The operation of coolers, pumps and cooling towers is carried out in accordance with safety and in the specified technological sequence for effective operation under different cooling loads. The first two methods are related to changes in the operation of the installation compared to the original design, by relatively simple modification of the equipment (in some cases it is not required at all); the other five methods involve a more substantial modification of the equipment to achieve high performance.

To solve this problem, we propose the following algorithm: two coolers operate individually and stop / start is controlled by the controller. Usually one chiller allows you to get the job done. The second cooler is added in the case of high cooling capacity or when the process flow is higher than the maximum evaporator flow of a single cooler. A second cooler is also added if one chiller is unable to do the job because one of the circuits is not running at rated power. The algorithm checks the outlet temperature and measures the cooling time and adds an extra chiller if necessary.

Igor Samusenko
Research and language supervisor: K. Kugai, senior teacher
Kyiv National University of Technologies and Design

WEB DEVELOPMENT AS A BRANCH OF INFORMATION TECHNOLOGY

Information technology is becoming more and more popular year by year. Today a lot of people want to find their place in this industry. And one of the most popular branches of information technology for today is Web development. Web development is the work involved in developing a web site for the Internet or an intranet . Web development gains momentum and becomes increasingly popular. Today there are a lot of people who try themselves in this industry.

Website development includes some main steps:

1. Website design
2. Front-end
3. Back-end

Website design – the first step of web development. It is a type of design whose task is to design web user interfaces for sites or web applications. A web designer must have a lot of knowledge about making a design for web industry. He has to know a lot about a colour scheme, using of fonts, design of pictures. Also he needs to know how to work in graphics editors (Adobe Photoshop, Sketch, Adobe Illustrator).

The second step in web development is front-end. It deals with the client side of user interface for the software and hardware of the service. A front-end developer way of studying is very consistent. First of all he must learn HTML, which is a foundation of website page. Then he must learn the CSS, which helps a front-end developer to make a style of a web page. The page with CSS looks more colourful and interesting. Next stage is mastering JavaScript. JavaScript – a multi-paradigm programming language. It supports object-oriented, imperative and functional styles. This is an ECMAScript implementation. It is the most widely used in browsers as a scripting language to make interactivity to web pages. JavaScript is a programming language, so you need more time to learn it well. Also there are a lot of frameworks, which make web developing with JavaScript faster.

Back-end – the final step of web development. Back-end is an invisible part of website. It is a server side: downloading information from the site server, sending messages, searching for information.

The back-end consists of three parts:

- server

- applications
- database

The back-end is responsible for:

- storing and organizing data;
- interacting with the interface;
- sending and receiving information displayed as a web page.

Back-end is the most difficult part of the web development for learning. A back-end developer must have basic knowledge of HTML. He must have advanced knowledge of sever programming languages such as Php, Java, Python. Also he needs to know SQL – a database query language. Back-end also has frameworks and the most popular is Yii.

So, Web development is very popular IT sphere nowadays, which grows very fast. A lot of people want to become a web developer, but this is not very simple, so you have to be patient, you need to practice more, follow the news of this industry and you will achieve success!

A.V. Serbeniuk
Language supervisor: I. A. Koliieva
Dnipro National University of Railway Transport named after Academician V. Lazarian.

DEBUGGING

Debugging is the routine process of locating and removing computer program bugs, errors or abnormalities, which is methodically handled by software developers via debugging tools. Software developers spend 35-50 percent of their time validating and debugging software. While tools, languages, and environments have reduced the time spent on individual debugging tasks, they have not significantly reduced the total time spent debugging, nor the cost of doing so.

Why is it important? Even the best program developers sometimes miss something essential, so bugs will always be present during the development process, and, most of the times, even when the product is released. The main reason debugging exists for is to find and eliminate these program errors before the user finds them first. If a user finds a malfunction in a program he paid money for, he will obviously be quite frustrated. This can negatively influence the developer`s reputation, and maybe even discourage the user from using the product at all, so it is best to prevent any mistakes in the program behavior before the release.

The ability to quickly and effectively find and resolve bugs in new and established systems is one of the most valuable engineering skills that you can develop. Since this skill enables the rapid development and maintenance of high-quality engineered systems, it is foundational for many technology companies, and is one of their most valued and sought-after skills. Nevertheless, this skill is not considered to be worth paying a lot of attention to during the education period in most cases, rarely evaluated in coding interviews, and often poorly understood and documented.

How are the bugs formed? The most sinister bugs occur when programmers falsely believe their mental models to be complete. This is the crux of the problem: they have assumed correctness in their implementations and by definition do not know where they went wrong. The only way programmers can hope to solve such bugs is through knowledge acquisition. It does seem clear that debugging is the more difficult task. Since solving bugs requires learning, the debugging process can be made easier by better understanding effective learning and teaching strategies.

How to debug properly? The most optimal procedure for finding and fixing bugs is essentially the same across all domains and haven`t changed a whole lot since it was introduced, since there is little need in further development of this process.