Software engineering Module 3 – Introduction to object oriented programming with C++

1. Introduction to C++

Q 1) Key differences between Procedural Programming and Object-oriented Programming are: -

Procedural Programming	**Object-Oriented Programming**
In this data is store in variables which makes the data less secure	In this the data store within an object and can be secured using access modifiers
Here the code in organized into functions	Here the code is organized in classes and objects
Here the program's made as a series of functions and in executed sequentially	Here the objects interact with each other through methods (functions)

Q 2) Advantages of OOP over POP are: -

- The data is stored within objects and can be hidden through protected and private mods.
- As the data and functions are organized into objects, updating code is easier as it won't affect the whole program
- It is easier to work in different teams as each team can work on different classes.

Q 3) Steps to set up a C++ development environment are: -

[Note: if you have already set up an C lang development environment then there is no need to follow this steps you can directly start developing]

- Search 'MinGW' on google and download the MinGW setup application
- Open the exe file and complete the set-up

- Open MinGW installer and select the ‘*mingw32-base*’ and ‘*mingw32-binutils*’ files to download.
- Find the path of MinGW folder in the C drive
- Open that folder and then open the bin folder
- Copy the path on the bin folder
- Now open ‘*Edit system environment variables’* application
- Click on Environment variables
- Now select the Path variable and click on edit
- Now click on New and then paste the path you copied before
- Open VS code and install C/C++ extension (by Microsoft) and code runner extension by (Jun Han)
- The set-up is done

Q 4) The main input/output operators in C++ are: -

- cout (output) – this is an object, which is used to print data on screen. It is used with left shift (<<) operator.
  Ex: - cout<<”This is an example”;

- cin (input) – this is an object, used to input from the users. It is used with the right shift (>>) operator
  ```
  Ex: -   int x;
          cin>>”\nEnter a number : “;
  ```

2. Variables, Data types and Operators

Q 1) Different data types available in C++ are: -

- integer – It is used to store whole numbers
- Long integer – it is also used to store whole numbers but provides a wider range
- Float – Used to store floating point numbers
- Double – also used to store floating point numbers but provides a wider range
- Char – used to store a single character

- String – Used to store array of characters

Q 2) In implicit type conversion the conversion is done automatically whereas in explicit type conversion is done manually by the programmer.

Q 3) Different types of operators in C++ are: -

- Arithmetic operator – These are used do basic arithmetic functions like addition, subtraction, division, and multiplication.
- Relational operator – These are used to find the relation between two variables. For example: <, >, <=, >=, ==.
- Assignment operators – These are used to assign values to variables. For example; =, *=, +=, -=, /=.
- Increment/ Decrement operators – These are used to increase or decrease the value of variable by 1. For example: --a, a--, ++a, a++.z
- Conditional operators – These are used to run different part of code basic on conditions. For example: - if, if … else, if …. else if …. Else, nested if.

Q 4) Purpose and use of constants in C++ is to store values which do not change throughout the execution of the program.

3. Control flow statements

Q 1) Conditional statements are statements used to perform different actions based on conditions. Conditional statements in C++ are: -

- If – It is used when only a single condition is to be evaluated. For example: If a student has scored more the 40% he passes.
- If … else – It is used when a single condition is evaluated on it being true or false. For Example: If a student scored more than 40% he passes else he fails.
- If … else if … else – it is used when multiple condition are to be evaluated. For example: Deciding student’s grade (A+, A, B+, B, C or fail) on basis of their score.
- Nested if – It is used when you need to evaluate a condition after evaluating some other conditions. For example:

- Switch case – It is an alternative to if…else if…else ladder but it can be used only with integer and character data types. For example: Any menu driven program.

Q 2) The Loops is C++ are: -

- For loop – It is used when you know the number of iterations
- While loop – It is used when you don't know the number of iterations
- Do-while loop – It is used when the loop is to be executed at least once

Q 3)

- Break – It is used to end the loop if a particular condition is true
- Continue – It is used to skip the iteration when a condition is true

Q 4) A nested control structure refers to placing one control structure like loops, conditionals, or switches inside another

4. Functions and scope

Q 1) Functions are a block of code. Explanation of functions declaration, definition and calling with example: -

Declaration: It is declared before the main function and after the header files (libraries)

Ex:-

```
// functions_return_type <function_name>(data_type(of the input variables));

void func();
```

Definition: The function is defined after the main function at the end of the code.

Ex:-

```
// functions_return_type <function_name>(data_type(of the input variables)
<variable_name>)

{
        //block of code
}
```

Calling: The function in called within the main function.

Ex:-

```
main(){
        func();
}
```

Q 2) Scope of variables refers to the part of program where a variable is accessible. Difference between local and global variable is: -

- Local variables – It is when the variable is declared inside a block of code (inside the {} brackets). It can be only accessed inside the {} brackets.
- Global variables – It is when the variable is declared outside of block of code. It can accessed anywhere in the program.

Q 3) Recursion is when a function calls itself, repeating a tasks over and over until the base condition is hit. Ex: - Calculating factorials.

Q 4) Function prototypes in C++ are

5. Arrays and Strings

Q 1) Arrays are collection of same kind of data, it is useful as we don't need use multiple variables to store the data. For example – To store scores of multiple subjects instead of using different variable for each subject we can create an array of the scores and access score of a particular subject with help of the index of the array.
One dimensional array store data in a list like format (like to store marks of a single student) whereas in multi-dimensional array the data can be store in form of tables (like to store data of multiple students).
Q 2) For string handling is C++ first we need to include fstream class file then: -

- To read from a file we create an object of class 'ifstream' then we use the open method and pass the name of the file with its extension as an argument.

- To write into a file we create an object of class ‘ofstream’ then we use the open method same way before to access the files
- If you want read and write both from using only singe object then you can create an object of class ‘fstream’.
- To use append mode create a ofstream object and while opening the file pass a second argument as ‘ ios :: app’

Q 3)

- Int arr[<size_of_array]; // initializing 1-D array
- Int arr[number_of_rows][number_of_columns]; // initializing 2-D array

Q 4) String operations and function in C++ are: -

- s.length() – It returns the length of the string
- s.append(s2) – Appends s2 to s
- s += s2 – this an alternate to the append method
- s.find(n) – It returns the index of n in s
- s.insert(i, n) – inserts n at i index in s

6. Introduction to Object-Oriented Programming

Q 1) Key concepts of Object-Oriented Programming are: -

- Class – It is a collection of data members and member functions
- Object – it allocates memory to a class
- Encapsulation – It is a wrapping up of data and functions altogether inside of the class.
- Inheritance – It means to drive properties of one class into anther class
- Polymorphism – It refers to functions or operators who have same name but have different forms.
- Abstraction – It refers to hiding the details (using the access modifiers) which are not needed to the user and only show data which is essential for the user.

Q 2)

- Class is a collection of data members and member functions
- A object allocates memory to the class

Example: -

We can create class 'rectangle' having data members – length and width and member function – get_details(), area() and perimeter()

And then use this in the main function we can define a of the class rectangle by writing

rectangle r;

and use the functions to assign object r a height and width and calculate its area and perimeter.

Q 3) Inheritance means using properties of one class in another class.

It's syntax is:

Class base{

};

Class derived : public base {

};

There many types of inheritance: -

- Single inheritance – When one class inherits properties from another class
  (Base -> derived)
- Multilevel inheritance – Here one class inherits a class and then that class is inherited by another. It's like a ladder of inheritance.
  (Base -> Intermediate Base -> derived)
- Multiple inheritance – Here one class inherits properties of more than one class.
  (Base1, Base 2 -> derived)
- Hierarchical inheritance – Inherit the classes by hierarchy
  (tree)
- Hybrid inheritance – It is a combination of any of the other types of inheritance.

Q 4) Encapsulation refers to wrapping data and functions together inside class.

It helps to keep data secure and only let data useful to the user be accessible.

It achieves this by using access modifiers: -

- Private – This is the default mode in C++. It allow the data to be used only within the class.
- Protected – This allows the data to be used in all classes in the program.
- Public – This allows the data to be used anywhere in the program.