Assignment 01

CS217 – Object Oriented Programming
Submission Due Date: 28-Feb-2021
Total Marks: 10

Q1. Define level of languages? Briefly explain each level of it with an example each.

Not included in mid-1

Machine Language:

The machine-level language is a language that comprises of a bunch of guidelines that are in the twofold structure 0 or 1. As we realize that PCs can see just machine directions, which are in twofold digits, i.e., 0 and 1, so the guidelines given to the PC can be just in paired codes. Making a program in a machine-level language is an extremely troublesome assignment as it is difficult for the developers to compose the program in machine directions. It is mistake inclined as it is difficult to comprehend, and its upkeep is additionally high. A machine-level language isn't compact as every PC has its machine guidelines, so in the event that we compose a program in one PC will presently don't be legitimate in another PC.

Assembly Language:

The low-level processing develop contains some conceivable orders, for instance, mov, add, sub, etc the issues which we were glancing in machine-level language are diminished fairly by using a widely inclusive kind of machine-level language known as low level registering build. Typical **examples** of large **assembly language** programs from this time are IBM PC DOS operating systems, the Turbo Pascal compiler and early applications such as the spreadsheet program Lotus 1-2-3.

High-Level Language:

The certain level language is a programming language that allows a computer programmer to make the undertakings which are self-sufficient of a particular kind of PC. The critical level vernaculars are considered as certain level since they are closer to human tongues than machine-level lingos.**Examples of high-level** programming **languages** in active use today include Python, Visual Basic, Delphi, Perl, PHP, ECMAScript, Ruby, C#, Java and many others.

Not included in exam

Q2) List any 4 names of programming paradigm with one coding example each.

Imperative Programming:

```
result = []
   i = 0
start:
   numPeople = length(people)
   if i >= numPeople goto finished
   p = people[i]
   nameLength = length(p.name)
   if nameLength <= 5 goto nextOne
```

```
  upperName = toUpper(p.name)
  addToList(result, upperName)
nextOne:
  i = i + 1
  goto start
finished:
  return sort(result)
```

Object Oriented Programming:

```
result:= List new.
people each: [:p |
 p name length greaterThan: 5 ifTrue: [result add (p name upper)]
]
result sort.
^result
```

Functional Programming:

```
sort(
 fix(λf. λp.
  if(equals(p, emptylist),
   emptylist,
   if(greater(length(name(head(p))), 5),
    append(to_upper(name(head(p))), f(tail(p))),
    f(tail(people)))))(people))
```

Declarative Programming:

```
select upper(name)
from people
where length(name) > 5
order by name
```

Q3) why do we need setter and getter? Rationalize with a coding example? Differentiate between public and private access specifiers with coding example each.

Getters and Setters allow you to effectively protect your data. This is a technique used greatly when creating classes. For each variable, a get method will return its value and a set method will set the value.

```
Class Account {

 Int accNum;

Public:

Int getaccNum(void){

Return accNum;}

Void setaccNum(int y){

  AccNum = x;}
```

Public: Allows Main Access to variables in class through the use of (.)

Private: Dosent Allow main the access of variables , variables can only be accessed through getter and setter functions.

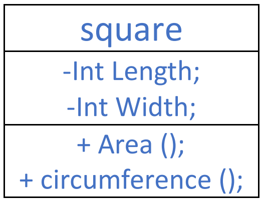
Public :

```
Class student {
     Public:
      String Name;
};
Int main(){
Student std;
Std.Name = ‘Anas’;
Cout<<std.Name<<endl;}
```

Private:

```
Class student{
Private:
String Name;
Public:
String getName(void){
Return Name;}
Void SetName(string str){
Name= str;
}
};
```

Q4) Draw the class diagram before writing a program for finding the area and circumference of a square.



square

-Int Length;
-Int Width;

+ Area ();
+ circumference ();