Prajapati Yash P. 20162121023 (BDA)

ESFP-2

Date
Page 1

Assignment - 1

1. Illustrate following concepts with example.
(a) Class & Object
(b) Data hiding
(c) Abstraction
(d) Inheritance

Ans (a) Class & Object :-

C++ is object oriented programming language. A class is user-defined data type that we can use in our program and it works as an object constructor or a "blueprint" for creating objects.

Eg:-
```
#include <iostream>
using namespace std;

class FirstClass() {          <-- class creation
  public:
  void data() {
    cout << "this is function";
  }
};
int main () {
  FirstClass F;               <-- object creation
  F.data();
  return 0;
}
```

(b) Data Hiding :-

Data Hiding is a process of combining data & functions into single unit. The data is hidden, so that it will be safe from accidental manipulation.

Eg:-
```
#include <iostream>
using namespace std;

class DataHiding {
private:
void dataSet() { int num;
cout << "enter num: ";
cin >> num;
}
public:
cout << "this data is public";
}

int main {
DataHiding D;
D.dataSet();
return 0;
}
```

(c) Abstraction :- Abstraction refers to the act of representing essential features without including the background detail of explanation.

Eg:-
```
#include <iostream>
using namespace std;

class Abs
{
private :
 int n1, n2;
public:
void set (int a, int b) {
    n1 = a;
    n2 = b;
  }
void display()
{
cout << n1;
cout << n2;
};
int main() {
  Abs a;
  a.set(50, 10);
  a.display();
  }
```

(d) Inheritence :-

Inheritence is used to reuse the properties & behaviours of existing class without redefining the old class.

There are 5 types of inheritance, single, multiple, multi-level, hierarical, and hybrid.

Eg:-
```
class father {
    public:
    int a, b, c;
};
class child : public father {
    int d;
}
```

(2) Explain the importance of following in C++ environment.

(a) setw manipulator

(b) static_cast

(c) using namespace.

Ans (a) This manipulator is used to specify the minimum no. of char position on the output field a variable will consume. It is declared under header file <iomanip>.

(b) Static - cast :- In C++, static - cast will allow the compiler to check whether the pointer & the data are of same type or not. If not it will raise incorrect pointer assignment exception during compilation.

(c) using namespace :- Namespaces allow us to group named entities that otherwise would have global scope into narrower scopes, this allows organising the elements of programs into different logical scopes refered by names.

(3) Explain concept of overloaded functions with an example.

Soln Function overloading is a feature in C++ where two or more functions can have the same name but different parameters.

Eg:-
```
#include <iostream>
using namespace std;

void print (int i) {
cout << " Here is int " << i;
}

void print (double f) {
cout << "Here is double " << f;
}
```

```
void print(char const* c)
{
 cout<<" Here is char* "<< c;
}
int main() {
 print(70);
 print(210.2);
 print("yash");
 return 0;
}
```

(4) Illustrate the concept of default argument function with example.

Soln Default argument is value provided in a function declaration that is auto assigned by the compiler if the caller of the function doesnt provide a value for the argument with a default value.

Eg:-
```
# include <iostream>
 using namespace std;

int sum(int a; int b; int c=0; int d=0)
{
  return (a+b+c+d);
}
int main() {
 cout << sum(2,3);
```

```
cout << sum (2,3,5);
cout << sum (2,3,5,7);
return 0;
}
```

(5) Describe how const keyword is applicable while passing arguments.

Sol" const variable is used in C++ to make an element constant. using keyword "const".

Example :-

```
# include <iostream>
using namespace std;
int main (){
 const int a=100;
 cout << a;
 return 0;
 }
```

output = 100.

(6) Write a program that contains the following functions. A starLine that accepts a no. & display the lines of stars. Area circle that accepts the radius & returns the area of circle. Area triangle that accepts the arguments & returns the area of triangle.

Sol"

```
#include <iostream>
using namespace std;

void star (int n)
{  int i=0;
   for(i=0; i<n; i++) {
   cout << "*";
   }
}

void areaC (int r)
{
   int areaC=0;
      areaC = 3.14*r*r;
   cout<<" Area of circle with radius "<<r<<" is: "
                                 << areaC;
   }

void areaT (int b; int h)
{
   int areaT;
cout<< "Area of triangle with base "<<b<<" and
          heigh "<<h<<" is: " << areaT;
   }
int main (){

   int n, r, b, h;
   cout <<" Enter no. of stars: ";
   cin >>n;
   cout << "Enter radius, base, height: ";
   cin >>r>>b>>h;
```

```
star (n);
areaC (r);
areaT (b, h);
return 0;
}
```

(7) Define a class to represent a bank acc. Every acc has name of depositor; acc no.; type of account & balance amount Account supports following functionalities.
(a) To assign initial values.
(b) To deposit an amount
(c) To withdraw an amount after checking the balance after verifying for the sufficient balance.
(d) To display all the details of bank account.

Sol^n

```
#include <iostream>
using namespace std;

class Bank {
public:
    char name[10];
    int acc_no;
    char acc_type[20];
    int balance;

    void initialize () {
        cout << "Enter bank acc. holder's name";
```

```
gets(name);
cout<<"Enter acc. type:";
gets(acc_type);
cout<<"Enter acc. number";
cin>>acc_no;
cout<<"Enter balance";
cin>>balance;
}
void deposit()
{
int bal;
cout<<"Enter amount to deposit";
cin>>bal;
balance+=bal;
cout<<"Balance after deposit"<<balance;
}
void check(){
int bal;
cout<<"Enter your balance";
cin>>balance;
cout<<"Enter amount to withdraw";
cin>>bal;

if(bal<=balance)
{
cout<<"Remaining Balance:";
balance=balance-bal;
cout<<balance;
}
else{exit(0);}}
```

```
void display () {
    cout << "Name";
    puts (name);
    cout << "In Balance ";
    cout << balance;
};

int main () {
    int i;
    Bank B;
    B.initialize();
    cout << "1. Information \n2. Deposit \n3. Withdraw";
    cin>>i;

    if (i==1)
      {
        B.display();
      }
    else if (i==2) {
        B.deposit();
      }

    else if (i==3) {
        B.check();
        }

    return 0;
}
```