1. Difference engine accepts binary code (0 and 1), and gears were used to set up a binary code.

 True

 False

**QUESTION 2**

1. Because ENIAC accepted only two numbers, 0 and 1, it did not take a long time to program in ENIAC.

 True

 False

**QUESTION 3**

1. Because a binary code uses only numbers 0 and 1, there is no way to represent text using a binary code.

 True

 False

**QUESTION 4**

1. Assembly language needs an assembler to convert its code to binary code.

 True

 False

**QUESTION 5**

1. Fortran is one of the old programming languages that is still used in a bank system.

 True

 False

**QUESTION 6**

1. Scheme is the first functional programming language.

 True

 False

**QUESTION 7**

1. Basic has a simple syntax because it was targeted at beginners.

 True

 False

**QUESTION 8**

1. Pascal is a language that combines the best features of Fortran, COBOL, and Python.

 True

 False

**QUESTION 9**

1. Java became popular because an interactive TV had huge success.

 True

 False

**QUESTION 10**

1. Python was developed as a hobby during spring break.

 True

 False

Solved Question 1

The answer is "False".

Difference Engine, an early calculating machine, verging on being the first computer, designed and partially built during the 1820s and ’30s by Charles Babbage. The Difference Engine was a digital device: it operated on discrete digits and the digits were decimal (0–9), represented by positions on toothed wheels, rather than the binary digits (“bits”) .

Solved Question 2

The answer is "False".

ENIAC (Electronic Numerical Integrator and Computer) was the first programmable, electronic, general-purpose digital computer.It was able to solve "a large class of numerical problems" through reprogramming. It was not like binary programming (0,1) but it was a complex programming.

It was never wasy to code with ENIAC. Some program would run for several hours, days and even years.

Solved Question 3

The answer is "False".

ASCII abbreviated from American Standard Code for Information Interchange, is a character encoding standard for electronic communication. ASCII codes represent text in computers, telecommunications equipment, and other devices.

All the alphabets have their ASCII codes. For example, A has ASCII code 065 which is represented as 01000001 in binary. When you use this given binary form, it represents 'A'. In this way text can be represented.

Solved Question 4

The answer is "True".

An assembler is a program that converts assembly language into machine code. It takes the basic commands and operations from assembly code and converts them into binary code that can be recognized by a specific type of processor.

Solved Question 5

The answer is "False".

The programming language is COBOL. From many federal government agencies to your local bank, COBOL is still in use. An estimated 43% of banking systems and 95% of ATM swipes utilize COBOL code.

Solved Question 6

The answer is "False".

LISP was the first functional programming language, was developed in the late 1950s for the IBM 700/7000 series of scientific computers by John McCarthy while at Massachusetts Institute of Technology (MIT).

Scheme was created during the 1970s.

Solved Question 7

The answer is "True".

As its name suggests, it was designed for beginners. BASIC (Beginners' All-purpose Symbolic Instruction Code) is a family of general-purpose, high-level programming languages whose design philosophy emphasizes ease of use. BASIC was a beginner code to learn when the code was first released that later on could be used to serve advanced purposes.

Solved Question 8

The answer is "False".

Pascal was the youngest member of the Algol family of languages. Algol, defined in 1960, was the first high-level language with a readable, structured, and systematically defined syntax.

Pascal, which was named after the mathematician Blasé Pascal, is a direct descendent from ALGOL 60. Pascal also draws programming components from ALGOL 68 and ALGOL-W.

Solved Question 9

The answer is "False".

Originally Java was created for writing software to be embedded in various consumer electronic devices like microwave ovens, remote controls, set-top box for TVs, etc. Although, it didn’t get much love from the consumer electronic space. It gain popularity in building Web Applications.

Solved Question 10

The answer is "True".

In December 1989, Python’s creator Guido Van Rossum was looking for a hobby project to keep him occupied in the week around Christmas. He had been thinking of writing a new scripting language that’d be a descendant of ABC and also appeal to Unix/C hackers. He chose to call it Python.

Please vote for me if it helped you.

**QUESTION 11**

1. Before C was developed, assembly language was usually used to write operating systems because of its fast performance. But now, in most cases, C is used to write operating systems. Explain why C is popularly used in the operating system instead of assembly language.

The first justification why C is approved as a great programming language over assembly language for the running systems is due to its dominant language which consists of a collection of built-in functions and its operators that benefit in writing the complex program. C programs are extremely important ever since they comprise of data types of varieties and operators which include robustness.

The effectiveness of an assembly language and sophisticated language are integrated in the C compiler, which methods it as the most suitable language for devising any operating system and system software.

C language is one of the simplest languages to learn as it has a fixed number of keywords that are 32 bits long with the fixed set of control primitives like if, for, while switch and do while. It also includes a set of standard C functions which are mostly used while writing simply as well as complex programs.

.

The C programming language has some advantages over assembly language which are as follows:

C syntax is very much simpler for learning in comparison with Assembly language syntax.

C is considerably easier to use for developing extra multifaceted programs.

C has a fast performance in comparison with assembly language.

C learning is additional beneficial than understanding assembly language because there is additional developing part around C language than assembly language.

One of the main reasons why C has always been a popular language because of its portability. With minimum changes, one can run a C program on any other operating system. Another important thing is that the C language supports structured as well as procedural programming syntax, which forms it as most applicable for structured programming

**QUESTION 12**

1. If a language supports an implicit variable declaration, does it mean that the language is dynamically typed? If not, prove that an implicit variable declaration does not mean that a language is dynamically typed.

Implicit typing can be able to get you out of a mess and make programming a bit smoother in numerous circumstances. For example, implicit declarations mean of associating variables with types through default conventions.

A **language is dynamically typed** if the type **is** linked with run-time values, and not named variables/fields/etc. This **indicates** that you as a programmer **be able to** write a little more rapidly for the reason that you **do** not have to specify types of every time or without utilizing a **statically** typed **language** with type interpretation.

Dynamic type binding:

- no declaration. A variable is bound to a type when it is assigned

a value in an assignment statement (at run time, take the type

of the value on the right-hand side)

- dramatic difference between static and dynamic type binding

- main advantage of dynamic type binding: flexibility.

A dynamic language (**Lisp**, **Perl**, **Python**, **Ruby**) is designed to optimize programmer efficiency, so you can implement functionality with less code.

1. Explicit declarations: a statement in a program that lists variable names and specifies their types

2.

Any dynamically typed if the type of a variable is checked during run-time. Common examples of dynamically typed languages involve JavaScript, Objective-C, PHP, Python, Ruby, Lisp, and Tcl

Python is a dynamically typed language. Java is a statically typed language. In a weakly typed language, variables can be implicitly coerced to unrelated types, whereas in a strongly typed language. Example both Java and Python are strongly typed languages

they cannot,

+++

and an explicit conversion is required. ...

**QUESTION 13**

1. Assume that the below statement is written in C. What is a value that variable ‘**testVar**’ has? (Assume that it uses C with the version that supports Boolean values.)

**int testVar = 15 + ‘b’ + true;**

### QUESTION 14

1. Describe briefly the difference between the pre-test loop and the post-test loop.

### QUESTION 15

1. What is the difference between Goomba and Paragoomba?

### QUESTION 16

1. Python does not support the switch(case) statement. Convert the below switch statement in Java to if statement in Python.
2. // Assume that curHero contains string value
3. **switch(curHero) {**
4. **case “Mario”:**
5. **System.out.println(“I am Mario!”);**
6. **break;**
7. **case “Yoshi”:**
8. **System.out.println(“I am Yoshi!”);**
9. **break;**
10. **case “Goomba”:**
11. **System.out.println(“I am Goomba!”);**
12. **break;**
13. **default:**
14. **System.out.println(“Who are you?”);**
15. **}**

**QUESTION 17**

1. Python does not support the post-test loop. Convert the below post-test loop in Pascal to the pre-test loop in Python.

|  |
| --- |
| **result := 1;**  **i := 2;**  **repeat**  **result := result + i;**  **i = i\*i;**  **until result >= 105** |

**QUESTION 18**

1. In the below C# program, write classes ‘Mario’ and ‘Yoshi’ to have the output result described in the output window. Write codes of both classes (Mario & Yoshi) in the answer section.

|  |
| --- |
| **class Hero**  **{**  **public virtual void PrintString()**  **{**  **Console.WriteLine(“Here is the list of heroes.”);**  **}**  **}**  **//Need to write Mario class**    **//Need to write Yoshi class**    **class Program**  **{**  **static void Main(string[] args)**  **{**  **Hero hero = new Hero();**  **hero.PrintString();**  **Console.WriteLine(“First Hero:”);**  **hero = new Mario();**  **hero.PrintString();**  **Console.WriteLine(“Second Hero:”);**  **hero = new Yoshi();**  **hero.PrintString();**  **}**  **}** |

**/C# code**

class Hero{  
   public virtual void PrintString(){  
       Console.WriteLine("Here is the list of heroes.");  
   }  
}  
class Mario:Hero{  
   public override void PrintString(){  
       Console.WriteLine("I am Mario!");  
   }  
}  
class Yoshi:Hero{  
   public override void PrintString(){  
       Console.WriteLine("I am Yoshi!");  
   }  
}  
class Program{  
   static void Main(string[] args){  
       Hero hero = new Hero();  
       hero.PrintString();  
       Console.WriteLine("First Hero:");  
       hero = new Mario();  
       hero.PrintString();  
       Console.WriteLine("Second Hero:");  
       hero = new Yoshi();  
       hero.PrintString();  
    }  
}

**SOURCE CODE:**

**\*Please follow the comments to better understand the code.**

**\*\*Please look at the Screenshot below and use this code to copy-paste.**

**\*\*\*The code in the below screenshot is neatly indented for better understanding.**

**Please give me an upvote dear, Much Appreciated.!!**

using System;

class Hero

{

public virtual void PrintString()

{

Console.WriteLine("Here is the list of heroes.");

}

}

//Need to write Mario class

class Mario : Hero // inherit from the Hero class

{

// override the PrintString() method from the parent

public override void PrintString()

{

Console.WriteLine("I am Mario!");

}

}

//Need to write Yoshi class

class Yoshi : Hero // inherit from the Hero class

{

// override the PrintString() method from the parent

public override void PrintString()

{

Console.WriteLine("I am Yoshi!");

}

}

class Program

{

static void Main(string[] args)

{

Hero hero = new Hero();

hero.PrintString();

Console.WriteLine("First Hero:");

hero = new Mario();

hero.PrintString();

Console.WriteLine("Second Hero:");

hero = new Yoshi();

hero.PrintString();

}

}

class Imposter

{

public virtual void PrintString()

{

Console.WriteLine("Here is the list of imposters.");

}

}

**//Need to write Kait class**

**class Kait: Imposter**

**{**

**public override void PrintString()**

**{**

**Console.WriteLine("I am Kait!");**

**}**

**}**

**//Need to write John class**

**class John: Imposter**

**{**

**public override void PrintString()**

**{**

**Console.WriteLine("I am John!");**

**}**

**}**

class Program

{

static void Main(string[] args)

{

Imposter imposter = new Imposter();

imposter.PrintString();

Console.WriteLine("First imposter:");

imposter = new Kait();

imposter.PrintString();

Console.WriteLine("Second imposter:");

imposter = new John();

imposter.PrintString();

}

}

**QUESTION 19**

1. Match the closest relevant concept to the short description.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  |  | | --- | --- | |  | Once a variable type is fixed, it cannot be modified. | |  | Types are checked everywhere in a source code. | |  | Types are checked only if there are called during run time. | |  | Type’s suitability is determined by checking whether it has certain methods. | |  | “If it walks like a duck and it quacks like a duck, then it must be a duck.” | |  | It produces type-error messages. | |  | It does not produce type-error messages. | |  | Types are checked at compile time. | |  | Variable type is determined based on a type of value, and it can be changed freely. | |  | Type errors are resolved via implicit type conversion. | | |  |  | | --- | --- | | a. | Dynamic Type Checking | | b. | Weak Typing | | c. | Strong Typing | | d. | Static Type Checking | | e. | Duck Typing | |

Match the closest relevant concept to the short description. a.

Once a variable type is fixed, it cannot be modified. c.

Types are checked everywhere in a source code. d.

Types are checked only if there are called during run time. e.

Type’s suitability is determined by checking whether it has certain methods. b.

“If it walks like a duck and it quacks like a duck, then it must be a duck. a.

It produces type-error messages. c.

It does not produce type-error messages. d.

Types are checked at compile time. e.

Variable type is determined based on a type of value, and it can be changed freely. b.

Type errors are resolved via implicit type conversion. a.

Dynamic Type Checking b.

Weak Typing c.

Strong Typing d.

Static Type Checking e.

Duck Typing

10 points

**Paragoomba** is a **Goomba** along with wings and is a part or representative **of** the **Koopa** Troop who works **for** Bowser. Their only **difference** visually from a standard

**Goomba** is their wings which they utilize to take off or jump alongside. They have made many developments in several Fantastic Mario Series games over the years, and are one **of** Mario's for the most part common foes

Describe briefly the difference between the pre-test loop and the post-test loop.

A pretest loop tests its situation prior for each iteration. A posttest loop tests its situation following for each iteration. A posttest loop will continually implement at minimum once. For the Reason that they are only implemented when a situation is true

A pre-test loop is one and only in which the loop situation is checked prior to entering the loop. There are numerous widespread modifications of the pre-test loop. The greatest ordinary of these is the While Loop. The merely discrepancy is that while an If Declaration performs once, a While Declaration can implement numerous times

The pretest loop verifies the situation and, if true, performs the declarations in its body. Such as i=0; while (i<3) {Console. WriteLine(i); i++;} This will output 1 1 1 but next code: i=4; while (i<3) {Console. WriteLine(i); i++;} has no outputs.

Because do while **loops** safety check the situation **when** the block is implemented, the command arrangement is often also known as a **post**-**test loop**.

This implies that the code be required to constantly be implemented first and then the communication or **test** situation is **evaluated**. If it is true, the code implements the body of the **loop** again.

|  |  |
| --- | --- |
| PreTest Do/Loop | PostTest Do/Loop |
| 1. Do-while examinations the loop situations each time all the way through the loop & it remains implementing while the experiment expression is a true value. | Loop while we examine the situation at the bottom. |
| 2.When the qualified articulation is false then | When the loop while is utilized, the body of loop |

|  |  |
| --- | --- |
| the declarations in Do/Loop are skipped. | is implemented at least once upon a time since the situation is at the bottom of the loop. |
| 3. General Syntax is :  Do while <condition>  <statements>  Loop General Syntax is | General Syntax is :  Do  <statements>  Loop while<condition> |
| 4. Ex.  Do While intNumber < 100  lblNumber.Caption=intNumber  intNumber = intNumber + 1  Loop | Ex. Dim intNum as integer  Do  Form1.print intNum  intNum = intNum +1  Loop while(intnum <= 10) |