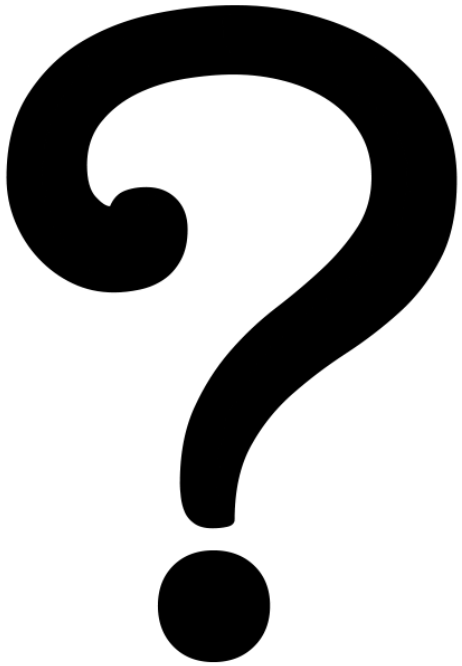
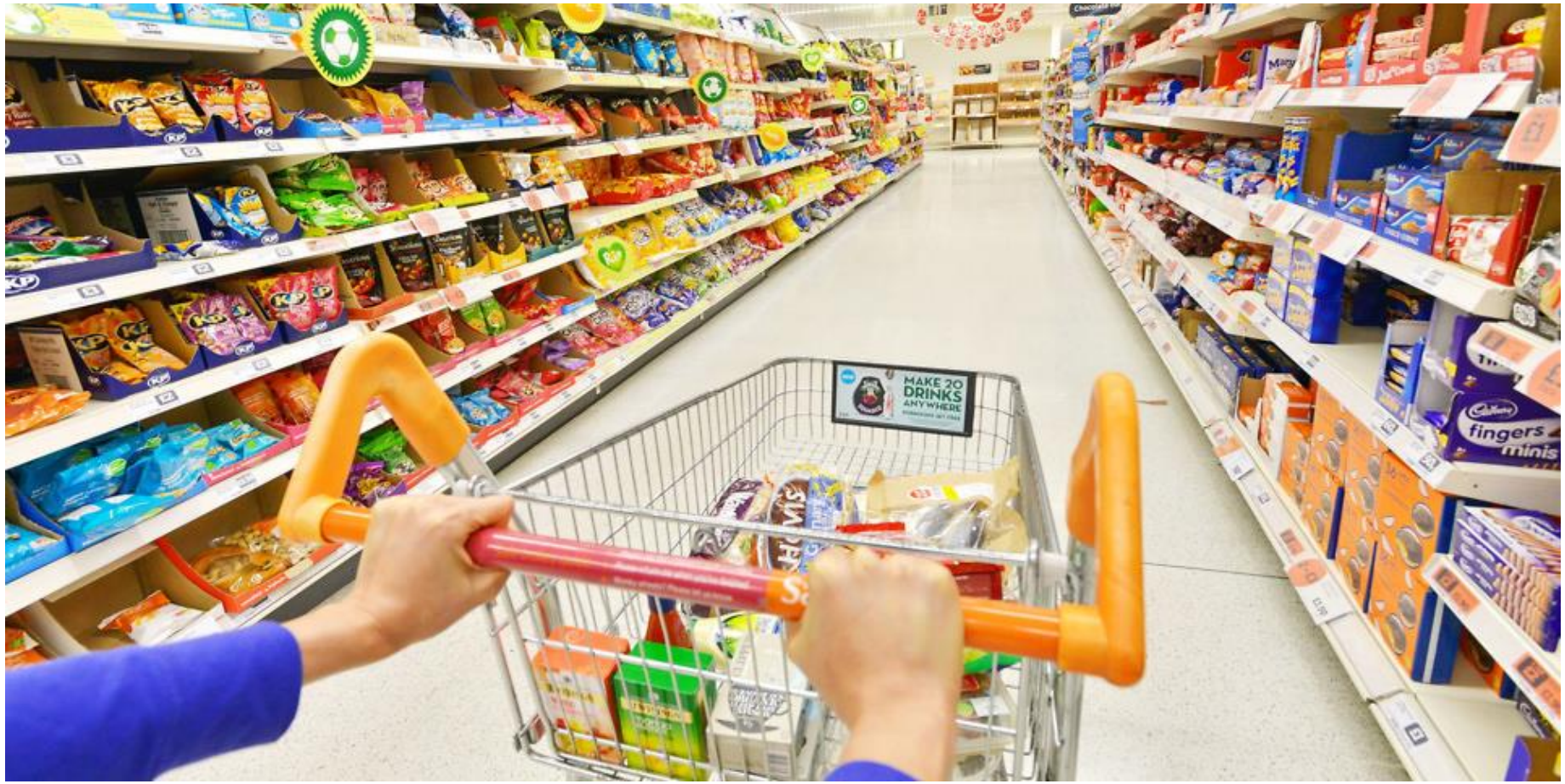
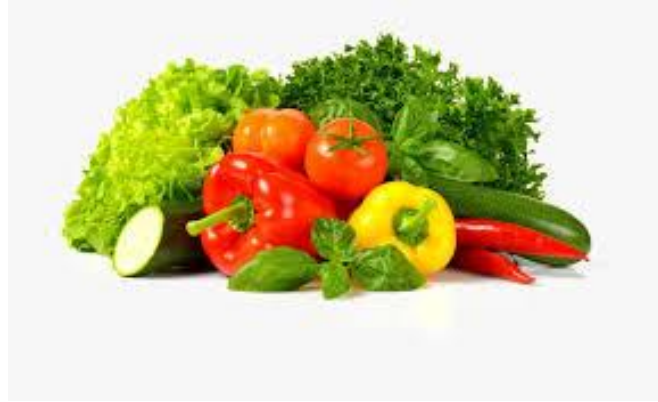

FACE

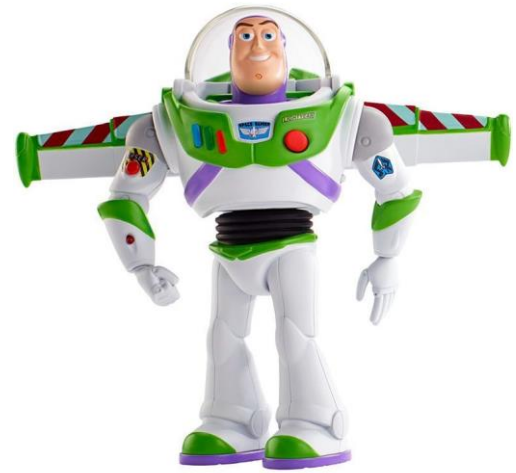
Object Oriented Programming



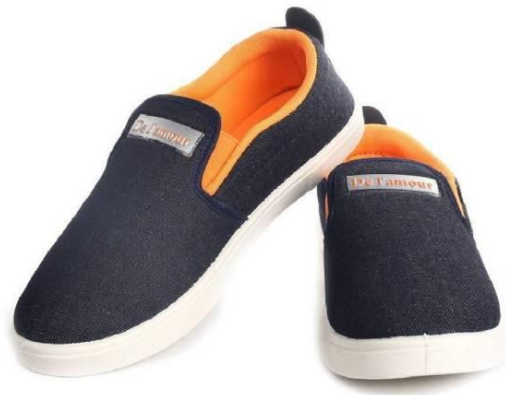
Guess the place!!





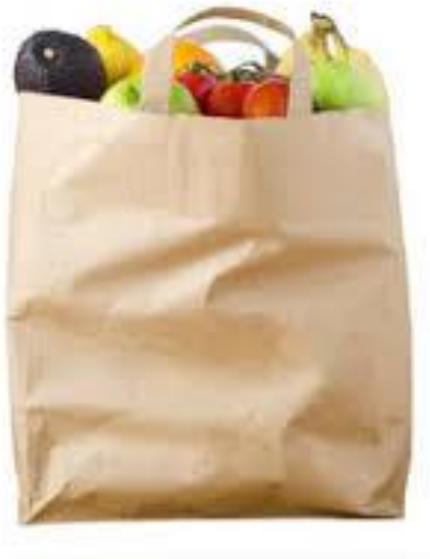



popchips
barbeque
Goldfish
Cheddar
M&M's
Cookies
Welch's
Fruit Snacks
CHEWY
Grandma's
OREO
CHEEZ-IT
White Cheddar
Chex Mix
ZOO
ANIMAL CRACKERS
POPCORNERS
SEA SALT
ANDY CAPP'S
HOT FRIES
Fiber One
AIR HEADS



Billing Counter

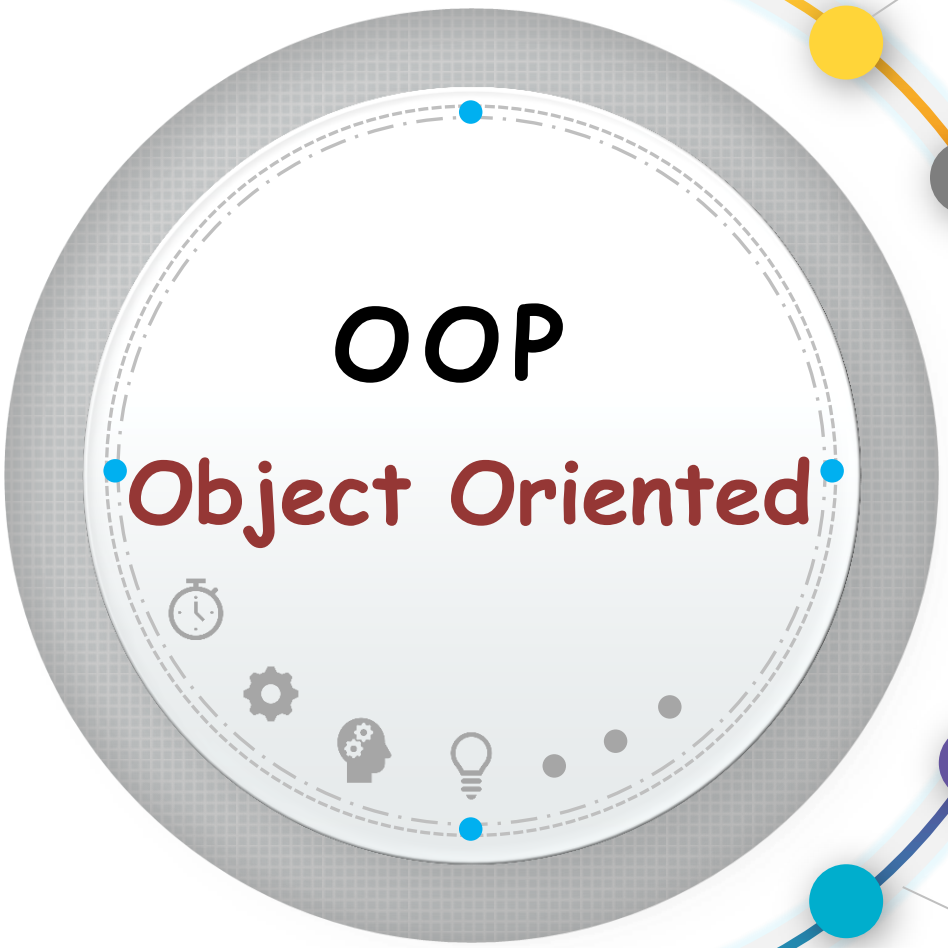
PACKING



Object Oriented Programming

- Revolves around data
- Main programming unit of OOP is object
- Although C++ is classified as a partial object-oriented programming language
- Data security is ensured

FACE



01 Class

02 Object

03 Abstraction

04 Encapsulation

05 Inheritance

06 Polymorphism

FACE

Class

- User defined data type
- Constitutes **Data members** and **Member Functions.**

Objects

- Are instances of class
- Holds separate copies of data members
- Are initialized using special class functions called Constructors
- And whenever the object is out of its scope, Destructor is called

class	object
Container - collection of variables and functions	Object is a instance of class
No memory is allocated -during class declaration	Memory is allocated - during object declaration
One class definition - only once in the program.	For one class multiple objects can be created.

```
Class Supermarket
{
        vegetables, snacks, toys, gifts
        purchase()
        billing()
        packing()
} Customer;
```

Data Members

Member Function

```
1  #include <iostream>
2  class SuperMarket                                    //Class Name
3  {
4        public:                                              //Access specifier
5        int data;                                            //Data member
6        void purchase()                                      //Member function
7        {
8        std::cout<<"I Purchased "<<data<<" Books";           //Standard Output
9        }
10 };
11 int main()                                                 //Driver Program
12 {
13       SuperMarket cus1;                                 //Object Creation
14       cus1.data=10;                                        //Assigning values
15       cus1.purchase();                                     //Function call
16       return 0;
17 }
```

Output

```
I Purchased 10 Books
```

```
1  #include <iostream>
2  class SuperMarket                                    //Class Name
3  {
4        public:                                        //Access specifier
5        int data;                                      //Data member
6        void purchase()                                //Member function
7        {
8        std::cout<<"I Purchased"<<data<<" Books";      //Standard Output
9        }
10 }SuperMarket cus1;                                  //Object Creation
11 int main()                                          //Driver Program
12 {
13       cus1.data=10;                                  //Assigning values
14       cus1.purchase();                               //Function call
15       return 0;
16 }
```

Output

```
I Purchased 10 Books
```

```
1  #include <iostream>
2  class SuperMarket
3  {
4        public:
5        int data;
6        void purchase()
7        {
8              std::cout<<"I Purchased "<<data<<" Books\n";
9        }
10 };
11 int main()
12 {
13       SuperMarket cus1,cus2,cus3;
14       std::cin>>cus1.data>>cus2.data>>cus3.data; //Input: 10 20 30
15       cus1.purchase();
16       cus2.purchase();
17       cus3.purchase();
18       return 0;
19 }
20
21
22
```

OUTPUT

```
10
20
30
```

```
1  #include <iostream>
2  class SuperMarket
3  {
4        public:
5        int data;
6        void purchase(int i)
7        {
8              std::cout<<"Customer "<<i+1<<" Purchased "<<data<<" Books\n";
9        }
10 };
11 int main()
12 {
13       int i,n;
14       std::cin>>n;
15       SuperMarket customer[n];              //Declaring array of Objects
16       for(i=0;i<n;i++)
17             std::cin>>customer[i].data;
18       for(i=0;i<n;i++)
19             customer[i].purchase(i);
20       return 0;
21 }
22
```

Console

```
3
20
23
2
Customer 1 Purchased 20 Books
Customer 2 Purchased 23 Books
Customer 3 Purchased 2 Books
```

Member functions in class

```
1  #include<iostream>
2  using namespace std;
3
4  class Box
5  {
6        public:
7           double l;
8           double b;
9           double h;
10          double getVolume(void)
11          {
12             return l * b * h;
13          }
14 };
15 int main()
16 {
17       Box b1;
18       b1.b;
19       b1.h = 1, b1.l = 1;
20       cout << b1.getVolume();
21 }
22
```

Member function – defined inside the class

Member functions in class

```
1  #include<iostream>
2  using namespace std;
3  class Box
4  {
5        public:
6           double l;
7           double b;
8           double h;
9           double getVolume(void);
10 };
11 double Box :: getVolume(void)
12 {
13       return l * b * h;
14 }
15
16 int main()
17 {
18       Box b1;
19       b1.b;
20       b1.h = 1, b1.l = 1;
21       cout << b1.getVolume();
22 }
```

Member function – declared inside the class

Member function – defined outside the class

Scope resolution operator

- Used to get hidden names due to variable scopes

- can be used as both unary and binary

FACE

```
1  #include <iostream>
2  using namespace std;
3  int my_var = 0;
4  int main()
5  {
6      int my_var = 0;
7      ::my_var = 1;
8      my_var = 2;
9      cout << ::my_var << ", " << my_var;
10     return 0;
11 }
12
13
14
15
16
17
18
19
20
21
22
```

OUTPUT

```
1, 2
```

```
1  #include<iostream>
2  using namespace std;
3
4  class A
5  {
6      public:
7      void fun();
8  };
9  void A::fun()
10 {
11    cout << "fun() called";
12 }
13
14 int main()
15 {
16    A a;
17    a.fun();
18    return 0;
19 }
20
21
22
```

OUTPUT

```
fun() called
```

Question 1

How structures and classes in C++ differ?

A) In Structures, members are public by default whereas, in Classes, they are private by default

B) In Structures, members are private by default whereas, in Classes, they are public by default

C) Structures by default hide every member whereas classes do not

D) Structures cannot have private members whereas classes can have

Question 2

Predict the output:

```
#include <iostream>
    using namespace std;
    class Box
    {
        public :
        double l;
        double b;
        double h;
    };
```

```
int main()
    {
        Box B1;
        double vol;
        B1.h = 5;
        B1.l = 6;
        B1.b = 7.1;
        vol=B1.h*B1.l*B1.b;
        cout << vol;
        return 0;
    }
```

A) 210 **B)** 216 **C)** 213 **D)** 214

Question 3

Predict the output:
// Assume that integers take 4 bytes.

```
#include<iostream>
 using namespace std;
 class Test
{
  static int i;
  int j;
};
```

```
int Test::i;
 int main()
{
    cout << sizeof(Test);
    return 0;
}
```

A) 2

B) 16

C) 8

D) 4

Question 4

What is the other name used for functions inside a class?

A) Member variables

B) Member functions

C) Class variables

D) Class functions

Question 5

C++ is _____

A) Procedural programming language

B) object oriented programming language

C) functional programming language

D) both procedural and object oriented programming language

Question 6

Which operator is used to allocate an object dynamically of a class in C++?

A) Scope resolution operator

B) Conditional operator

C) New operator

D) Membership access

Question 7

What happens when we try to compile the class definition in following code snippet?

```
class Birds {};
class Peacock : protected Birds {};
```

Question 7

A) It will not compile because class body of Birds is not defined.

B) It will not compile because class body of Peacock is not defined.

C) It will not compile because a class cannot be protectedly inherited from other class.

D) It will compile successfully.

Question 8

Object oriented programming employs_____ programming approach.

A) top-down

B) procedural

C) bottom-up

D) all of these

E) none of these

FACE

Question 9

In C++, Class object created statically(e.g. Car obj; and dynamically (Car *obj = new Car() ;) are stored in memory

A) Stack, heap

B) Heap, heap

C) Heap, stack

D) Stack, stack

Question 10

What will be the output of the following C++ code?

```
#include <iostream>
using namespace std;
class rect
  {
   int x, y;
   public:
   void val (int, int);
   int area ()
   {
     return (x * y);
   }
  };
```

```
void rect::val (int a, int b)
 {
   x = a;
   y = b;
  }
  int main ()
  {
  rect rect;
  rect.val (3, 4);
 cout << "rect area: " <<
                  rect.area();
   return 0;
    }
```

Question 10

A) rect area: 24

B) rect area: 12

C) compile error because rect is as used as class name and variable name in line #20

D) rect area: 56

Question 11

What will be the output of the following C++ code?

```
#include <iostream>
    using namespace std;
    class number
    {
        int i;
        public:
        int geti();
        void puti(int j);
    };
    int number::geti()
    {
        return i;
    }
```

```
void number::puti(int j)
    {
        i = j;
    }
    int main()
    {
        number s;
        s.puti(10);
        cout << s.geti( );
        return 0;
    }
```

Question 11

A) 10

B) 11

C) 20

D) 22

FACE

Question 12

How many objects can present in a single class?

A) 1

B) 2

C) 3

D) As many as possible

FACE

THANK YOU

FACE