Address Typecasting :-

why *i not pointer p?

int * p;

Pointer p.

*p → p के pass जो address है वहा जाता है or us value ko read krna है।

if we use this then we would not get idea about how much bytes we have to read and how have we interpreted them

int i = 65;

char c = i;

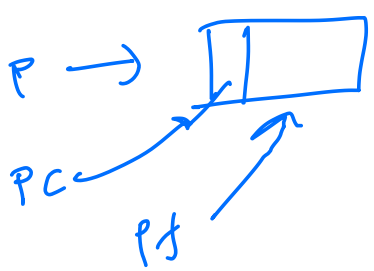
c 65

int * p = &i;

Char *pc = p;

*p

float *pf = p;

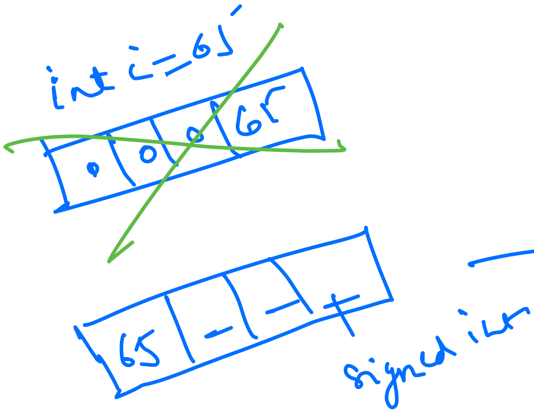


Typecasting ;- a type of data copy in another different type e.g:-

int i = 65;

char c = i;

implicit typecasting.



Called :

(1) Little endian

mostly system mein means least byte pehle likhi

(2) Big endian

int i = 65;

int *p = &i;

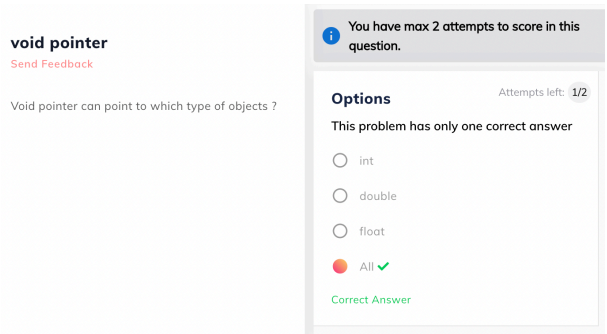
char *pc = (char*) p

explicit typecasting

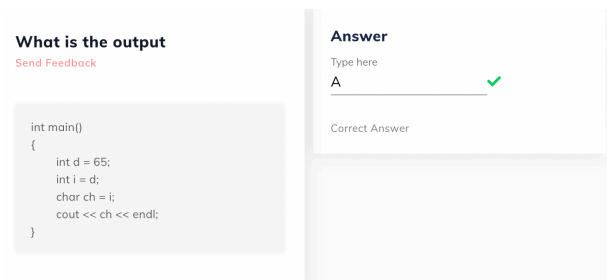
```
#include <iostream>
using namespace std;

int main() {
  int i = 65;
  char c = i;
  cout << c << endl;

  int * p = &i;
  char * pc = (char*)p;

  cout << p << endl;
  cout << pc << endl;

  cout << *p << endl;
  cout << *pc << endl;
  cout << *(pc + 1) << endl;
  cout << *(pc + 2) << endl;
  cout << *(pc + 3) << endl;
}
```

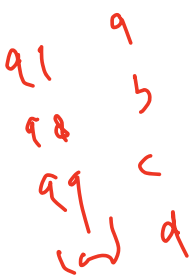
```
Codings-Mac-mini:Inception codingninjastudio$ ./a.out
A
0x7fff59755b0c
A
65
A


Codings-Mac-mini:Inception codingninjastudio$
```

Q1:-

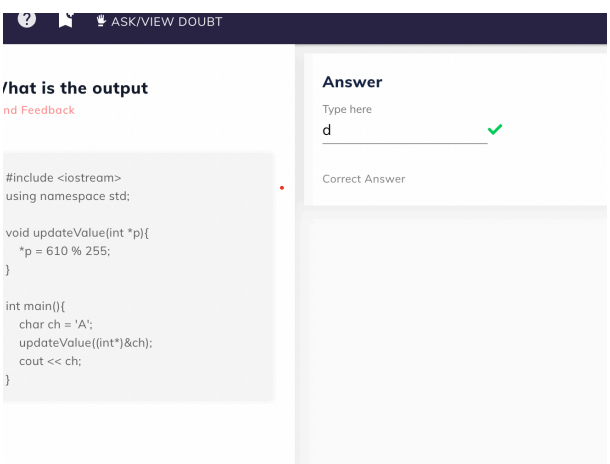

void pointer
Send Feedback
Void pointer can point to which type of objects ?
You have max 2 attempts to score in this question.
Options
Attempts left: 1/2
This problem has only one correct answer
int
double
float
All
Correct Answer

Q2:-

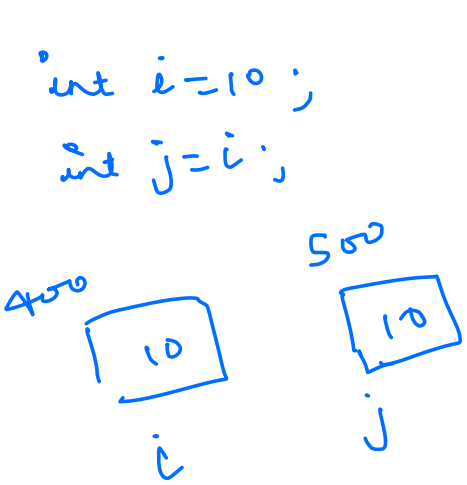

What is the output
Send Feedback
int main()
{
int d = 65;
int i = d;
char ch = i;
cout << ch << endl;
}
Answer
Type here
A
Correct Answer



Q3:-


ASK/VIEW DOUBT
/hat is the output
nd Feedback
#include <iostream>
using namespace std;
void updateValue(int *p){
*p = 610 % 255;
}
int main(){
char ch = 'A';
updateValue((int*)&ch);
cout << ch;
}
Answer
Type here
d
Correct Answer

Reference and pass by Reference :-



But basically we can use same address for both i and j by using ampersent (&)


```
 1 #include <iostream>
 2 using namespace std;
 3
 4 void increment(int& i) {
 5   i++;
 6 }
 7
 8 int& f(int n) {
 9   int a = n;
10   return a;
11 }
12
13 int* f2() {
14   int i = 10;
15   return &i;
16 }
17
18 int main() {
19   int* p = f2();
20
21   int i;
22   i = 10;
23
24   int& k = f(i);
25
26
27   increment(i);
28   cout << i << endl;
29
30   int& j = i;
31
32   i++;
33   cout << j << endl;
34   j++;
"reference.cpp" 41L, 372C written
```

```
 1 #include <iostream>
 2 using namespace std;
 3
 4 void increment(int& i) {
 5   i++;
 6 }
 7
 8 // bad practice
 9 int& f(int n) {
10   int a = n;
11   return a;
12 }
13
14 // bad practice
15 int* f2() {
16   int i = 10;
17   return &i;
18 }
19
20 int main() {
21   int* p = f2();
22
23   int i;
24   i = 10;
25
26   int& k = f(i);
27
28
29   increment(i);
30   cout << i << endl;
31
32   int& j = i;
33
34   i++;
"reference.cpp" 43L, 404C written
```

Reference Variable

Send Feedback

What is the correct syntax of declaring and defining a reference?

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- int a = 10; int &b; b = a;
- int a = 10; int &b = a; ✔
- int a = 10; int b = &a;
- int a = 10; int &b = &a;

Correct Answer

What is the output

Send Feedback

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

void func(int p){
   cout << p << " ";
}

int main(){
   int i = 10;
   int &p = i;
   func(p++);
   cout << i;
}
```

Answer

Type here

10 11 ✔

Correct Answer

What is the output

Send Feedback

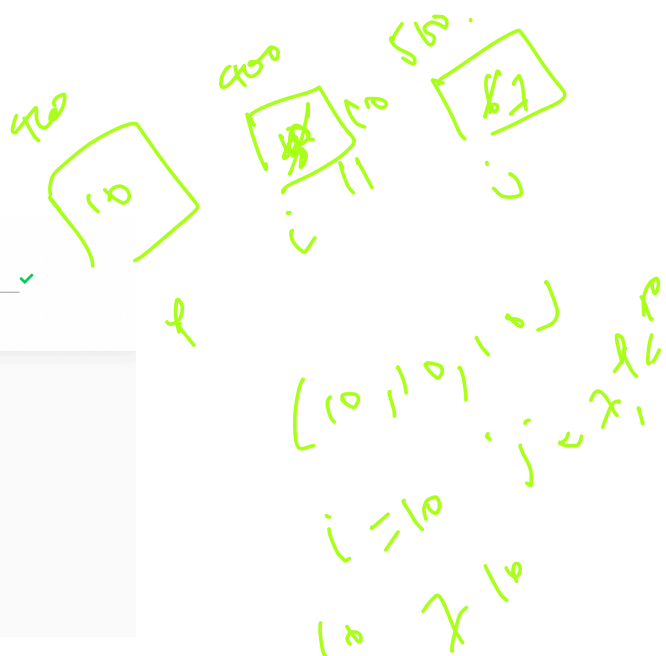
```
#include <iostream>
using namespace std;

void func(int i, int& j, int p){
   i++;
   j++;
   p++;
}

int main(){
   int i = 10;
   int j = 6;
   int &p = i;
   func(i, j, p);
   cout << i << " " << j << " " << p;
}
```

Answer

Type here

10 7 10 ✔

Correct Answer



Dynamic memory allocation:-

```
int a[20]
```

✓ Compile time pr

Ha clear Hona chaiye Ki kis size ka array banana chahte ho!

```
int n;
cin>>n;
int a[n];
```

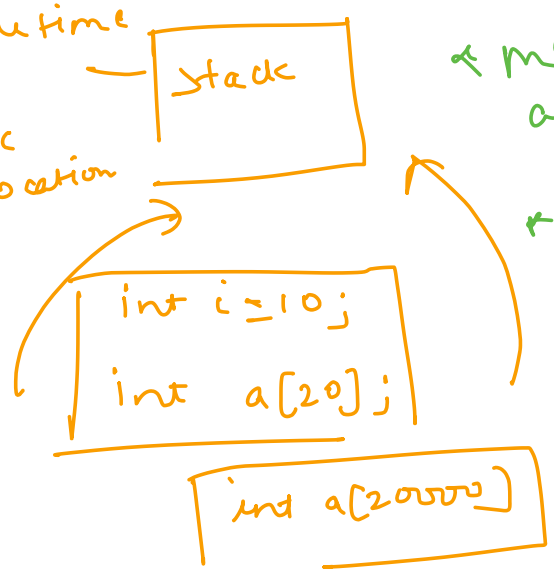
X

We should not do this Runtime pr to it will declare size of array.

Note: When we allocate memory there is two type of memory

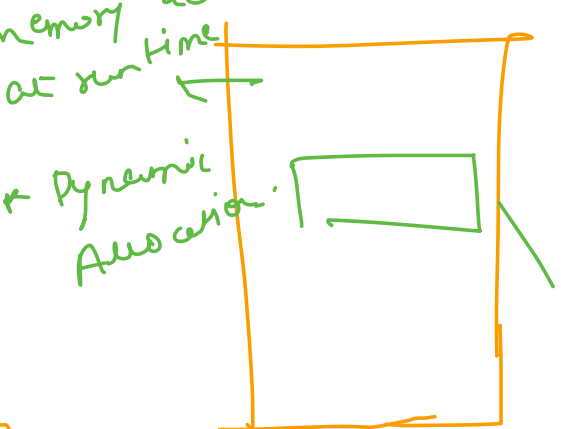
(i) Stack — little bit less memory

* decide at compile time
* Static Allocation



(ii) Heap — Big memory

* memory decide at run time
* Dynamic Allocation.



```
int n;
cin>>n; → System stack
int a[n]; → Heap
```

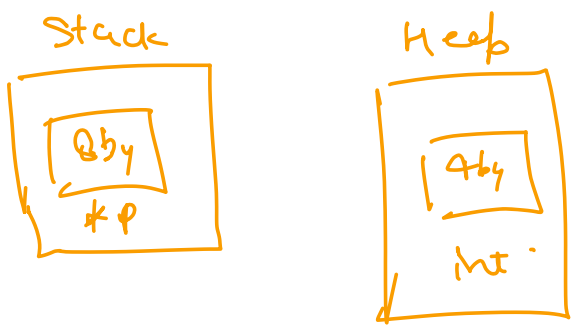
Lekar chalya.

Dynamic memory allocation.

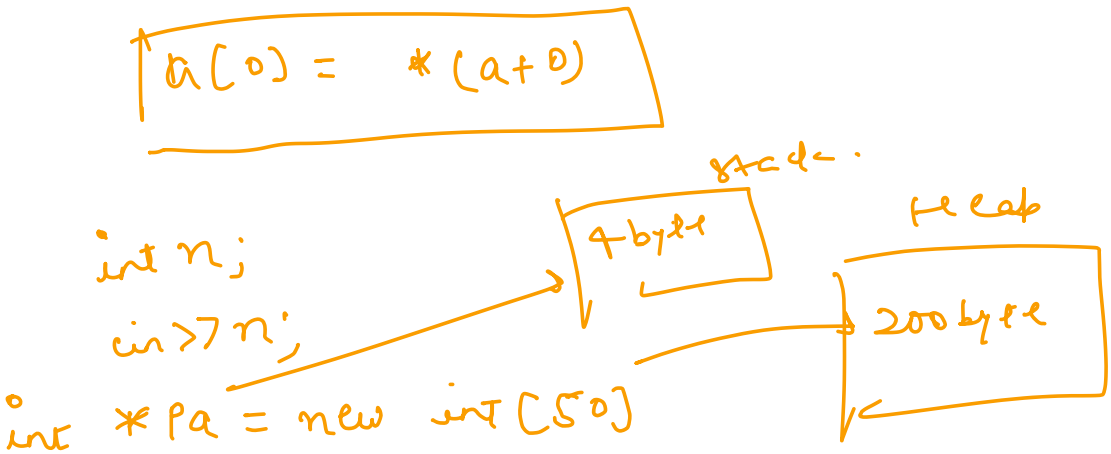
it gives us memory address:

```
int *P = new int;
```

```
~/Documents/Inception — vi dy
 1 #include <iostream>
 2 using namespace std;
 3
 4 int main() {
 5   int * p = new int;
 6   *p = 10;
 7   cout << *p << endl;
 8
 9   double *pd = new double;
10
11   char* c = new char;
12
13
14 }
15
~
~
```



for array - int *P = new a[50];




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

int main() {
  int * p = new int;
  *p = 10;
  cout << *p << endl;

  double *pd = new double;

  char* c = new char;

  int* pa = new int[50];
  int n;
  cout << "Enter num of elements" << endl;
  cin >> n;

  int* pa2 = new int[n];
  for (int i = 0; i < n; i++) {
    cin >> pa2[i];
  }

  int max = -1;
  for (int i = 0; i < n; i++) {
    if (max < a[i]) {
      max = a[i];
    }
  }
  cout << max << endl;
}
```

~/Documents/inception — vi dynamic_allocation2.cpp

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

int main() {

  while (true) {
    int *p = new int;
  }

  while (true) {
    int i = 10;
  }


}
```

Static memory

① auto release on basis of scope

Dynamic memory

② manual release is required.

int *p = new int; → memor allocate

deallocate ← delete p; → it is not deleting p it is delete basically where p was pointing.

p = new int;
delete p; → single deletion

p = new int[100];
delete [] p; → array deletion.

malloc()

take one argument

malloc() allocates a memory block of given size (in bytes) and returns a pointer to the beginning of the block. malloc() doesn't initialize the allocated memory. If you try to read from the allocated memory without first initializing it, then you will invoke undefined behavior, which will usually mean the values you read will be garbage.

calloc()

take two argument

calloc() allocates the memory and also initializes every byte in the allocated memory to 0. If you try to read the value of the allocated memory without initializing it, you'll get 0 as it has already been initialized to 0 by calloc().

Dynamic memory allocation

Send Feedback

In CPP, dynamic memory allocation is done using _____ operator.

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- calloc()
- malloc()
- allocate
- new ✔

Correct Answer

Deallocate memory

Send Feedback

How will you free the memory allocated by following program -

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

int main() {
    int *a = new int;
}
```

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- free a;
- delete *a;
- delete a; ✔
- new a;

Correct Answer

Create array

Send Feedback

Correct statement for creating an integer array of size 5, dynamically -

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- int *arr[] = new int[5];
- int *arr = new int[5]; ✔
- int arr = new int[5];
- int *arr[5] = new int;

Correct Answer

Deallocate array

Send Feedback

Correct statement for deallocating the array is -

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- delete [] arr; ✔
- delete arr;
- delete *arr;
- delete [] *arr;

Correct Answer

Delete memory

Send Feedback

On deleting a dynamic memory, if the pointer value is not modified, then the pointer points to?

You have max 2 attempts to score in this question.

Options Attempts left: 0/2

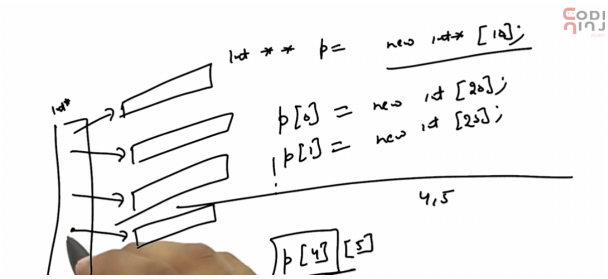
This problem has only one correct answer

- NULL
- Other dynamically allocated memory
- The same deleted memory location ✔
- It points back to location it was initialized v

Correct Answer

Dynamic Allocation of 2D Arrays :-

new int[10][20] X (Not allowed)



```
~/Documents/Inception — vi multidimarray.cpp
 1 #include <iostream>
 2 using namespace std;
 3
 4 int main() {
 5   int m, n;
 6   cin >> m >> n;
 7   int** p = new int*[m];
 8   for (int i = 0; i < m; i++) {
 9     p[i] = new int[n];
10     for (int j = 0;j <n; j++) {
11       cin >> p[i][j];
12     }
13   }
14 }
15
```


if you create new at any place then you have to deallocate that memory.

~/Documents/Inception — vi multidimarray.cpp

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

int main() {
  int m, n;
  cin >> m >> n;
  int** p = new int*[m];
  for (int i = 0; i < m; i++) {
    p[i] = new int[i + 1];
    for (int j = 0;j <n; j++) {
      cin >> p[i][j];
    }
  }

  for (int i = 0; i < m; i++) {
    delete [] p[i];
  }
  delete [] p;
}
```


2D arrays

Send Feedback

How to Dynamically create a Two Dimensional Array in C++?

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem has only one correct answer

- int *twoDArray = new int[10][10];
- int **twoDArray = new int*[10]; ✔
- int **twoDArray = new int[10][10];

Correct Answer

Jagged array is array **of** arrays such that member arrays can be of different sizes, i.e., we can create a 2-D array but with a variable number of columns in each row. These type of arrays are also known as Jagged arrays.

Example:

```
arr[][] = { {0, 1, 2},
            {6, 4},
            {1, 7, 6, 8, 9},
            {5}
          };
```

Jagged arrays

Send Feedback

Can we create Jagged Arrays in C++ ?

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem has only one correct answer

- Yes ✔
- No

Correct Answer

Given the following CPP Program, Select the CORRECT Syntax of deleting this 2D array? Check all that apply.

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

int main()
{
   int **twoDArray = new int*[10];
   for(int i = 0; i < 10; i++)
      *(twoDArray + i) = new int[10];
}
```

Options are -

```
A.
delete twoDArray [][];

B.
for(int i = 0; i < 10; i++)
   delete twoDArray[i];
   delete twoDArray[];

C.
for(int i = 0; i < 10; i++)
   delete [] twoDArray[i];
   delete[] twoDArray;

D.
for(int i = 0; i < 10; i++)
   delete twoDArray[i][];
   delete[] twoDArray;
```

You have max 2 attempts to score in this question.

Options

This problem has only one correct answer

- A
- B
- C ✔
- D

Correct Answer

Macros And Global Variable :-

define.

π = 3.14

return 3.14 * r * r ;

① it will be hard to change this value.

double pi = 3.14

pi * r * r ;

#include <iostream> → Preprocessor directory.

#define PI 3.14

Basically code run karne से पहले Jaha bhi PI होगा वो उसको 3.14 कर देगा।

Global Variables.

Never use global variable क्योंकि कोई भी कहीं भी change kr सकता है |

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

int a;

void g() {
  a++;
  cout << a << endl;
}

void f() {
  cout << a << endl;
  a++;
  g();
}

int main() {
  a = 10;
  f();
  cout << a << endl;
}
```

5, 8

9 +

What is the output

Send Feedback

What is the output of the following program?

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

int x = 1;

void print(){
  int x = 2;
  {
    int x = 3;
    cout << x << endl;
  }
}
int main(){
  print();
  return 0;
}
```

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only only one correct answer

- 1
- 2
- 3 ✔
- Error

Correct Answer

Inline and Default argument :-

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

int main() {
  int a, b;
  cin >> a >> b;

  int c;
  if (a > b) {
    c = a;
  } else {
    c = b;
  }
}
```

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

int main() {
  int a, b;
  cin >> a >> b;

  int c = (a > b) ? a : b;

}
```

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

int max(int a, int b) {
  return (a > b)? a : b;
}

int main() {
  int a, b;
  cin >> a >> b;

  int c = max(a, b);

  int x,y;
  x = 12;
  y = 34;

  int z = max(x, y);
}
```

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

inline int max(int a, int b) {
  return (a > b)? a : b;
}

int main() {
  int a, b;
  cin >> a >> b;

  int c = max(a, b);

  int x,y;
  x = 12;
  y = 34;

  int z = max(x, y);
}
```

inline → basically function की body को जिस Place पर use हो रहा वहा copy कर देता.

* Note :- inline function छोटे 1 line के code के लिए Acha kaam karta है यदि 2-3 line

वह toh depend Karta की काम करेगा या नहीं or 3 से ज्यादा वह से काम नहीं करेगा।

Default argument:- it mean ye humpe depend Karta ki कौन कौन Hum Kitne argu pass kr rahe से agar argument kam से दो तो फिर default value देगी usko access करेगा

Output de dega.

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

int sum(int a[], int size, int si = 0) {
  int ans = 0;
  for (int i = si; i < size; i++) {
    ans += a[i];
  }
  return ans;
}

int sum2(int a, int b = 0, int c = 0, int d = 0) {
  return a + b + c + d;
}

int main() {
   int a[20];
   // input code for a

   cout << sum(a, 20) << endl;
}
```

Inline functions

Send Feedback

Inline functions are useful when _____

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- Function is large with many nested loc
- Function has many static variables
- Function is small and we want to avoi
- None of the above

Correct Answer

Solution Description

Inline functions are used to reduce the function call overhead. Inline function is a function that is expanded in line when it is called.

What is the output

Send Feedback

What is the output of the following program ?

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

int getValue(int x = 0, int y = 0, int z){
   return (x + y + z);
}

int main(){
  cout << getValue(10);
  return 0;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem has only one correct answer

- 10
- 0
- 20
- Compilation Error ✔

Correct Answer

Solution Description

Default parameters must be added last in C++ functions

Default arguments

Send Feedback

Which of the following statement is correct?

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem has only one correct answer

- Only one parameter of a function can be a default paramete
- Minimum one parameter of a function must be a default par
- All the parameters of a function can be default parameters.
- No parameter of a function can be default.

Correct Answer

Constant Variable:-

const int i ; } X
i = 10

const int i = 10 ; } X
i = 12 ;

* Path const होता है actual Storage const नहीं होता है।

you can write any one of them

const int

int const

int j = 12 ;

const int &K = j ;

K++ ;

j++ ;

j → 800

K → 800
↓
const

Constant.

800

Const int &K = j

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5    // constant int
6    const int i = 10;
7    int const i2 = 10;
8
9    // constant reference from a non const int
10   int j = 12;
11   const int & k = j;
12   j++;
13
14   cout << k << endl;
15
16   // constant reference from a const int
17   int const j2 = 12;
18   int const &k2 = j2;
19
20   // reference from a const int
21   int const j3 = 123;
22   int & k3 = j3;
23   k3++;
24
25
26 }
27
```

* you can't store the address of const Variable:

```
int const i = 10;
int      *p = &i;
```
} ✗

```
int const i = 10;
int const *p = &i
```
} ✓

Benefit of const:

Its major Benefit is in function calling

```
~/Documents/Inception — vi constant_pointers.cpp
 1 #include <iostream>
 2 using namespace std;
 3
 4 int main() {
 5   int i = 10;
 6   int j = 21;
 7   int const * p = &i;
 8
 9   p = &j;
10
11   int * const p2 = &i;
12   (*p2)++;
13   //p2 = &j;
14
15
16   int const * const p3 = &i;
17 }
18
```

What is the output

Send Feedback

What is the output of the following program ?

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

int main(){
   int  const  p = 5;
   cout << ++p;
   return 0;
}
```

You have max 2 attempts to score in this question.

Options Attempts left: 1/2

This problem has only one correct answer

- 5
- 6
- Error ✔
- Garbage

Correct Answer

Solution Description

Values of constant variables cannot be altered

p = 5

4 ✗

5

What is the output

Send Feedback

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

int main(){
   int p = 5;
   int const *q = &p;
   p++;
   cout << p << endl;
   return 0;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem has only one correct answer

- ○ Error
- ○ 5
- ● 6 ✔
- ○ None

Correct Answer

Solution Description

Value of p can be increased or decreased as it is not a const value.

Constants

Send Feedback

Which statement(s) will give an error for the following code -

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

int main(){
   int p = 5;
   int const *q = &p;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 0/2

This problem may have one or more correct answers

- ☑ p++;
- ☐ q++;
- ☐ (*q)++; ✔

The solution to this problem has been viewed

Constants

Send Feedback

Which statement(s) will give an error for the following code -

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

int main(){
   int p = 5;
   int * const q = &p;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem may have one or more correct answers

- ☐ p++;
- ☑ q++; ✔
- ☐ (*q)++;

Correct Answer

Constants

Send Feedback

Which statement(s) will give an error for the following code -

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

int main(){
   int p = 5;
   int const * const q = &p;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem may have one or more correct answers

- ☐ p++;
- ☑ (*q)++; ✔
- ☑ q++; ✔

Correct Answer

Constants

Send Feedback

Which statement(s) will give an error for the following code -

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

int main(){
    const int p = 5;
    int const *q = &p;
}
```

You have max 2 attempts to score in this question.

Options

Attempts left: 1/2

This problem may have one or more correct answers

- [x] p++; ✔
- [ ] q++
- [x] (*q)++; ✔

Correct Answer