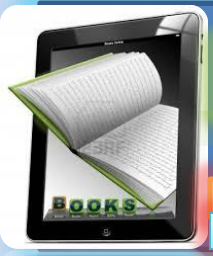

RANCHO SOLUTION

Programming Language 1 Assignment

Make sure you read your Books, Remember RANCHO will not solve problems in exams class.

Click on the Exercise to jump to destination Exercise

You can RUN those program in your system.

QUESTIONS

Exercise 1

Answer No: (1) (2) (3) (4)

Exercise 2

Answer No: (8) (9) (10) (11) (12)

Exercise 3

Answer No: (1) (2) (3) (4)

Exercise 4

Answer No: (1)

Exercise 7

Answer No: (1) (8) (10)

Exercise 8

Answer No: (1) (3) (4)

Exercise 12

Answer No: (1) (2) (3)

Exercise 1.

(1) Using an illustration explain the four features of an algorithm.

Illustration:

1. **Deterministic** - e.g 4 + 5 = 9 **OR** 11 + 11 = 22. Always the same no matter what.
2. **Finite** - e.g Solving the problem of graduating from school **OR** Solving the problem of traveling from one place to another. It must be completed in a finite amount of time.
3. **Scheduled** – e.g The process of cooking food **OR** the process of making cake. This will run over a series of steps it is scheduled to the next step.
4. **Effective** – e.g The problem of getting to Onitsha from Enugu.

(2) In producing an algorithm, there are three main phases implementation, development and maintenance. List the stages in the three phases.

Answers

Reference Page 2

(3) Assuming you are the adviser to the governor on youth matters. Produce an algorithm to solve youth unemployment in Anambra State using the steps outlined above.

Answers

Development Phase: (a) Define the problem – Youth unemployment means an existing youth without a paid job.

(b) Logical sequence steps to solve – 1. Government building many manufacturing companies. 2. Organizing seminary for job creation. 3. Giving out loan. 4. Advertising for job opportunities, Seminary and Loan giving out.

Implementation Phase: Using Quick Basic

```
CLS

PRINT "Building many manufacturing companies";

PRINT "Go join the seminary for job creation";

INPUT "How many youth attended the seminary :", NUM;

PRINT NUM;

END
```

(4) There are four main forms of expressing an algorithm. List them
   (a) Pseudo code
   (b) Flowchart
   (c) Programming language
   (d) Control table

Reference page 3.

Exercise 2.

(8) List five properties of a programming language

(a) Reliability

(b) Robustness

(c) Usability

(d) Portability

(e) Maintainability

(f) Efficiency/performance

(g) Readability.

Reference Page 10.

(9) List three types of translators that there are.

(a) Assemblers

(b) Compilers

(c) Interpreters

(10) There are five generations of programming languages what are their characteristics

(a) First generation: 1. They are directly executable by the machine. 2. No translator is use to compile or assemble the first generation programming language.

(b) Second generation: 1. Their code can be read and written by a programmer.

2. The language is specific to a particular processor family and environment.

(c) Third generation: 1. They are high level language. 2. They are portable across many different architectures.

(d) Fourth generation: They are mostly non procedural meaning. They didn't need the programmer to specify every step for the program.

Reference page 13 and 14.

(11) List the five programming paradigm you know

(a) Imperative paradigm.

(b) Declarative programming paradigm

(c) Functions programming paradigm

(d) Object oriented programming

Reference Page 15

(12) There are three types of programming errors list them

(a) Syntax error

(b) Logical error

(c) Runtime error

Reference Page 20.

Exercise 3.

(1) What is the difference between debugging and software testing?

Answer

Debugging is a methodical process of finding and reducing bug in a computer program. This is for removing error.

Software testing is the investigation carried out on a piece of software or program to provide stakeholders with information about the software. This is for knowing the effectiveness the a software.

Reference Page 21.

(2) List three types of software testing
- (a) White-box testing
- (b) Black-box testing
- (c) Grey-box testing

Reference page 22.

(3) List the debugging techniques you know
- (a) Print debugging
- (b) Remote debugging
- (c) Post mortem debugging

Reference page 21.

Exercise 4.

(4) Discuss the following – objects, fields, classes and methods

Answer

Reference Page 36.

Exercise 7.

(1) Write a program in C++ which will calculate the simple interest given that the principal is 6,000 naira the rate is 20 percent and the time is 5 years. The principal, time and rate must be declared first.

```
// Topic : Simple Interest.cpp
// Cousre : Programmeing language 1
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	int principal = 6000, time = 5, rate = 20;
	cout << "The principal : " << principal << endl;
	cout << "Enter the rate : " << rate << endl;
	cout << "Enter the time : " << time << endl;
	cout << "The Sinple interest : " << (principal * rate * time)/100 << endl;
	return 0;
}
```

(8) Write a program in C++ that calculates the circumference of a circle where the radius is 9cm and Pi is given as 3.142.

(1) Using the radius method

```
/ circumference of a circle in C++.cpp
// Using the method of Radius.
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	float const PI = 3.142;
	float radius = 20;
	cout << "The radius : " << radius << endl;
	cout << "The Circumference of a Circle : " << 2 * PI * radius << endl;
	return 0;
}
```

(10) List 6 data types you know in C++

Reference Page 69 and 70.

Exercise 8.

(1) Write a program that outputs the result of 18 modulus 7

```
// No 5 The working of modules operators.cpp
//
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	int divide, remainder;
	divide = 223 / 12;
	cout << "Two number dividing 223 divid by 12 : " << divide << endl;
	remainder = 223 % 12;
	cout << "The remainder is : " << remainder << endl;
	return 0;
}
```

(3) Write a program in C++ that adds 16 to 24 then divides the result by 4

```
// Adding and substraing C++.cpp
// Cousre : Programming language 1
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	int Num1 = 16, Num2 = 24;
	float divide, add;
	add = Num1 + Num2;
	cout << "Adding num1 and num2 : " << add << endl;
	divide = add / 4;
	cout << "Dividing by 4 : " << divide << endl;
	return 0;
}
```

(4) Write a program in C++ that compares the value of 5 and 55 and outputs the one that is greater.

```
// No 6 using of IF statment and relational oper.cpp
#include "stdafx.h"
#include <iostream>
using namespace std;
int main()
{
	int n1 = 5, n2 = 55;
	cout << "Enter first number : " << n1 << endl;
	cout << "Enter second number : " << n2 << endl;
	if (n1 == n2)
		cout << "Number are equal" << endl;
	if (n1 > n2)
		cout << "First is greater second number" << endl;
	if (n1 < n2)
		cout << "First number is less than second number";
	return 0;
}
```

Exercise 12.

(1) Declare a one dimensional array called friends which contains a list of characters of five of your closet friends

```
// Supported By : Sep Club.
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;
int main()
{
	string friends[5] = {"Young dude ", "Talented ", "Ghandi ", "Joyce ", "Onyinye "};
	int i = 0;
	cout << "The names of my closet friend is : ";
	for (i = 0; i < 5; i++)
	{
		cout << "\n";
		cout << friends[i] << endl;
	}
	cout << "\n";
	return 0;
}
```

(2) Write a C++ program and declare an array called pals which is a two dimensional array which contains a list of 7 of your closet friends and their course of study.

```
// Two dimension Array.cpp
// Topic : Friends Names and course
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;
int main()
{
	string pal[7][2] = {"Talented ", "Business Admin ", "Young dude ", "Computer ", "Onyinye ",
"Nusring" , "Joyce   ", "Accounting" , "Ghandi ", "Engeering" , "Franklin ", "Maths"};
	int a = 0, i = 0;
	cout << "My friends name and course" << endl;
	for (int i = 0; i < 7; i++)
	{
		for (int a = 0; a < 2; a++)
		{
			cout << "\t " << pal[i][a];
		}
		cout << "\n";
	}
	return 0;
}
```

(2) Write a program in C++ declaring the class car and its objects and methods.

```
// Creating classes in C++.cpp
#include "stdafx.h"
#include <iostream>
#include <string>
using namespace std;
class Car {
public :
      void MercedesCar();
      void ToyotaCar();
};
void Car::MercedesCar() {
      cout << "This is Mercedes method and object" << endl;
}
void Car::ToyotaCar() {
      cout << "This is Toyota method and object" << endl;
}
int main()
{
      Car c;
      c.MercedesCar();
      c.ToyotaCar();
      return 0;
}
```