**References**

**Writer:** Sayak Haldar

**1. Which value we cannot assign to reference?**

a) integer

b) floating

c) unsigned

d) null

**Answer: d**

Explanation: If it can be assigned with a null value means, it is a copy of pointer.

**#include<cstdio>**

**int main()**

**{**

**int &ref\_var=NULL;**

**return 0;**

**}**

This will generate a compilation error.

Even if your code contains such kind of code for error protection:

if (&object1) {}

You can get a warning message saying reference cannot be bound to dereferenced null pointer in well-defined C++ code; pointer may be assumed to always convert to true

**2. Which reference modifier is used to define reference variable?**

a) &

b) $

c) #

d) none of the mentioned

**Answer: a**

**3. What is the output of this program?**

#include <iostream>

using namespace std;

void swap(int &a, int &b);

int main()

{

int a = 5, b = 10;

swap(a, b);

cout << "In main " << a << b;

return 0;

}

void swap(int &a, int &b)

{

int temp;

temp = a;

a = b;

b = temp;

cout << "In swap " << a << b;

}

a) In swap 105 In main 105

b) In swap 105 In main 510

c) In swap 510 In main 105

d) None of the mentioned

**Answer: a**

Explanation: As we are calling by reference the values in the address also changed. So the main and swap values also changed.

Output:

$ g++ ref.cpp

$ a.out

In swap 105 In main 105

**4. What does a reference provide?**

a) Alternate name for the class

b) Alternate name for the variable

c) Alternate name for the pointer

d) None of the mentioned

**Answer: b**

Explanation: Because we are pointing memory address using temp variable.

**5. What is the output of this program?**

#include <iostream>

using namespace std;

int main()

{

int a = 9;

int & aref = a;

a++;

cout << "The value of a is " << aref;

return 0;

}

a) 9

b) 10

c) error

d) 11

View Answer

Answer: b

Explanation: The value is declared and it is post incremented, so it’s value is 10.

$ g++ ref1.cpp

$ a.out

10

**6. What is the output of this program?**

#include <iostream>

using namespace std;

void print (char \* a)

{

cout << a << endl;

}

int main ()

{

const char \* a = "Hello world";

print(const\_cast<char \*> (a) );

return 0;

}

a) Hello world

b) Hello

c) world

d) compile time error

**Answer: a**

Explanation: In this program we used the concept of constant casting to cast the constness away from the variable and printing it.

Output:

$ g++ ref2.cpp

$ a.out

Hello world

However,   
  
 **#include <iostream>**

**using namespace std;**

**void print (const char \* a)**

**{**

**cout << a << endl;**

**}**

**int main ()**

**{**

**const char \* a = "Hello world";**

**print(a);**

**//print(const\_cast<char \*> (a) );**

**return 0;**

**}**

You can obviously do it.

What you cannot do is (it will invoke compilation error)

**#include <iostream>**

**using namespace std;**

**void print (char \* a)**

**{**

**cout << a << endl;**

**}**

**int main ()**

**{**

**const char \* a = "Hello world";**

**print(a);**

**//print(const\_cast<char \*> (a) );**

**return 0;**

**}**

**7. Identify the correct sentence regarding inequality between reference and pointer.**

a) we can not create the array of reference

b) we can create the Array of reference

c) we can use reference to reference

d) none of the mentioned

**Answer: a**

**8.What will be the output of the following code:**#include<cstdio>

#include<iostream>

using namespace std;

int main()

{

int &x=6;

//x is a non const reference variable

//6 is an integral constant

cout<<"X is: "<<x<<endl;

return 0;

}

a)6

b)Compilation error

c) Memory location of 6

d) Undefined behaviour  
  
**Answer) is b)**

A non const reference cannot be initialized with a const l value. (literal)

1. **What is the output of the following code:**

#include<cstdio>

#include<iostream>

using namespace std;

int main()

{

int x=6,y=7;

int &ref=x\*y;

//x\*y is an expression: r value

//ref is a non const reference

cout<<"Ref is: "<<ref<<endl;

return 0;

}

a)42

b)Compilation error

c) Memory location of 42

d) Undefined behaviour

**Answer) is b)**

A non const reference cannot be initialized with a const r value

1. **Identify the incorrect statement**

a) Reference is the alternate name of the object

b) A reference value once defined can be reassigned

c) A reference value once defined cannot be reassigned

d) None of the mentioned

**Answer) b)**  
It cannot be done. Consider the following example:

**#include<cstdio>**

**#include<iostream>**

**using namespace std;**

**int main()**

**{**

**int y=6;**

**int z=7;**

**int &x=y;**

**&x=z;**

**//you cannot reassign it**

**x=z;**

**//now, this is different. This will change y's value to 7**

**return 0;**

**}**