 <span class="javanumbercolor"

style="color:#88c999">

 std::cout <<<span class="javastringcolor"

style="color:#88c999">"Hello World!"
 <span class="javakeywordcolor"

style="color:#88c999">return <span class="javanumbercolor"

style="color:#88c999">0
}

</div>

<div class="op123">

Output:

<pre>Hello World!</pre>

</div>

` Example 2. `

<div class="w3-code nottranslate javaHigh"><span

```

class="javanumbercolor"
        style="color:#88c999">
    </span> #include <iostream><br>using namespace std;<br><span
class="javakeywordcolor"
        style="color:#88c999">int</span>    main()    {<br>&nbsp;    <span
class="javanumbercolor"
        style="color:#88c999">
    </span>cout    &lt;&lt;    <span    class="javastringcolor"
style="color:#88c999">"Hello
        World!"</span>;<br>&nbsp;    <span    class="javakeywordcolor"
style="color:#88c999">return</span>
        <span    class="javanumbercolor"    style="color:#88c999">0</span>;<br>}
</span>
</div>
<div class="op123">
    <b>Output:</b>
    <pre>Hello World!</pre>
</div>
</div>
<div class="nextprev">
    <a class="left wntd" href="c++.html"><strong>< Previous</strong></a>
    <a class="right wntd" href="#"><strong>Next ></strong></a>
</div>
</main>
</div>
<!-- JQuery cdn link -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</body>
</html>

<!--C#-->
<!DOCTYPE html>
<html lang="en">

```

```

<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Google font cdn link -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
rel="stylesheet">
  <!-- font awesome cdn link -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
  <!-- Custom CSS file link -->
  <link rel="stylesheet" href="css/course.css">
  <title>C#</title>
</head>
<body>
  <div class="container-xxl my-md-4 bd-layout">
    <aside class="bd-sidebar">
      <nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
        <ul class="list-unstyled mb-0 py-3 pt-md-1">
          <li class="mb-1">
            <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
            data-bs-target="#getting-started-collapse" aria-expanded="false">
              C#
            </button>
            <div class="collapse" id="getting-started-collapse">
              <ul class="list-unstyled fw-normal pb-1 small">
                <li><a href="cs.html" target="_top" class="active"
class="d-inline-flex align-items-center
rounded">Introduction</a></li>
                <li><a href="prostr.html" class="d-inline-flex align-items-center
rounded">Program

```



```

                Structure</a></li>
        <li><a href="snt.html" class="d-inline-flex align-items-center
rounded">Syntax</a></li>
        <li><a href="#" class="d-inline-flex align-items-center rounded">Output
&amp;
                Input</a></li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Variables</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">String</a></li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Loops</a></li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Function</a>
        </li>
</ul>
</div>
<button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
        data-bs-target="#C++-Classes" aria-expanded="false">
        Class & Object
</button>
<div class="collapse" id="C++-Class-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">
                <li><a href="#" class="d-inline-flex align-items-center rounded">C#
OOP</a></li>
                <li><a href="#" class="d-inline-flex align-items-center
rounded">Classes &amp;
                Objects</a></li>
                <li><a href="#" class="d-inline-flex align-items-center rounded">Class
                Methods</a></li>
                <li><a href="#" class="d-inline-flex align-items-center
rounded">Encapsulation</a>

```

```

        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Abstraction</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Polymorphism</a></li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Inheritance</a>
        </li>
    </ul>
    <hr>
    <hr>
    <hr>
    <hr>
    </div>
</li>
</ul>
</nav>
</aside>
<main class="bd-main order-1" id="sntx">
    <div class="bd-intro ps-lg-4">
        <div class="d-md-flex flex-md-row-reverse align-items-center justify-content-
between">
            <h1 class="bd-title" id="syntex">"Introduction to C#"
            <a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#"
href="#syntex"
                style="padding-left: 0.375em;">
            </h1>
            <div class="nextprev">
                <a class="left wntd" href="programming.html"><strong>◀
Home</strong></a>
                <a class="right wntd" href="prostr.html"><strong>Next ▶</strong></a>
            </div>

```

</div>

<hr>

<p>C# is a simple, modern, general-purpose, object-oriented programming language developed by Microsoft

within its .NET initiative led by Anders Hejlsberg. This tutorial will teach you basic C#

programming and will also take you through various advanced concepts related to C# programming

language.
This tutorial has been prepared for the beginners to help them understand basic C#

programming.
C# programming is very much based on C and C++ programming languages, so if you

have a basic understanding of C or C++ programming, then it will be fun to learn C#.</p>

</div>

<div class="bd-toc mt-4 mb-5 my-md-0 ps-xl-3 mb-lg-5 text-muted">

<strong class="d-block h6 my-2 pb-2 border-bottom">What's On this page

<nav id="TableOfContents" style="font-size: 15px;">

Introduction to C#

What is C#?

Features of C#

Environment of C#

IDE for C#

</nav>

</div>

<div class="bd-content ps-lg-4">

<h2 id="c++1">"What is C#?"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#c++1"

style="padding-left: 0.375em;">

</h2>

<p>C# is a modern, general-purpose, object-oriented programming language developed by Microsoft and

approved by European Computer Manufacturers Association (ECMA) and International Standards

Organization (ISO).
C# was developed by Anders Hejlsberg and his team during the development of

.Net Framework.
 C# is designed for Common Language Infrastructure (CLI), which consists of the

executable code and runtime environment that allows use of various high-level languages on different

computer platforms and architectures.</p>

<p>The following reasons make C# a widely used professional language –</p>

<ul class=bd-leadl>

It is a modern, general-purpose programming language

It is object oriented.

It is component oriented.

It is easy to learn.

It is a structured language.

It produces efficient programs.

It can be compiled on a variety of computer platforms.

It is a part of .Net Framework.

<hr>

<h2 id="c++2">"Features of C#"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#c++2"

</h2>

<p>Although C# constructs closely follow traditional high-level languages, C and C++ and being an

object-oriented programming language. It has strong resemblance with Java, it has numerous strong

programming features that make it endearing to a number of programmers worldwide.

Following is the list of few important features of C#

- Boolean Conditions
- Automatic Garbage Collection
- Standard Library
- Assembly Versioning
- Properties and Events
- Delegates and Events Management
- Easy-to-use Generics
- Indexers
- Conditional Compilation
- Simple Multithreading
- LINQ and Lambda Expressions
- Integration with Windows

Environment of C#

[Anchor](#c++3)

In this chapter, we will discuss the tools required for creating C# programming. We have already

mentioned that C# is part of .Net framework and is used for writing .Net applications. Therefore,

before discussing the available tools for running a C# program, let us understand how C# relates to

the .Net framework.

The .Net Framework

The .Net framework is a revolutionary platform that helps you to write the following types of

applications –</p>

<ul class="bd-leadl">

Windows applications

Web applications

Web services

<p>The .Net framework applications are multi-platform applications. The framework has been designed in

such a way that it can be used from any of the following languages: C#, C++, Visual Basic, Jscript,

COBOL, etc. All these languages can access the framework as well as communicate with each other.</p>

<p>The .Net framework consists of an enormous library of codes used by the client languages such as C#.

Following are some of the components of the .Net framework –</p>

<ul class="bd-leadl">

Common Language Runtime (CLR)

The .Net Framework Class Library

Common Language Specification

Common Type System

Metadata and Assemblies

Windows Forms

ASP.Net and ASP.Net AJAX

ADO.Net

Windows Workflow Foundation (WF)

Windows Presentation Foundation

Windows Communication Foundation (WCF)

LINQ

<hr>

<h2 id="c++4">"Integrated Development Environment (IDE) for C#">

style="padding-left: 0.375em;">

</h2>

<p>Microsoft provides the following development tools for C# programming

<ul class="bd-lead">

Visual Studio 2010 (VS)

Visual C# 2010 Express (VCE)

Visual Web Developer

<p>The last two are freely available from Microsoft official website. Using these tools, you can write

all kinds of C# programs from simple command-line applications to more complex applications. You can

also write C# source code files using a basic text editor, like Notepad, and compile the code into

assemblies using the command-line compiler, which is again a part of the .NET Framework.
Visual

C# Express and Visual Web Developer Express edition are trimmed down versions of Visual Studio and

has the same appearance. They retain most features of Visual Studio. In this tutorial, we have used

Visual C# 2010 Express.
You can download it from <a target="_blank" rel="nofollow"

href="https://www.microsoft.com/visualstudio/eng/downloads">Microsoft Visual Studio. It gets

installed automatically on your machine.</p>

<p>Note: You need an active internet connection for installing the express edition.</p>

</div>

<div class="nextprev">

◀ Home

Next ▶

</div>

```

        </main>
    </div>
    <!-- JQuery cdn link
    <script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
-->
</body>
</html>
<!--C# second content-->
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <!-- Google font cdn link -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link      href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
rel="stylesheet">
    <!-- font awesome cdn link -->
    <link      rel="stylesheet"      href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
    <!-- Custom CSS file link -->
    <link rel="stylesheet" href="css/course.css">
    <title>C#</title>
</head>
<body>
    <div class="container-xxl my-md-4 bd-layout">
        <aside class="bd-sidebar">
            <nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
                <ul class="list-unstyled mb-0 py-3 pt-md-1">
                    <li class="mb-1">
                        <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"

```



```

        data-bs-target="#getting-started-collapse" aria-expanded="false">
        C#
    </button>
    <div class="collapse" id="getting-started-collapse">
        <ul class="list-unstyled fw-normal pb-1 small">
            <li><a href="cs.html" class="d-inline-flex align-items-center
rounded">Introduction</a>
            </li>
            <li><a href="prostr.html" target="_top" class="active"
class="d-inline-flex align-items-center rounded">Program
Structure</a></li>
            <li><a href="snt.html" class="d-inline-flex align-items-center
rounded">Syntax</a>
            </li>
            <li><a href="#" class="d-inline-flex align-items-center rounded">Output
&amp;
                Input</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Variables</a>
            </li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">String</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Loops</a></li>
            <li><a href="#" class="d-inline-flex align-items-center
rounded">Function</a>
            </li>
        </ul>
    </div>
    <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
        data-bs-target="#C#-Classes" aria-expanded="false">
        Class & Object
    </button>

```

```

<div class="collapse" id="C#-Class-collapse">
  <ul class="list-unstyled fw-normal pb-1 small">
    <li><a href="#" class="d-inline-flex align-items-center rounded">C#
OOP</a></li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Classes &
Objects</a></li>
    <li><a href="#" class="d-inline-flex align-items-center rounded">Class
Methods</a></li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Encapsulation</a>
</li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Abstraction</a>
</li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Polymorphism</a></li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Inheritance</a>
</li>
  </ul>
  <hr>
  <hr>
  <hr>
  <hr>
</div>
</li>
</ul>
</nav>
</aside>
<main class="bd-main order-1" id="sntx">
  <div class="bd-intro ps-lg-4">
    <div class="d-md-flex flex-md-row-reverse align-items-center justify-content-
between">

```

```

<h1 class="bd-title" id="syntax">"Program Structure in C#"
  <a class="anchorjs-link" aria-label="Anchor" data-anchorjs-icon="#"
href="#syntax"
    style="padding-left: 0.375em;">
</h1>
<div class="nextprev">
  <a class="left wntd" href="cs.html"><strong>◀ Previous</strong></a>
  <a class="right wntd" href="#"><strong>Next ▶</strong></a>
</div>
</div>
<hr>
<p>Before we study basic building blocks of the C# programming language, let us
look at a bare minimum
  C# program structure so that we can take it as a reference in upcoming
chapters.</p>
</div>
<div class="bd-toc mt-4 mb-5 my-md-0 ps-xl-3 mb-lg-5 text-muted">
  <strong class="d-block h6 my-2 pb-2 border-bottom">What's On this
page</strong>
  <nav id="TableOfContents" style="font-size: 15px;">
    <ul>
      <li><a href="#syntax">Program Structure in C#</a></li>
      <li><a href="#c++1">Creating Hello World Program</a></li>
      <li><a href="#c++2">Compiling and Executing the Program</a></li>
    </ul>
  </nav>
</div>
<div class="bd-content ps-lg-4">
  <h2 id="c++1">"Creating Hello World Program"
    <a class="anchorjs-link" aria-label="Anchor" data-anchorjs-icon="#"
href="#c++1"
      style="padding-left: 0.375em;">
    </a>

```

</h2>

<p>A C# program consists of the following parts –</p>

<ul class="bd-leadl">

Namespace declaration

A class

Class methods

Class attributes

A Main method

Statements and Expressions

Comments

<p>Let us look at a simple code that prints the words "Hello World" –</p>

<div>

<div class="op123">

<h3>Example</h3>

<p>When this code is compiled and executed, it produces the following result

–</p>

Output

<pre>Hello World </pre>

</div>

</div>

<p>Let us look at the various parts of the given program –</p>

<ul class="bd-leadl">

<p>The first line of the program using System; - the using keyword is used to

include the System namespace in the program. A program generally has multiple

using statements.</p>

<p>The next line has the namespace declaration. A

namespace is a collection of classes. The *HelloWorldApplication* namespace contains the class *HelloWorld*.

The next line has a **class** declaration, the class *HelloWorld* contains the data

and method definitions that your program uses. Classes generally contain multiple methods.

Methods define the behavior of the class. However, the *HelloWorld* class has only one

method **Main**.

The next line defines the **Main** method, which is the **entry point** for all C#

programs. The **Main** method states what the class does when executed.

The next line */*...*/* is ignored by the compiler and it is put to add **comments** in the

program.

The **Main** method specifies its behavior with the statement **Console.WriteLine("Hello**

World");

WriteLine is a method of the *Console* class defined in the *System*

namespace. This statement causes the message "Hello, World!" to be displayed on the screen.

</p>

<p></p>

<p>The last line Console.ReadKey(); is for the VS.NET Users. This makes the program wait

for a key press and it prevents the screen from running and closing quickly when the program

is launched from Visual Studio .NET.</p>

<p>It is worth to note the following points —</p>

<ul class="bd-leadl">

<p>C# is case sensitive.</p>

<p>All statements and expression must end with a semicolon (;).</p>

<p>The program execution starts at the Main method.</p>

<p>Unlike Java, program file name could be different from the class name.</p>

<hr>

<h2 id="c++2">"Compiling and Executing the Program"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#c++2"

style="padding-left: 0.375em;">

</h2>

<p>If you are using Visual Studio.Net for compiling and executing C# programs, take the following steps

–</p>

<ul class="bd-leadl">

<p>Start Visual Studio.</p>

<p>On the menu bar, choose File -> New -> Project.</p>

<p>Choose Visual C# from templates, and then choose Windows.</p>

<p>Choose Console Application.</p>

<p>Specify a name for your project and click OK button.</p>

<p>This creates a new project in Solution Explorer.</p>

<p>Write code in the Code Editor.</p>

<p>Click the Run button or press F5 key to execute the project. A Command Prompt window appears

that contains the line Hello World.</p>

<p>You can compile a C# program by using the command-line instead of the Visual Studio IDE –</p>

<ul class="bd-leadl">

- <p>Open a text editor and add the above-mentioned code.</p>
- <p>Save the file as helloworld.cs</p>
- <p>Open the command prompt tool and go to the directory where you saved the file.</p>
- <p>Type csc helloworld.cs and press enter to compile your code.</p>
- <p>If there are no errors in your code, the command prompt takes you to the next line and generates helloworld.exe executable file.</p>
- <p>Type helloworld to execute your program.</p>
- <p>You can see the output Hello World printed on the screen.</p>

```
</div>
<div class="nextprev">
  <a class="left wntd" href="cs.html"><strong>◀ Previous</strong></a>
  <a class="right wntd" href="#.html"><strong>Next ▶</strong></a>
</div>
</main>
</div>
<!-- JQuery cdn link -->
<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>
</body>
</html>
<!--C programming introduction-->
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta http-equiv="X-UA-Compatible" content="IE=edge">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <!-- Google font cdn link -->
  <link rel="preconnect" href="https://fonts.googleapis.com">
  <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
  <link      href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
```



```

rel="stylesheet">
  <!-- font awesome cdn link -->
  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
  <!-- Custom CSS file link -->
  <link rel="stylesheet" href="css/course.css">
  <title>C</title>
</head>
<body>
  <div class="container-xxl my-md-4 bd-layout">
    <aside class="bd-sidebar">
      <nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
        <ul class="list-unstyled mb-0 py-3 pt-md-1">
          <li class="mb-1">
            <button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
            data-bs-target="#getting-started-collapse" aria-expanded="false">
              C
            </button>
            <div class="collapse" id="getting-started-collapse">
              <ul class="list-unstyled fw-normal pb-1 small">
                <li><a href="c.html" target="_top" class="active"
                    class="d-inline-flex align-items-center rounded">Introduction</a></li>
                <li><a href="c_1.html" class="d-inline-flex align-items-center
                    rounded">Elements of C</a>
                </li>
                <li><a href="#" class="d-inline-flex align-items-center rounded">Output
                    &
                    Input</a></li>
                <li><a href="#" class="d-inline-flex align-items-center
                    rounded">Variables</a>
                </li>
                <li><a href="#" class="d-inline-flex align-items-center

```

```

rounded">String</a></li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Array</a></li>
    <li><a href="#" class="d-inline-flex align-items-center
rounded">Loops</a></li>
</ul>
</div>
<button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
    data-bs-target="#C++-Classes" aria-expanded="false">
    Functions
</button>
<div class="collapse" id="C++-Class-collapse">
    <ul class="list-unstyled fw-normal pb-1 small">
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Library Functions</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center rounded">User
Defined
            Functions</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Function Definition</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Function Call</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center rounded">Return
Function</a>
        </li>
        <li><a href="#" class="d-inline-flex align-items-center
rounded">Parameter &
            Argument</a>
        </li>

```

Declaration

</div>

data-bs-target="#C++-Classes" aria-expanded="false">

Recursion

</button>

<div class="collapse" id="C++-Class-collapse">

<ul class="list-unstyled fw-normal pb-1 small">

Function

Recursive Function

Recursion

Recursion


```

        </ul>
        <hr>
        <hr>
        <hr>
        <hr>
    </div>
</li>
</ul>
</nav>
</aside>
<main class="bd-main order-1" id="sntx">
    <div class="bd-intro ps-lg-4">
        <div class="d-md-flex flex-md-row-reverse align-items-center justify-content-between">
            <h1 class="bd-title" id="syntex">"Introduction to C"
                <a class="anchorjs-link" aria-label="Anchor" data-anchorjs-icon="#"
href="#syntex"
                    style="padding-left: 0.375em;">
            </h1>
            <div class="nextprev">
                <a class="left wntd" href="programming.html"><strong><
Home</strong></a>
                <a class="right wntd" href="#"><strong>Next ></strong></a>
            </div>
        </div>
        <hr>
        <p>C is a procedural programming language. It was initially developed by Dennis
Ritchie in the year
            1972. It was mainly developed as a system programming language to write an
operating system. The
            main features of the C language include low-level memory access, a simple set
of keywords, and a
            clean style, these features make C language suitable for system programmings

```

like an operating

system or compiler development.
Many later languages have borrowed syntax/features directly

or indirectly from the C language. Like syntax of Java, PHP, JavaScript, and many other languages

are mainly based on the C language. C++ is nearly a superset of C language (Few programs may compile

in C, but not in C++). </p>

</div>

<div class="bd-toc mt-4 mb-5 my-md-0 ps-xl-3 mb-lg-5 text-muted">

<strong class="d-block h6 my-2 pb-2 border-bottom">What's On this page

<nav id="TableOfContents" style="font-size: 15px;">

Introduction to C

Beginning with C Programming

Writing First Program

</nav>

</div>

<div class="bd-content ps-lg-4">

<h2 id="c++1">"Beginning with C Programming:"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#c++1"

style="padding-left: 0.375em;">

</h2>

<ol class="bd-leadl">

Structure of a C program
After the above discussion, we can formally

assess the structure of a C program. By structure, it is meant that any program can be written

in this structure only. Writing a C program in any other structure will hence lead to a

Compilation Error.
The structure of a C program is as follows:

<ol class="bd-leadl">

<p>The components of the above structure are: </p>

Header Files Inclusion: The first and foremost component is the

inclusion of the Header files in a C program.
A header file is a file with

extension .h which contains C function declarations and macro definitions to be shared

between several source files.
Some of C Header files:

stddef.h – Defines several useful types and macros.

stdint.h – Defines exact width integer types.

stdio.h – Defines core input and output functions

stdlib.h – Defines numeric conversion functions, pseudo-random network

generator, memory allocation

string.h – Defines string handling functions

math.h – Defines common mathematical functions

Main Method Declaration: The next part of a C program is to declare the

main() function. The syntax to declare the main function is:
Syntax to

Declare the main method:


```
<pre>int main()<br>{ }</pre>
```

<ol class="bd-leadl">

Variable Declaration: The next part of any C program is the variable

declaration. It refers to the variables that are to be used in the function. Please note that in

the C program, no variable can be used without being declared. Also in a C program, the

variables are to be declared before any operation in the

function.
Example:


```
<pre>int                                main()<br>{<br>&nbsp;&nbsp;<strong>int<br>a;</strong><br>.<br>.</pre>
```

<ol class="bd-leadl">

Body: The body of a function in the C program, refers to the operations that

are performed in the functions. It can be anything like manipulations, searching, sorting,

printing, etc.
Example:


```
<pre>int                                main()<br>{<br>&nbsp;&nbsp;<strong>int<br>a;</strong><br><br>&nbsp;&nbsp;<strong>printf("%d", a);</strong><br>.<br>.</pre>
```

<ol class="bd-leadl">

Return Statement: The last part of any C program is the return statement. The

return statement refers to the returning of the values from a function. This return statement

and return value depend upon the return type of the function. For example, if the return type is

void, then there will be no return statement. In any other case, there will be a return

statement and the return value will be of the type of the specified return

type.
Example:


```

</ol>
<pre>int main() {
    int a;
    printf("%d", a);
return 0;
}</pre>
<hr>
<h2 id="c++2">Writing First Program
    <a class="anchorjs-link" aria-label="Anchor" data-anchorjs-icon="#"
href="#c++2"
style="padding-left: 0.375em;">
</a>
</h2>
<ol class="bd-lead1">
    <li><strong>First Program:</strong>&nbsp;<br>Following is First Program in
C<br>&nbsp;</li>
</ol>
<div class="op123">
    <h3>Example</h3>
    
    <br>
    <b>Output</b><br>
    
</div>
<ol class="bd-lead1">
    <p><strong>Let us analyze the program line by line.</strong></p>
    <li><i><strong>Line 1: [ #include &lt;stdio.h>
]</strong></i> In a C program, all lines that start with <strong>#
</strong>are
        processed by a preprocessor which is a program
        invoked by the compiler. In a very basic term, the preprocessor takes a C
        program and
        produces another C program. The produced program has no lines starting with
        #, all such lines
        are processed by the preprocessor. In the above example, the preprocessor
        copies the
        preprocessed code of stdio.h to our file. The .h files are called header files in

```


C. These

header files generally contain declarations of functions. We need `stdio.h` for the function

`printf()` used in the program.
Line 2 `[int main(void)`
] There

must be a starting point from where execution of compiled C program begins. In C, the execution

typically begins with the first line of `main()`. The `void` written in brackets indicates that the

`main` doesn't take any parameter (See this for more details). `main()` can be written to take parameters also. We will be covering that in future

posts.
The `int` was written before `main` indicates return type of `main()`. The value

returned by `main` indicates the status of program termination. See this post for more details

on the return type.
Line 3 and 6: `[{ and }]` In C language, a pair

of curly brackets define scope and are mainly used in functions and control statements like `if`,

`else`, `loops`. All functions must start and end with curly brackets.
Line 4 `[`

`printf("GeeksQuiz");]` `printf()` is a standard library function to print something on standard output. The semicolon at the end of `printf` indicates

line termination. In C, a semicolon is always used to indicate end of a statement.
Line 5 `[return 0;]` The return statement returns

the value from `main()`. The returned value may be used by an operating system to know the

termination status of your program. The value 0 typically means successful termination.

How to execute the above program:
In order to execute the above

program, we need to have a compiler to compile and run our programs.
can be used to start C without installing a
compiler.<p><i>Windows:</i>

There are many compilers available freely for the compilation of C
programs like Code Blocks

and Dev-CPP. We strongly recommend
Code Blocks.
<i>Linux:</i> For Linux, gcc comes
bundled with

Linux, Code Blocks can also be used with
Linux.
 </p>

</div>

<div class="nextprev">

< Home

Next >

</div>

</main>

</div>

<!-- JQuery cdn link -->

<script src="https://cdnjs.cloudflare.com/ajax/libs/jquery/3.6.0/jquery.min.js"></script>

</body>

</html>

<!--Algorithms-->

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<!-- Google font cdn link -->

<link rel="preconnect" href="https://fonts.googleapis.com">

<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>

```

<link href="https://fonts.googleapis.com/css2?family=Baloo+Bhai+2&display=swap"
rel="stylesheet">
<!-- font awesome cdn link -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-
awesome/5.15.3/css/all.min.css">
<!-- Custom CSS file link -->
<link rel="stylesheet" href="css/course.css">
<style>
li{
font-size: large;
}
</style>
<title>C++</title>
</head>
<body>
<div class="container-xxl my-md-4 bd-layout">
<aside class="bd-sidebar">
<nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
<ul class="list-unstyled mb-0 py-3 pt-md-1">
<li class="mb-1">
<button class="btn d-inline-flex align-items-center rounded collapsed" data-
bs-toggle="collapse"
data-bs-target="#getting-started-collapse" aria-expanded="false">
Algorithms
</button>
<div class="collapse" id="getting-started-collapse">
<ul class="list-unstyled fw-normal pb-1 small">
<li><a href="algorithm.html" target="_top" class="active"
class="d-inline-flex align-items-center rounded">Introduction</a></li>
<li><a href="#" target="_top"
class="d-inline-flex align-items-center rounded">Breadth First
Search</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">Depth

```

First Search (DFS)

Topological Sort

Johnson's algorithm

Binary Search

Quick Sort

Merge Sort

Selection Sort

Bubble Sort

Insertion Sort

 Heap Sort

</div>

</nav>

</aside>

<main class="bd-main order-1" id="intro">

<div class="bd-intro ps-lg-4">

<div class="d-md-flex flex-md-row-reverse align-items-center justify-content-between">

<h1 class="bd-title" id="content">"Introduction to algorithms"

<a class="anchorjs-link " aria-label="Anchor" data-anchorjs-icon="#" href="#content"

</h1>

</h1>

<div class="nextprev">

< Home

Next >

</div>

</div>

<p class="bd-lead">A programming algorithm is a procedure or formula used for solving a problem.

It is based on conducting a sequence of specified actions in which these actions describe how to do something,

and your computer will do it exactly that way every time.

An algorithm works by following a procedure, made up of inputs.

Once it has followed all the inputs, it will see a result, also known as output.

<p>Characteristics of an algorithm:

Precision – the steps are precisely stated.

 Uniqueness – results of each step are uniquely defined and only depend on the input and the result of the preceding steps.

 Finiteness – the algorithm stops after a finite number of instructions are executed.

Input – the algorithm receives input.

 Output – the algorithm produces output.

 Generality – the algorithm applies to a set of inputs.

</p>

<p>There are seven different types of programming algorithms:

 Sort algorithms

Search algorithms

Hashing

Dynamic Programming

Exponential by squaring

String matching and parsing

 Primality testing algorithms

</p>

<p>

The advantages of programming algorithms include:

 stepwise representation of a solution to a given problem, making it easy to understand.

Uses a definite procedure.

 Not dependent on a particular programming language.

 Every step in an algorithm has its own logical sequence, making it easy to debug.

 In Data Defined, we help make the complex world of data more accessible by explaining some of the most complex aspects of the field. </p>

</p>

</div>

</main>

</div>

</body>

</html>

<!--QUIZ-->

<!DOCTYPE html>

<head>

<title>Quiz</title>

<!-- Custom CSS file link -->

<link rel="stylesheet" href="css/course.css">

<script type="text/javascript" src="js/quiz.js"></script>

<style>

#button{

background-color: black;

color:white;

font-size: 10px;

```
padding: 7px;
}
</style>
</head>
<body>
<div class="container-xxl my-md-4 bd-layout">
<aside class="bd-sidebar">
<nav class="collapse bd-links" id="bd-docs-nav" aria-label="Docs navigation">
<ul class="list-unstyled mb-0 py-3 pt-md-1">
<li class="mb-1">
<button class="btn d-inline-flex align-items-center rounded collapsed" data-bs-
toggle="collapse"
data-bs-target="#getting-started-collapse" aria-expanded="false">
Quiz
</button>
<div class="collapse" id="getting-started-collapse">
<ul class="list-unstyled fw-normal pb-1 small">
<li><a href="quiz.html" target="_top" class="active"
class="d-inline-flex align-items-center rounded">C</a></li>
<li><a href="#" target="_top"
class="d-inline-flex align-items-center rounded">C++</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">C#</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">HTML</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">Javascript</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">CSS</a></li>
<li><a href="#" class="d-inline-flex align-items-center rounded">Algorithm</a></li>
</ul>
</div>
</li>
</ul>
</nav>
</aside>
<form name="quiz" id="quiz">
<center> <h1>C++ quiz</h1></center>
```

<div>

<p>

1. Which of the following is the correct syntax to add the header file in the C++ program?

</p>

<p>

<input type="radio" name="ques1" value="A">A.#include<userdefined>

</p>

<p>

<input type="radio" name="ques1" value="B">B.#include "userdefined.h"

</p>

<p>

<input type="radio" name="ques1" value="C">C.<include> "userdefined.h"

</p>

<p>

<input type="radio" name="ques1" value="D">D.Both A and B

</p>

</div>

<div>

<p>

2. Which of the following is the correct syntax to print the message in C++ language?

</p>

<p>

<input type="radio" name="ques2" value="A">A.cout <<"Hello world!";

</p>

<p>

<input type="radio" name="ques2" value="B">B.Cout << Hello world! ;

</p>

<p>

<input type="radio" name="ques2" value="C">C. Out <<"Hello world!;

</p>

<p>

<input type="radio" name="ques2" value="D">D.None of the above

</p>

</div>

<div>

<p>

3.Which of the following is the address operator?

</p>

<p>

<input type="radio" name="ques3" value="A">A. @

</p>

<p>

<input type="radio" name="ques3" value="B">B. #

</p>

<p>

<input type="radio" name="ques3" value="C">C. &

</p>

<p>

<input type="radio" name="ques3" value="D">D. %

</p>

</div>

<div>

<p>

4. Which of the following is the original creator of the C++ language?

</p>

<p>

<input type="radio" name="ques4" value="A">A.Dennis Ritchie

</p>

<p>

<input type="radio" name="ques4" value="B">B.Ken Thompson

</p>

<p>

<input type="radio" name="ques4" value="C">C.Bjarne Stroustrup

</p>

<p>

<input type="radio" name="ques4" value="D">D.Brian Kernighan

</p>

</div>

```

<div>
<p>
5.C++ is a ____ type of language.
</p>
<p>
<input type="radio" name="ques5" value="A">A. High-level Language
</p>
<p>
<input type="radio" name="ques5" value="B">B. Low-level language
</p>
<p>
<input type="radio" name="ques5" value="C">C.Middle-level language
</p>
<p>
<input type="radio" name="ques5" value="D">D. None of the above
</p>
</div>
<input type="button" name="" value="done" id="button" onclick="check()">
</form>
<center><p id="result"></p></center>
</div>
</body>
</html>

```

```

function check(){
var c=0;
var q1=document.quiz.ques1.value;
var q2=document.quiz.ques2.value;
var q3=document.quiz.ques3.value;
var q4=document.quiz.ques4.value;
var q5=document.quiz.ques5.value;
var result=document.getElementById('result');
var quiz=document.getElementById("quiz");
if (q1=="D") {c++}

```

```

if (q2=="A") {c++}
if (q3=="C") {c++}
if (q4=="C") {c++}
if (q5=="C") {c++}
quiz.style.display="none";
if(c<=3){
result.textContent='Your result is '+c+'.It is not so good please try to work on yourself.'
}
else
{
result.textContent='Your result is '+c+'.It is awesome,keep it up!'
}
}

/*common stylesheet for programming course,algorithms,quiz*/
:root{
--gradient:linear-gradient(90deg, #5d9fcc, #f1c40f);
}

*, ::after, ::before {
box-sizing: border-box;
}

*{
font-family: 'Baloo Bhai 2', cursive;
margin: 0; padding: 0;
box-sizing: border-box;
border: none; outline: none;
text-decoration: none;
text-decoration: capitalize;
transition: all .2s linear;
}

*::selection{
background: #5d9fcc;
color: #fff;

```

```
}  
html{  
font-size: 62.5%;  
overflow-x: hidden;  
border-top-right-radius: 10px;  
border-bottom-right-radius: 10px;  
scroll-behavior: smooth;  
}  
html::-webkit-scrollbar{  
width: 1.5rem;  
}  
html::-webkit-scrollbar-track{  
background: #333;  
}  
  
html::-webkit-scrollbar-thumb{  
background: linear-gradient(#5d9fcc, #f1c40f);  
border-radius: 10px;  
}  
.bd-sidebar {  
grid-area: sidebar;  
position: sticky;  
width: 29rem;  
height: 102vh;  
margin: -15px auto auto -2px;  
top: 1.5px;  
background-color: rgb(212 255 255);  
border-top-right-radius: 10px;  
border-bottom-right-radius: 10px;  
overflow: hidden;  
white-space: nowrap;  
box-shadow: 0px 0px 20px #d4d4d4;  
}  
::-webkit-scrollbar{
```

```
width: 1.5rem;
}
::-webkit-scrollbar-track{
background: #333;
}
::-webkit-scrollbar-thumb{
background: linear-gradient(#5d9fcc, #f1c40f);
border-radius: 10px;
}
aside {
display: block;
}
aside #mnu1{
font-size: 3rem;
color: crimson;
cursor: pointer;
padding-left: 10px;
padding-top: 6px;
display: none;
}
body {
margin: 0;
font-family: var(--bs-font-sans-serif);
font-size: 1rem;
font-weight: 400;
line-height: 1.5;
color: #212529;
background-color: #fff;
-webkit-text-size-adjust: 100%;
-webkit-tap-highlight-color: transparent;
}
.bd-links .btn {
padding: 1.5rem 2.5rem;
margin: 0px;
```

```
font-size: 25px;
font-weight: 600;
color: rgba(0,0,0,0.65);
background-color: transparent;
border: 0;
}

.bd-links a {
padding: .1875rem .5rem;
margin-top: .125rem;
border: 2px solid #064552;
border-top-right-radius: 12px;
border-bottom-left-radius: 12px;
font-size: 20px;
margin-left: 1.25rem;
color: rgba(0,0,0,0.65) !important;
text-decoration: none;
transition: .4s;
}

.bd-links a:hover{
padding-left: 45px;
background-color: #0dcaf0;
color: #ffffff;
}

.d-inline-flex {
display: inline-flex !important;
padding: 5px 11px;
margin: 4px 6px;
}

.bd-links a.active {
background-color: #0dcaf0;
color: #ffffff;
padding: 5px 11px;
margin: 4px 6px;
}
```

```
.bd-navbar .navbar-nav .nav-link {
padding-right: .25rem;
padding-left: .25rem;
color: rgba(255,255,255,0.85);
}
.navbar-dark .navbar-nav .nav-link {
color: rgba(255,255,255,.55);
}
.collapse ul a{
display: block;
}
.navbar-expand-md .navbar-nav .nav-link {
padding-right: .5rem;
padding-left: .5rem;
}
.navbar-nav .nav-link {
padding-right: 0;
padding-left: 0;
}
.nav-link {
display: block;
padding: .5rem 1rem;
color: #0d6efd;
text-decoration: none;
transition: color .15s ease-in-out,background-color .15s ease-in-out,
border-color .15s ease-in-out;
}
a {
color: #0d6efd;
text-decoration: underline;
}
a:-webkit-any-link {
color: -webkit-link;
cursor: pointer;
```

```
}  
li {  
text-align: -webkit-match-parent;  
}  
.navbar-nav {  
display: flex;  
flex-direction: column;  
padding-left: 0;  
margin-bottom: 0;  
list-style: none;  
}  
ul {  
list-style-type: disc;  
}  
:root {  
--bs-blue: #0d6efd;  
--bs-indigo: #6610f2;  
--bs-purple: #6f42c1;  
--bs-pink: #d63384;  
--bs-red: #dc3545;  
--bs-orange: #fd7e14;  
--bs-yellow: #ffc107;  
--bs-green: #198754;  
--bs-teal: #20c997;  
--bs-cyan: #0dcaf0;  
--bs-white: #fff;  
--bs-gray: #6c757d;  
--bs-gray-dark: #343a40;  
--bs-primary: #0d6efd;  
--bs-secondary: #6c757d;  
--bs-success: #198754;  
--bs-info: #0dcaf0;  
--bs-warning: #ffc107;  
--bs-danger: #dc3545;
```



```
--bs-light: #f8f9fa;
--bs-dark: #212529;
--bs-font-sans-serif:system-ui,-apple-system,"SegoeUI",Roboto,"HelveticaNeue",Arial,"Noto
Sans","Liberation Sans",sans-serif,"Apple Color Emoji","Segoe UI Emoji","Segoe UI
Symbol","NotoColorEmoji";--
bsfontmonospace:SFMonoRegular,Menlo,Monaco,Consolas,"Liberation Mono","Courier
New",monospace;--bs-gradient: linear-gradient(180deg, rgba(255, 255, 255, 0.15), rgba(255,
255, 255, 0));
}
.bd-intro {
grid-area: intro;
}
hr {
margin: 20px -16px;
}
.bd-title {
justify-content: space-between!important;
}
.bd-title {
font-size: calc(1.425rem + 2.5vw);
}
.h1, h1 {
font-size: calc(1.375rem + 1.5vw);
}
h1 {
display: block;
font-size: 40px;
margin-block-start: 0.67em;
margin-block-end: 0.67em;
margin-inline-start: 0px;
margin-inline-end: 0px;
font-weight: bold;
}
h2 {
```

```
font-size: 30px;
text-decoration: underline;
}
.h1, .h2, .h3, .h4, .h5, .h6, h1, h2, h3, h4, h5, h6 {
margin-top: 0;
margin-bottom: .5rem;
padding-top: 10px;
font-weight: 500;
line-height: 1.2;
}
p {
margin-top: 2rem;
margin-bottom: 1em;
font-size: 20px;
}
.bd-toc {
grid-area: toc;
}
.border-bottom {
border-bottom: 1px solid #dee2e6!important;
}
.d-block {
display: block!important;
}
.h6, h6 {
font-size: 1.8rem;
}
.bd-toc nav {
font-size: .875rem;
}
.codespan{
font-size: 105%;
background-color: #fff;
color: crimson;
```

```
padding: 4px;
}
.example {
padding: 8px 20px;
margin: 24px -20px;
box-shadow: none!important;
}
.example {
background-color: #E7E9EB;
border-radius: 5px;
border: 2px solid #0dcaf0;
padding: 0.01em 26px;
margin: 20px 0;
box-shadow: 0 2px 4px 0 rgba(0,0,0,0.16),0 2px 10px 0 rgba(0,0,0,0.12)!important;
}
h3 {
font-size: 27px;
margin-top: 3rem;
}
h4 {
font-size: 20px;
}
h4 {
display: block;
margin-block-start: 0.5em!important;
margin-block-end: 1.33em;
margin-inline-start: 0px;
margin-inline-end: 0px;
font-weight: bold;
border-bottom: 4px solid #569fa5;
}
.op123 {
margin: 0;
padding: 0;
```

```
border: 0;
font-size: 23px;
vertical-align: baseline;
}
pre {
background-color: #4e4e4e;
border-radius: 10px;
color: #e0e0e0;
font-family: Consolas,monospace;
margin-bottom: 10px;
padding: 15px;
font-size: 11pt;
overflow: auto;
white-space: pre;
}
.bd-lead {
font-size: calc(1.275rem + .3vw);
font-weight: 300;
font-size: 20px;
}
.bd-leadl {
font-size: calc(1.275rem + .3vw);
font-weight: 300;
font-size: 20px;
margin-left: 45px;
}
.container, .container-fluid, .container-lg, .container-md, .container-sm, .container-xl,
.container-xxl {
width: 100%;
padding-right: var(--bs-gutter-x,.75rem);
padding-left: var(--bs-gutter-x,.75rem);
margin-right: auto;
margin-left: auto;
}
```

```
.bd-content {
grid-area: content;
min-width: 1px;
}

.bd-content>h2:not(:first-child) {
margin-top: 3rem;
}

.co code {
color: crimson;
}

.col {
font-size: 23px;
color: crimson;
}

.w3-code, .w3-codespan {
font-family: Consolas,"courier new";
font-size: 16px;
}

.w3-code {
width: auto;
padding: 8px 12px;
border: 2px solid #0dcaf0;
word-wrap: break-word;
border-radius: 5px;
background-color: rgb(40,44,52);
color: white;
}

.w3-section, .w3-code {
margin-top: 16px!important;
margin-bottom: 16px!important;
}

.w3-code, w3-codeline {
font-size: 16px;
}
```

```
.code {
border-left: 4px solid #5fd5de;
margin-top: 16px!important;
margin-bottom: 16px!important;
font-size: 16px;
width: auto;
padding: 8px 12px;
word-wrap: break-word;
}

.nextprev a.right, .nextprev a.left {
background-color: #41c2dcc2;
color: rgb(255, 24, 70);
border-color: #0dcaf0;
border: 2px solid #064552;
border-top-right-radius: 12px;
border-bottom-left-radius: 12px;
}

.wntd, .wntd:link, .wntd:visited {
color: #FFFFFF;
background-color: #4CAF50;
}

.nextprev a {
font-size: 17.5px;
font-family: 'Source Sans Pro', sans-serif;
padding: 5px 20px;
}

.left {
float: left!important;
}

.right {
float: right!important;
}

.wntd, .button {
border: none;
```

```
display: inline-block;
padding: 8px 16px;
vertical-align: middle;
overflow: hidden;
text-decoration: none;
color: inherit;
background-color: inherit;
text-align: center;
cursor: pointer;
white-space: nowrap;
}

.table {
width: 100%;
max-width: 100%;
margin-bottom: 20px;
}

.table-bordered {
border: 1px solid #ddd;
}

table {
background-color: transparent;
border-collapse: collapse;
border-spacing: 0;
}

table {
display: table;
border-collapse: separate;
box-sizing: border-box;
text-indent: initial;
border-spacing: 2px;
border-color: grey;
}

tbody {
display: table-row-group;
```

```

vertical-align: middle;
border-color: inherit;
}
table {
border-collapse: separate;
text-indent: initial;
border-spacing: 2px;
}
tr {
display: table-row;
vertical-align: inherit;
border-color: inherit;
}
.table>thead>tr>th, .table>tbody>tr>th, .table>tfoot>tr>th, .table>thead>tr>td,
.table>tbody>tr>td, .table>tfoot>tr>td {
padding: 8px;
line-height: 1.42857143;
/* vertical-align: top; */
font-size: 19.5px;
border-top: 1px solid #ddd;
}
.table-bordered>thead>tr>th, .table-bordered>tbody>tr>th, .table-bordered>tfoot>tr>th,
.table-bordered>thead>tr>td, .table-bordered>tbody>tr>td, .table-bordered>tfoot>tr>td {
border: 1px solid #ddd;
}
th {
background: #eee;
}
th {
display: table-cell;
vertical-align: inherit;
font-weight: bold;
text-align: -internal-center;
}

```



```
@media (min-width: 768px){
.bd-layout {
display: grid;
gap: 1.5rem;
grid-template-areas: "sidebar main";
grid-template-columns: 1fr 3fr;
}
}

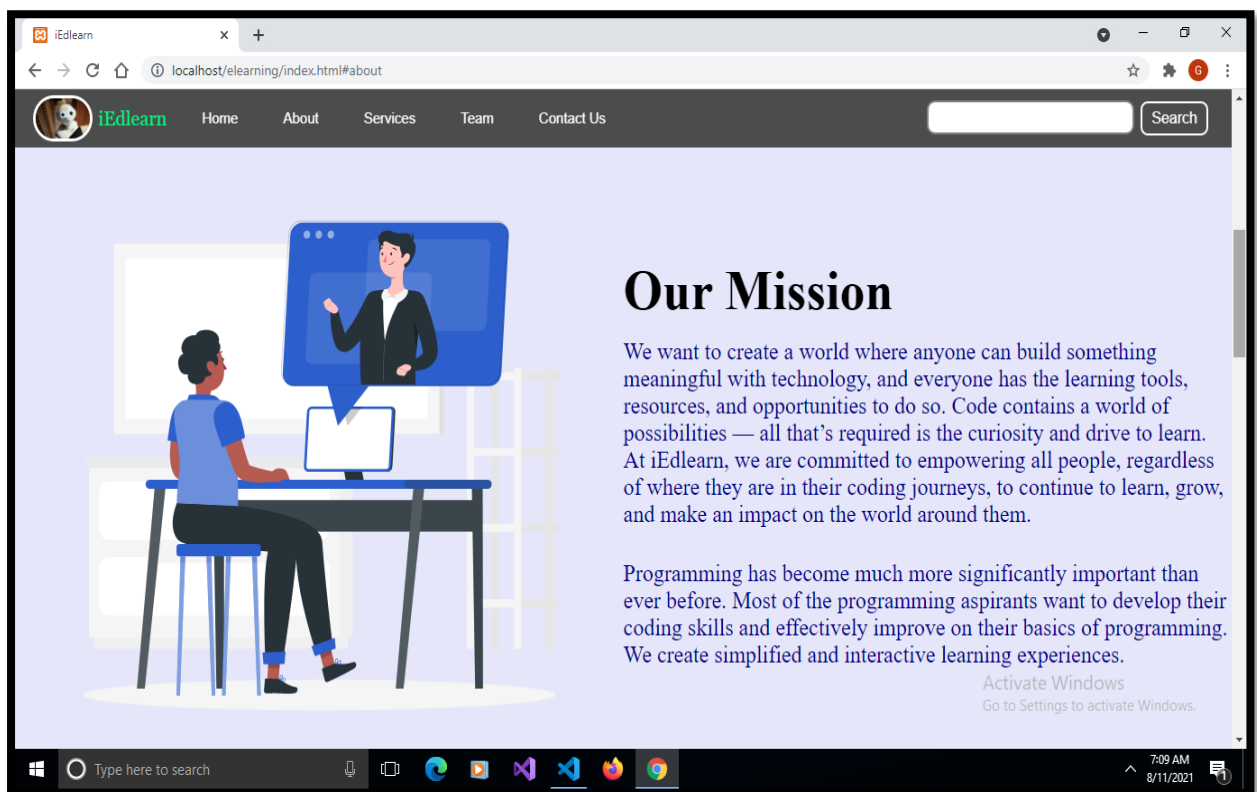
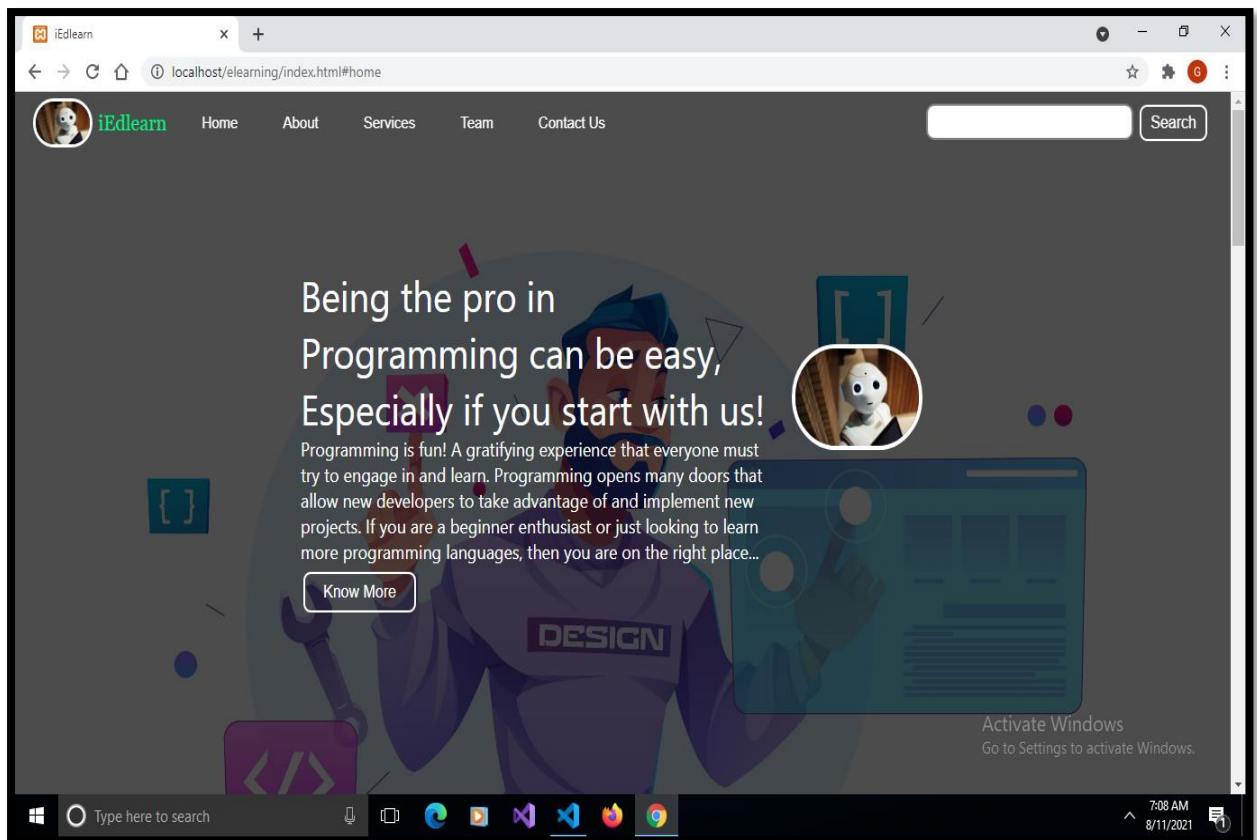
@media (min-width: 768px){
.my-md-4 {
margin-top: 1.5rem!important;
margin-bottom: 1.5rem!important;
grid-template-columns: 0.309fr 1fr;
}
}

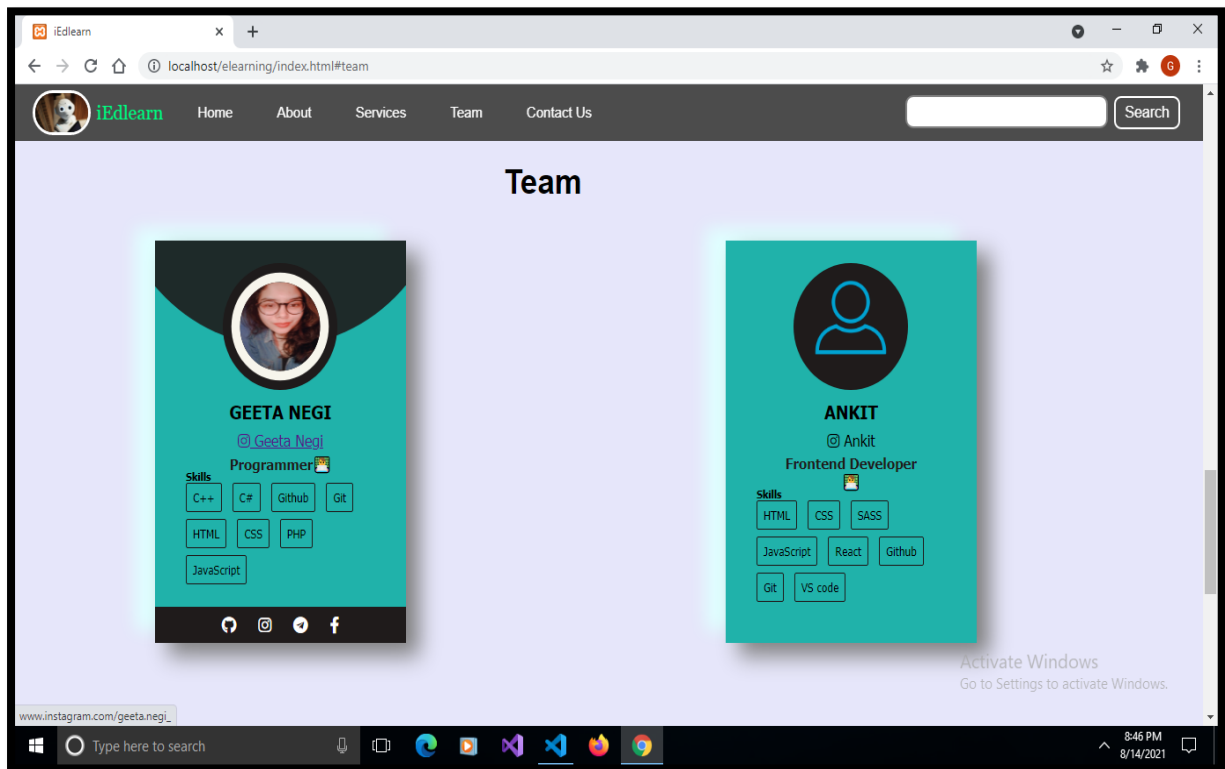
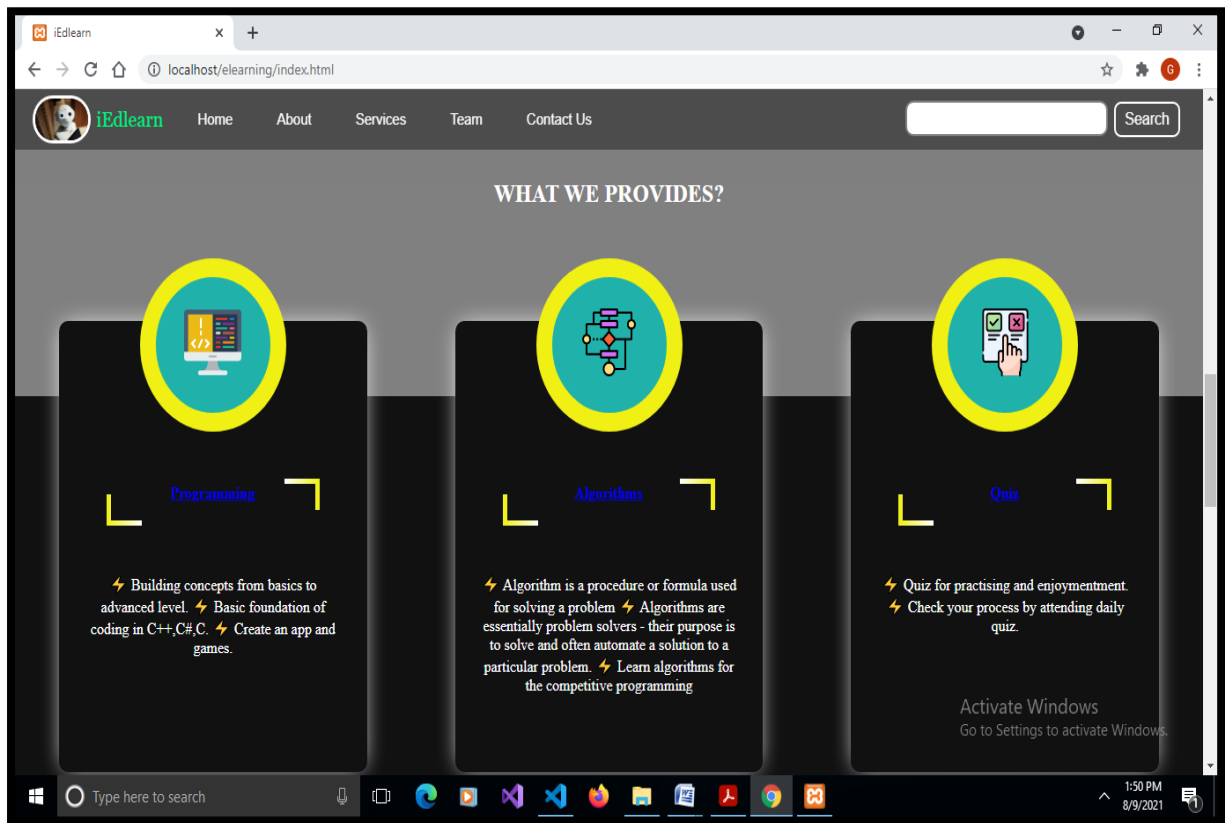
@media (min-width: 768px){
.bd-main {
display: grid;
gap: inherit;
grid-template-areas:
"intro"
"toc"
"content";
grid-template-rows: auto auto 1fr;
}
}

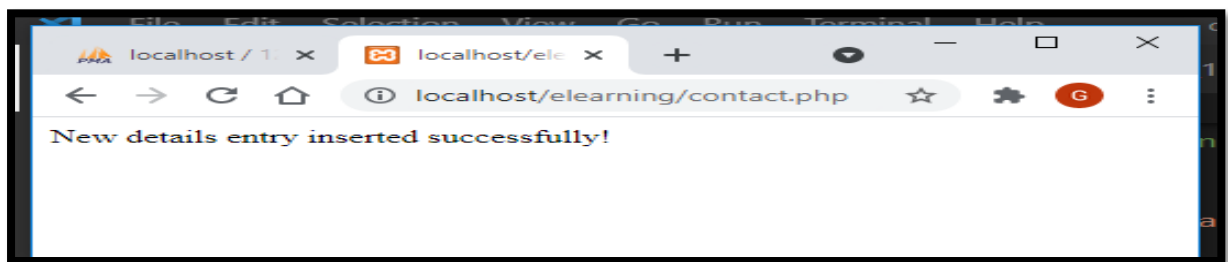
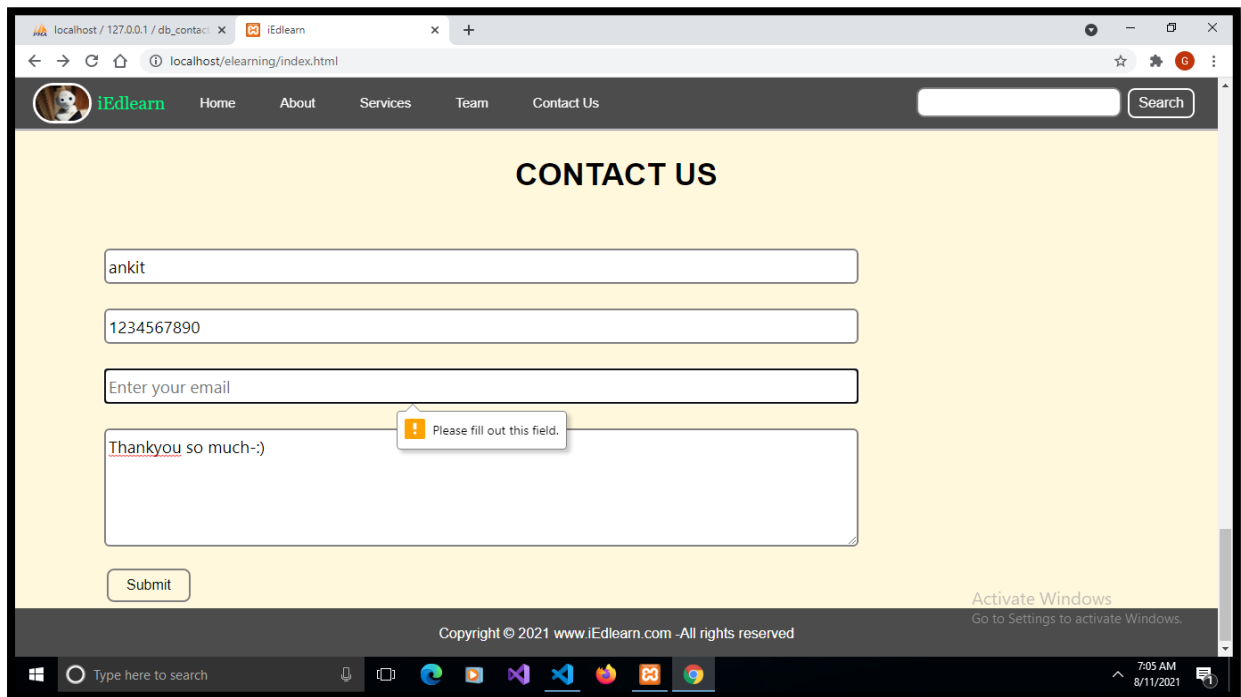
@media (min-width: 768px){
.bd-links {
position: -webkit-sticky;
position: sticky;
top: 0rem;
display: block !important;
height: calc(113vh - 7rem);
padding-left: .25rem;
```

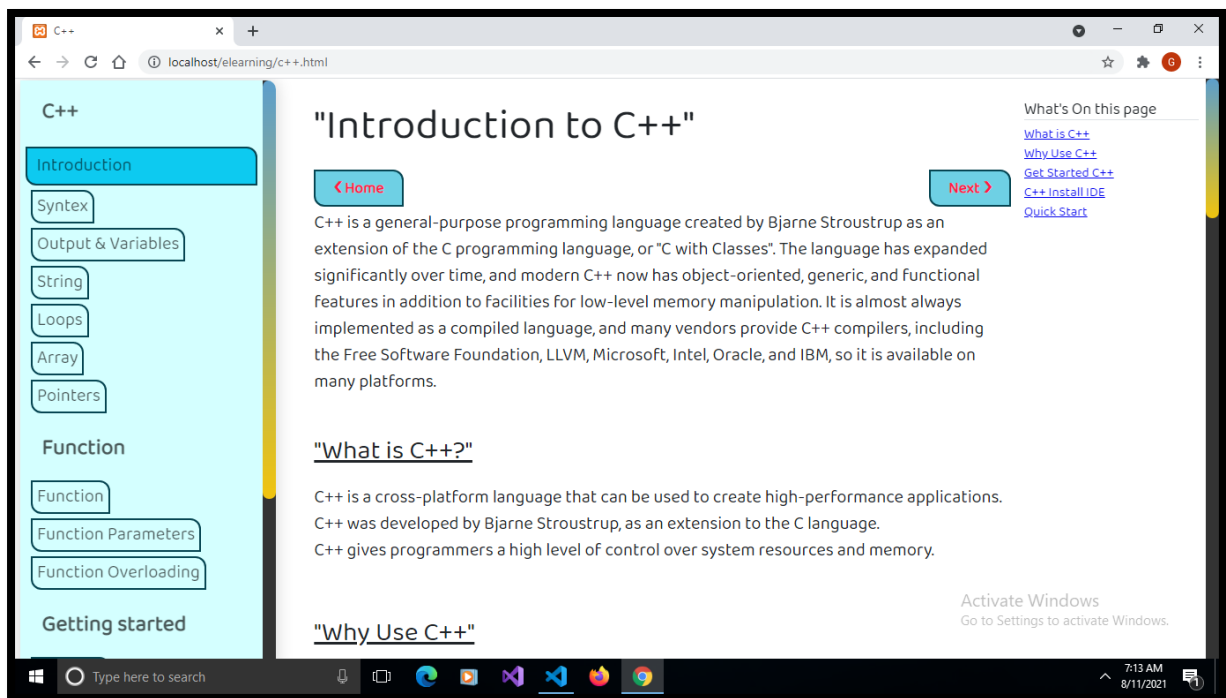
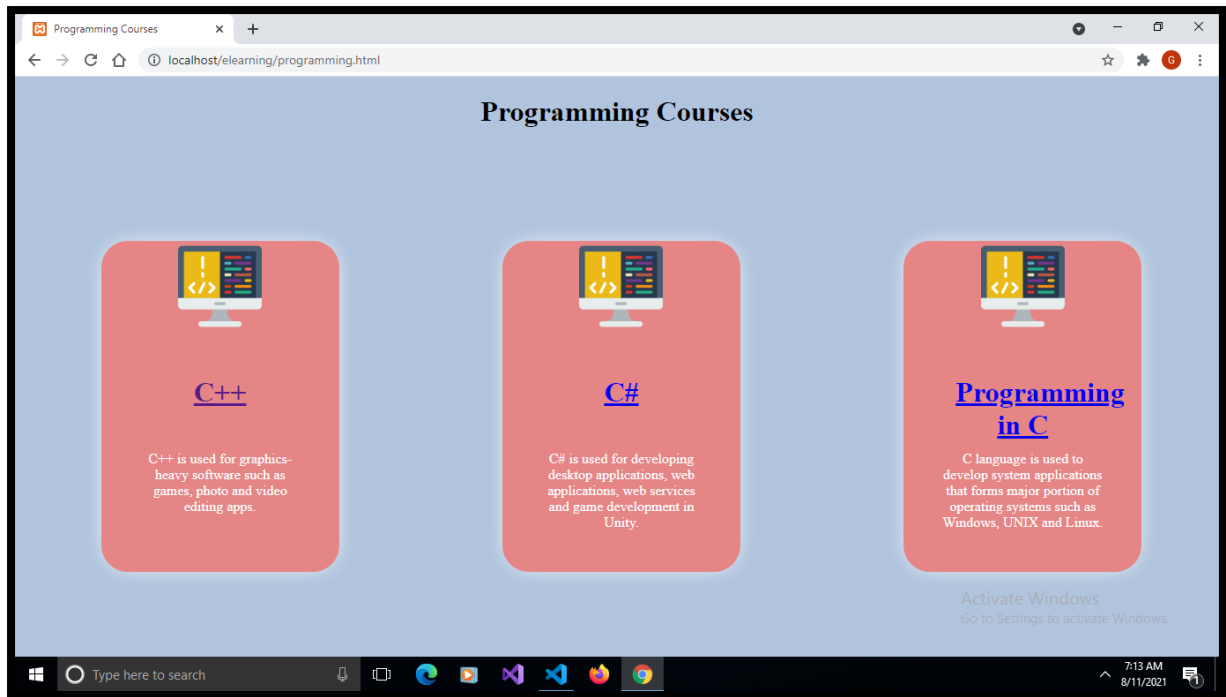
```
margin-left: -.25rem;
overflow-y: auto;
}
}
@media (min-width: 992px){
.bd-toc {
position: -webkit-sticky!important;
position: sticky;
top: 0rem;
right: 0;
z-index: 2;
height: calc(100vh - 7rem);
overflow-y: auto;
}
.ps-lg-4 {
padding-left: 0.5rem!important;
}
}
@media (min-width: 992px){
.bd-main {
grid-template-areas:
"intro toc"
"content toc";
grid-template-columns: 4fr 1fr;
grid-template-rows: auto 1fr;
}
}
```

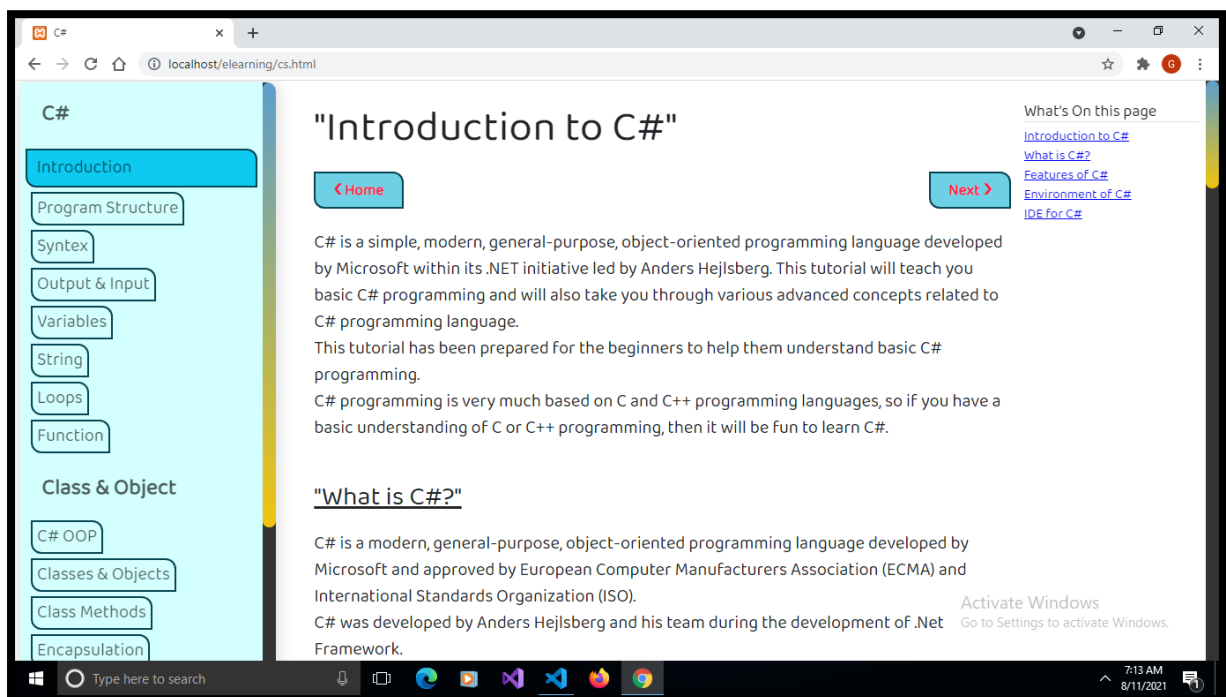
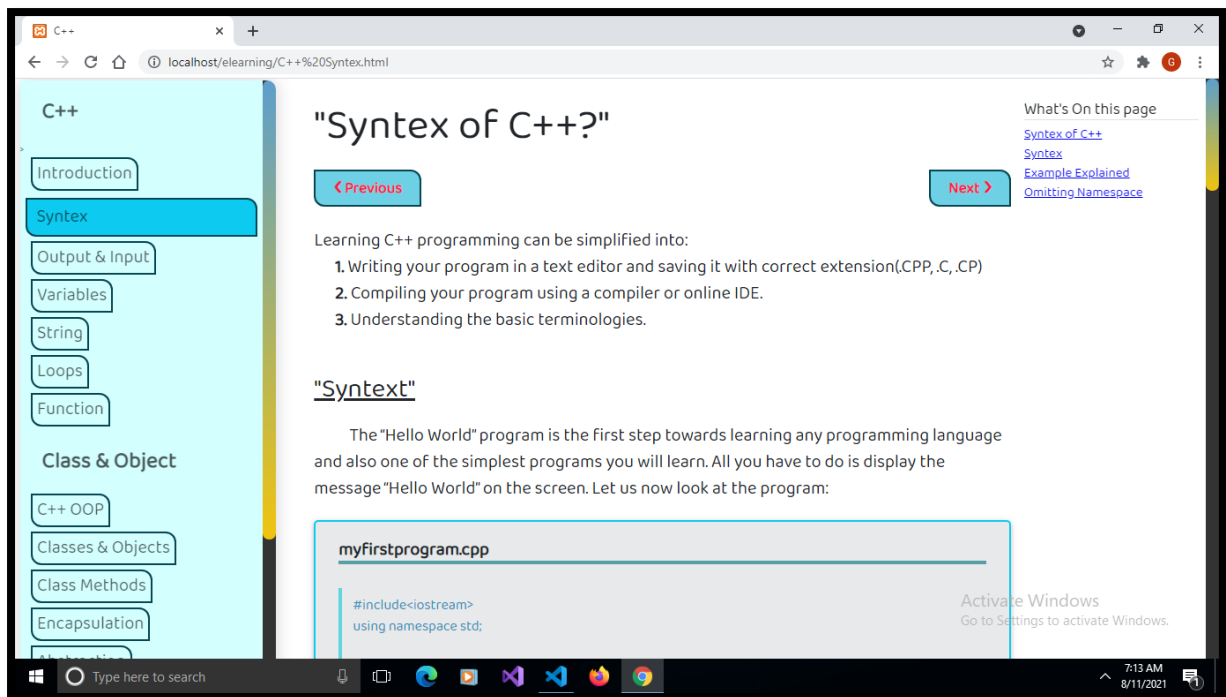
Input/Output From Screenshots

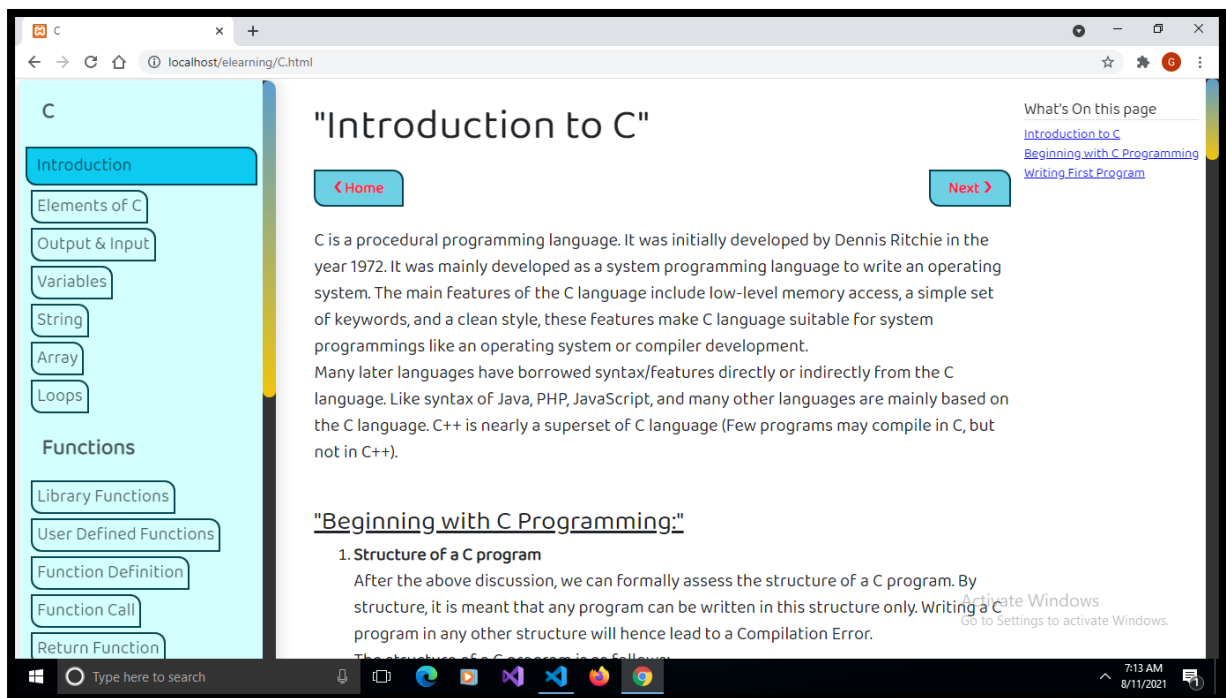
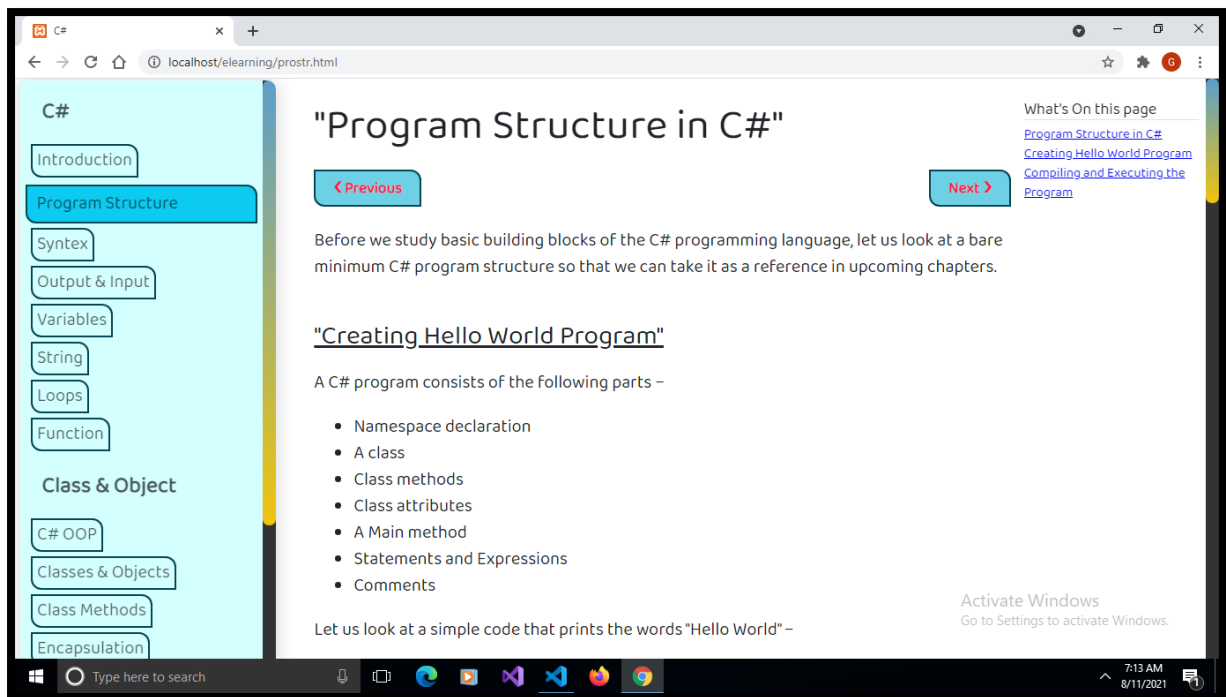












Algorithms

- Introduction
- Breadth First Search
- Depth First Search (DFS)
- Topological Sort
- Johnson's algorithm
- Binary Search
- Quick Sort
- Merge Sort
- Selection Sort
- Bubble Sort
- Insertion Sort
- Heap Sort

"Introduction to algorithms"

[< Home](#) [Next >](#)

A programming algorithm is a procedure or formula used for solving a problem. It is based on conducting a sequence of specified actions in which these actions describe how to do something, and your computer will do it exactly that way every time. An algorithm works by following a procedure, made up of inputs. Once it has followed all the inputs, it will see a result, also known as output.

Characteristics of an algorithm:

- Precision – the steps are precisely stated.
- Uniqueness – results of each step are uniquely defined and only depend on the input and the result of the preceding steps.
- Finiteness – the algorithm stops after a finite number of instructions are executed.
- Input – the algorithm receives input.
- Output – the algorithm produces output.
- Generality – the algorithm applies to a set of inputs.

There are seven different types of programming algorithms:

- Sort algorithms

Activate Windows
Go to Settings to activate Windows.

C++ quiz

Quiz

- C
- C++
- C#
- HTML
- Javascript
- CSS
- Algorithm

1. Which of the following is the correct syntax to add the header file in the C++ program?

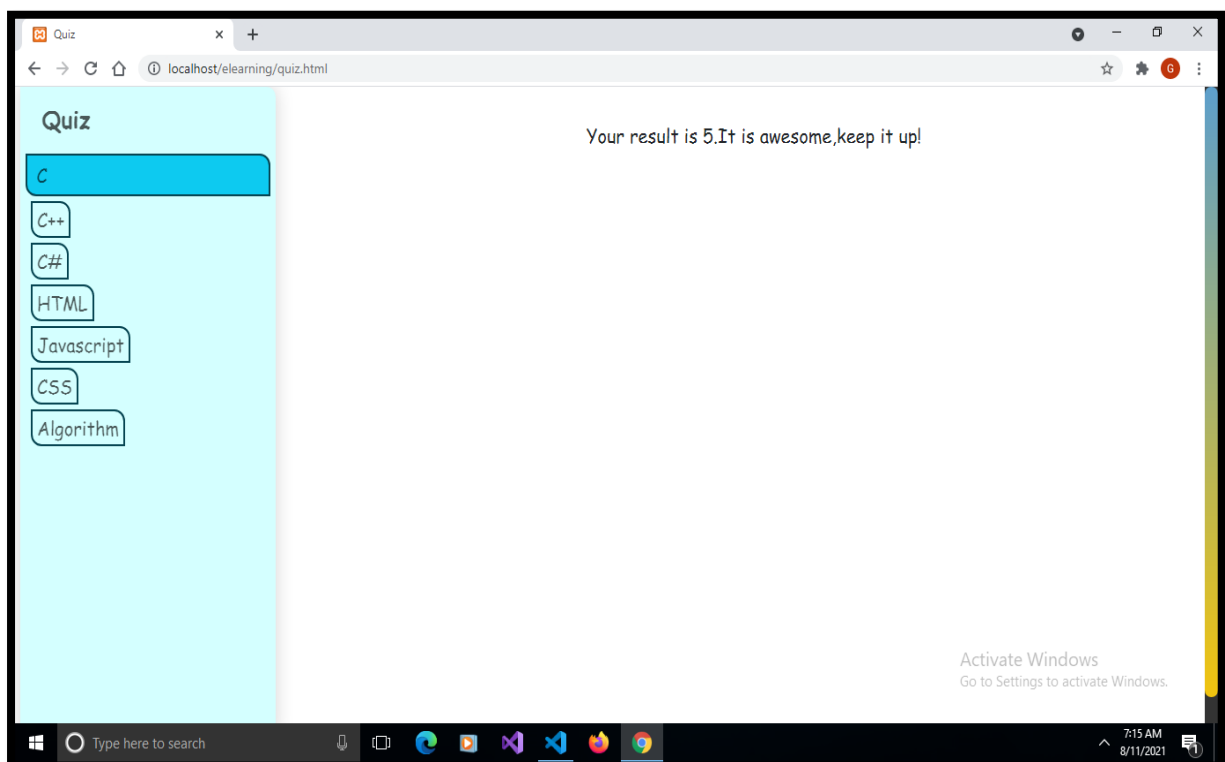
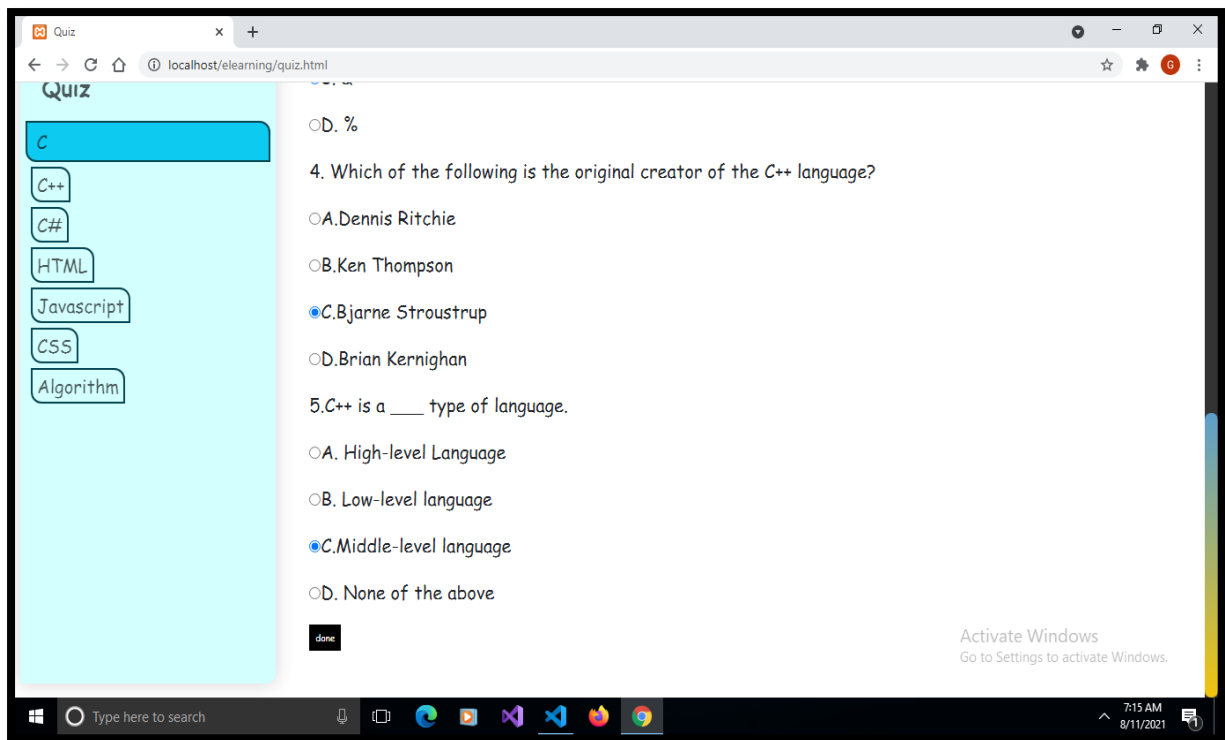
- ☐ A. #include
- ☐ B. #include "userdefined.h"
- ☐ C. "userdefined.h"
- ☒ D. Both A and B

2. Which of the following is the correct syntax to print the message in C++ language?

- ☒ A. cout << "Hello world!";
- ☐ B. Cout << Hello world! ;
- ☐ C. Out << "Hello world!;
- ☐ D. None of the above

3. Which of the following is the address operator?

Activate Windows
Go to Settings to activate Windows.



localhost / 127.0.0.1 / db_contact x iEdlearn x +

localhost/phpmyadmin/index.php?route=/sql&server=1&db=db_contact&table=details&pos=0

phpMyAdmin

Recent Favorites

- New
- db_contact
 - New
 - details
- ellearn
- ellearning
- information_schema
- logandregistration
- login
- mysql
- performance_schema
- phpmyadmin
- test

Server: 127.0.0.1 » Database: db_contact » Table: details

Browse Structure SQL Search Insert Export Import Privileges Operations Triggers

Current selection does not contain a unique column. Grid edit, checkbox, Edit, Copy and Delete features are not available.

Showing rows 0 - 2 (3 total, Query took 0.0006 seconds.) [phone: 9998887770... - 1234567890...]

SELECT * FROM `details` ORDER BY `phone` DESC

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 | Filter rows: Search this table

+ Options

fname	phone	email	umessage
Geeta	9998887770	geetanegi550@gmail.com	Thankyou for providing such a good content.
Geeta Negi	8670986723	geeta123@email.com	
ankit	1234567890	ankit34@gmail.com	Thankyou so much:-)

Show all | Number of rows: 25 | Filter rows: Search this table

Query results operations

Print Copy to clipboard Export Display chart Create view

Activate Windows
Go to Settings to activate Windows.

Console

Type here to search

7:08 AM
8/11/2021

TESTING

Methodology used for testing

The completion of a system will be achieved only after it has been thoroughly tested. Though this gives a feel the project is completed, there cannot be any project without going through this stage. Hence in this stage it is decided whether the project can undergo the real time environment execution without any break downs, therefore a package can be rejected even at this stage.

Testing methods:

Software testing methods are traditionally divided into black box testing and white box testing. These two approaches are used to describe the point of view that a test engineer takes when designing test cases.

1. Black box testing - Black box testing treats the software as a "black box," without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning, boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.
2. White box testing - White box testing, by contrast to black box testing, is when the tester has access to the internal data structures and algorithms (and the code that implement these). White box testing methods can also be used to evaluate the completeness of a test suite that was created with black box testing methods. This allows the software team to examine parts of a system that are rarely tested and ensures that the most important function points have been tested.
3. Grey Box Testing - Grey box testing involves having access to internal data structures and algorithms for purposes of designing the test cases, but testing at the user, or black-box level. Manipulating input data and formatting output do not qualify as "grey box," because the input and output are clearly outside of the "black-box" that we are calling the system under test. This distinction is particularly important when conducting integration testing between two modules of code written by two different developers, where only the interfaces are exposed. Grey box testing may also include reverse engineering to determine, for instance, boundary values or error messages.
4. Acceptance testing - Acceptance testing can mean one of two things:

- A smoke test is used as an acceptance test prior to introducing a build to the main testing process.
 - Acceptance testing performed by the customer is known as user acceptance testing (UAT).
5. Regression Testing - Regression testing is any type of software testing that seeks to uncover software regressions. Such regression occurs whenever software functionality that was previously working correctly stops working as intended. Typically, regressions occur as an unintended consequence of program changes. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged.
6. Non-Functional Software Testing - Special methods exist to test non-functional aspects of software.
- Performance testing checks to see if the software can handle large quantities of data or users. This is generally referred to as software scalability. This activity of Non-Functional Software Testing is often times referred to as Load Testing.
 - Stability testing checks to see if the software can continuously function well in or above an acceptable period. This activity of Non-Functional Software Testing is often times referred to as indurations test.
 - Usability testing is needed to check if the user interface is easy to use and understand.
 - Security testing is essential for software which processes confidential data and to prevent system intrusion by hackers.
 - Internationalization and localization are needed to test these aspects of software, for which a pseudo localization method can be used.

Maintenance

MAINTENANCE

Once the website is delivered and developed, it enters the maintenance phase. All websites need maintenance. Websites need to be maintained because there are often some residual errors or bugs remaining in the websites that must be removed as they are discovered. Many of these surfaces only after the websites has been in operation sometimes for a long time. Websites need routine maintenance to ensure that all of the components (links, inquiry or contact forms, images,) are working in a safe and efficient manner.

Some of the steps we are going to follow to maintain our websites are pointed below:

Weekly task -

- Updating the content regularly.
- Check that all the web pages are loading without errors
- Search for 404 errors and fix or redirect
- Check the pages to see if there are any broken links

Monthly task-

- Checking the loading speed of the website and ensure that nothing is bogging it down.
- Reviewing the security scans and make sure nothing is out of place
- Analyze website statistics from the previous month

Conclusion

Conclusion

Our project is only a humble venture to satisfy the needs to manage their project work. Several user friendly coding have also adopted. This package shall prove to be a powerful package in satisfying all the requirements of the programmers. The objective of software planning is to provide a frame work that enables the manger to make reasonable estimates made within a limited time frame at the beginning of the software project and should be updated regularly as the project progresses.

At the end it is concluded that we have made effort on following points...

- A description of the background and context of the project and its relation to work already done in the area.
- Made statement of the aims and objectives of the project.
- The description of Purpose, Scope, and applicability.
- We define the problem on which we are working in the project.
- We describe the requirement Specifications of the system and the actions that can be done on these things
- We included features and operations in detail, including screen layouts.
- We designed user interface and security issues related to system
- Finally the system is implemented and tested according to test cases

Future Scope

Future scope of the project:

In a nutshell, it can be summarized that the future scope of the project circles around maintaining information regarding:

- We can give more advance features for **E-learning platform for programmers** including more facilities.
- We will make a registration and login system for the user.
- We will make an admin portal so that admin can easily uploads and manipulate data.
- We will host the platform on online servers to make it accessible worldwide
- Integrate multiple load balancers to distribute the loads of the system
- Implement the backup mechanism for taking backup of codebase and database on regular basis on different servers.

The above mentioned points are the enhancements which can be done to increase the applicability and usage of this project. Here we can maintain the records of process of the user. Also, as it can be seen that now-a-days the players are versatile, i.e. so there is a scope for introducing a method to maintain the E-learning Management System. Enhancements can be done to maintain all the programming courses, quiz, user records.

We have left all the options open so that if there is any other future requirement in the system by the user for the enhancement of the system then it is possible to implement them.. In the last we would like to thanks all the persons involved in the development of the system directly or indirectly. We hope that the project will serve its purpose for which it is develop there by underlining success of process.

References

References

Bibliography

During the development of E-learning platform for programmers project, We have gone through the following books and websites.

Books:-

- HTML & CSS Design and Build Websites by Jon Duckett
- Php and mysql web development

Websites:-

- www.Google.com
- www.w3schools.com
- www.geekforgeeks.com

Images and Icons:-

- www.storyset.com
- www.freepik.com

