**ASSIGNMENT 1**

**SUBJECT: OOPS**

**SUBMISSION DATE: 6th JULY 2022**

**Total marks: 25**

**Theory based questions:**

**Ques1** Relate about object- oriented programming and distinguish it with procedure-oriented programming?

Answer:-

|  |  |  |
| --- | --- | --- |
| **Parameter** | **Procedural Programming** | **Object Oriented Programming** |
| Definition | This programming language makes use of a step by step approach for breaking down a task into a collection of routines (or subroutines) and variables by following a sequence of instructions. It carries out each step systematically in order so that a computer easily gets to understand what to do. | This programming language uses objects and classes for creating models based on the real-world environment. This model makes it very easy for a user to modify as well as maintain the existing code while new objects get created by inheriting the characteristics of the present ones. |
| Security | Procedural Programming does not offer any method of hiding data. Thus, it is less secure when compared to Object Oriented Programming. | Hiding data is possible with Object Oriented Programming due to the abstraction. Thus, it is more secure than the Procedural Programming. |
| Method | The main program gets divided into minute parts on the basis of the functions. It then treats them as separate programs for smaller programs individually. | It involves the concept of classes and objects. Hence, it divides the program into minute chunks known as objects. These are actually instances of classes. |
| Division of Program | Procedural Programming divides the program into small programs and refers to them as functions. | Object Oriented Programming divides the program into small parts and refers to them as objects. |
| Movement of Data | Available data is capable of moving freely within the system from one function to another. | The objects are capable of moving and communicating with each other through the member functions. |
| Approach | The Procedural Programming follows a Top-Down approach. | The Object Oriented Programming follows a Bottom-Up approach. |
| Importance | This programming model does not give importance to data. It prioritizes the functions along with the sequence of actions that needs to follow. | This programming model gives importance to the data rather than functions or procedures. It is because it works on the basis of the real world. |
| Orientation | It is Structure/Procedure oriented. | It is Object Oriented. |
| Basis | The main focus in Procedural Programming is on how to do the task, meaning, on the structure or procedure of the program. | The main focus in Object Oriented Programming is on data security. Hence, it only permits objects to access the class entities. |
| Type of Division | It divides any large program into small units called functions. | It divides the entire program into small units called objects. |
| Inheritance | It does not provide any inheritance. | It achieves inheritance in three modes- protected, private, and public. |
| Virtual Classes | There is no concept of virtual classes. | The concept of virtual functions appears at the time of inheritance. |
| Overloading | The case of overloading isn’t possible in the case of Procedural Programming. | Overloading is possible in the form of operator overloading and function overloading in the case of Object Oriented Programming. |
| Reusability of Code | No feature of reusing codes is present in Procedural Programming. | Object Oriented Programming offers the feature to reuse any existing codes in it by utilizing a feature known as inheritance. |
| Most Important Attribute | It prioritizes function over data. | It prioritizes data over function. |
| Modes of Access | The Procedural Programming offers no specific accessing mode for accessing functions or attributes in a program. | The Object Oriented Programming offers three accessing modes- protected, private, and public. These, then, serve as a share to access functions of attributes. |
| Size of Problems | It is not very suitable for solving any big or complex problems. | It is suitable for solving any big or complex problems. |
| Addition of New Function and Data | It is not very easy to add new functions and data in the Procedural Programming. | It is very easy to add new functions and data in the Object Oriented Programming. |
| Access to Data | In the Procedural Programming, most of the functions use global data for sharing. They can access freely from one function to another in any given system. | In the Object Oriented Programming, the present data cannot easily move easily from one function to another. One can keep it private or even public. Thus, a user can control the data access. |
| Data Sharing | It shares the global data among the functions present in the program. | It shares data among the objects through its member functions. |
| Data Hiding | No proper way is available for hiding the data. Thus, the data remains insecure. | It can hide data in three modes- protected, private, and public. It increases the overall data security. |
| Basis of World | The Procedural Programming follows an unreal world. | The Object Oriented programming follows the real world. |
| Friend Classes or Friend Functions | It doesn’t involve any concept of friend function. | Any class or function is capable of becoming a friend of any other class that contains the keyword “friend.”  **Note** – The keyword “friend” only works for C++. |
| Examples | Some common examples of Procedural Programming are C, Fortran, VB, and Pascal. | The examples of Object Oriented Programming languages are Java, C++, VB.NET, Python, and C#.NET. |

(4)

**Ques 2** Point out the need of pre-processor directive #include<iostream>

Answer:- The **#include** directive allows a programmer to include contents of one file inside another file. This is commonly used to separate information needed by more than one part of a program into its own file so that it can be included again and again without having to re-type all the source code into each file.

C++ generally requires you to *declare* what will be used before using it. So, files called [**headers**](https://en.wikipedia.org/wiki/Header_(computing)) usually include declarations of what will be used in order for the compiler to successfully compile source code. This is further explained in the [File Organization Section](https://en.wikibooks.org/wiki/C%2B%2B_Programming/Programming_Languages/C%2B%2B/Code/File_Organization) of the book. The **standard library** (the repository of code that is available with every standards-compliant C++ compiler) and 3rd party libraries make use of headers in order to allow the inclusion of the needed declarations in your source code, allowing you to make use of features or resources that are not part of the language itself.

(2)

**Ques3** Describe the advantages of using new operator as compared to function malloc() (2)

Answer:- Advantages of new over malloc () :

1. new does not need the sizeof() operator where as malloc() needs to know the size before memory allocation.
2. Operator new can make a call to a constructor where as malloc() cannot.
3. new can be overloaded malloc() can never be overloaded.
4. new could initialise object while allocating memory to it where as malloc () cannot.
5. Malloc() function has a prototype:
6. ptr = (cast-type\*) malloc(byte-size)
7. sizeof(data type) is used to allocate the memory for datatype.
8. Once the program requirement is over, we have to explicitly free the malloc() allocated memory by using free() function.
9. In case of new, delete is used instead of free().

**Ques4** Distinguish between an inline function and a pre-processor macro? (2)

| **BASIS FOR COMPARISON** | **INLINE** | **MACRO** |
| --- | --- | --- |
| Basic | Inline functions are parsed by the compiler. | Macros are expanded by the preprocessor. |
| Syntax | inline return\_type funct\_name ( parameters ){ . . . } | #define macro\_name char\_sequence |
| Keywords Used | inline | #define |
| Defined | It can be defined inside or outside the class. | It is always defined at the start of the program. |
| Evaluation | It evaluates the argument only once. | It evaluates the argument each time it is used in the code. |
| Expansion | The compiler may not inline and expand all the functions. | Macros are always expanded. |
| Automation | The short functions, defined inside the class are automatically made onto inline functions. | Macros should be defined specifically. |
| Accessing | An inline member function can access the data members of the class. | Macros can never be the members of the class and can not access the data members of the class. |
| Termination | Definition of inline function terminates with the curly brackets at the end of the inline function. | Definition of macro terminates with the new line. |
| Debugging | Debugging is easy for an inline function as error checking is done during compilation. | Debugging becomes difficult for macros as error checking does not occur during compilation. |
| Binding | An inline function binds all the statements in the body of the function very well as the body of the function start and ends with the curly brackets. | A macro faces the binding problem if it has more than one statement, as it has no termination symbol. |

**Application based 5 Marks each (3 question)**

**Ques1** Define a class to represent a bank account includes the following members. (5)

Data members:

1. Name of depositor
2. Account number
3. Type of account
4. Balance amount in the account

Member function:

1. To assign initial value
2. To deposit an amount
3. To withdraw an amount after checking the balance
4. To display name and balance

Write a main program to test the program.

Code:- #include <iostream>

Using namespace std

Class account

{

Char name[50];

Int acnumber;

Char type[30];

float balance;

public:

void getdata(void);

void asignvalue(void);

void depositeammount(void);

void withdraw(void);

void display(void);

};

Int am, total=0, value;

Void bank :: getdata(void)

{

Cout<<”Enter your name”;

Cin>>name;

Cout<<”Enter your account number”;

Cin>>acnumber;

Cout<<”Enter account type”;

Cin>>type;

Cout<<”Enter account balance”;

Cin>>balance;

}

Void bank :: asignvalue(void)

{

Cout<<”assign initial value for balance”;

Cin>>value;

};

Void bank :: depositeammount(void)

{

Cout<<”Enter deposite amount”;

Cin>>am;

total=am+value;

}

Void bank :: withdraw(void)

{int with, sub=0;  
cout<<total;

Cout<<”Enter withdraw amount”;

Cin>>with ;

sub=value-with;

}

Void bank :: display(void)

{

Cout<<name;

Cout<< acnumber;

Cout<< type;

Cout<<balance;

Return 0;

}

**Ques 2** Write a program to read a matrix of size mxn from keyboard and display the same on the screen using function. (5)

**Ques3** Write a program in C++ to take the value from the user as input all sides of triangle and check whether the triangle is valid or not using switch statement. (5)